

Mehrkörpersimulation eines ebenen Koppelgetriebes mittels *Matlab / ADAMS*-Co-Simulation

Dipl.-Ing. Christian Gollee

Institut für Naturstofftechnik, Professur für Verarbeitungsmaschinen/Verarbeitungstechnik

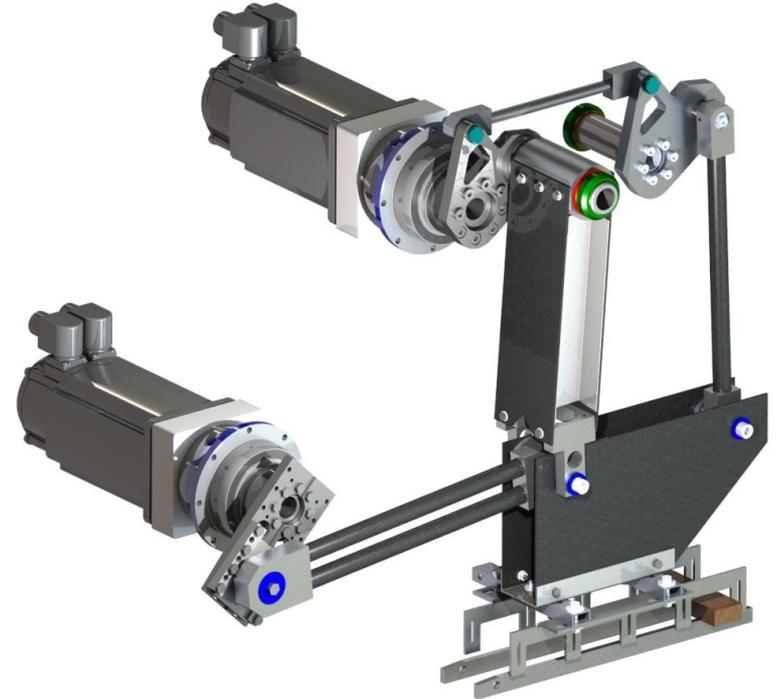


Gliederung

1. Problemstellung
2. Aufbau *ADAMS*-Modell
3. Einrichten einer Schnittstelle
4. Aufbau *Simulink*-Modell
5. Beispielanwendung
6. Ergebnisdarstellung

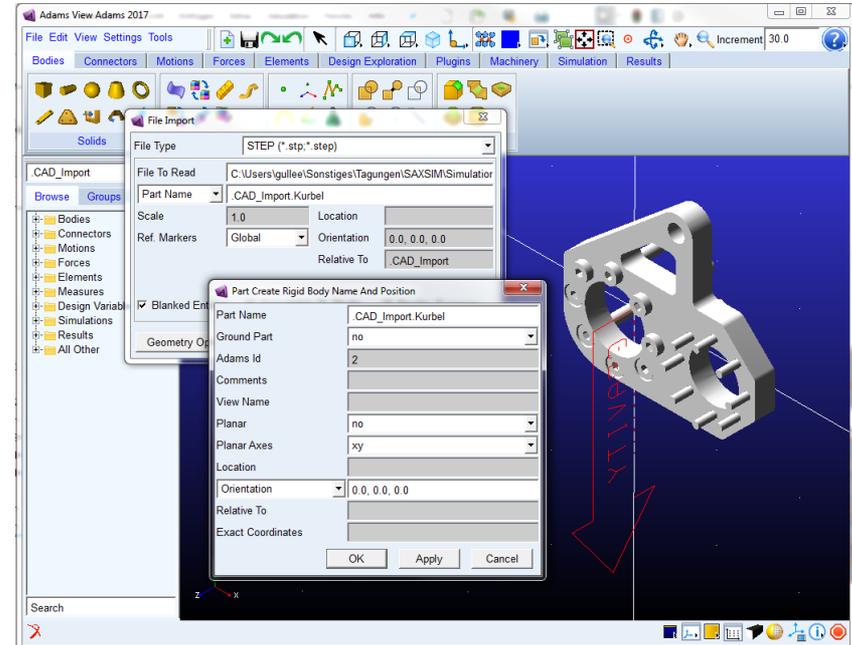
1. Problemstellung

- System: Ebenes Koppelgetriebe für intermittierenden Stückguttransport (Laborversuchsstand)
- Ziel: Aufbau eines Simulationsmodells für Bewegungsplanung, Prozessanalyse, Strukturuntersuchung, virtuelle Inbetriebnahme, etc.
- Anforderungen: Berücksichtigung von Bauteilelastizitäten, Motor- und Getriebereibung und Reglerverhalten



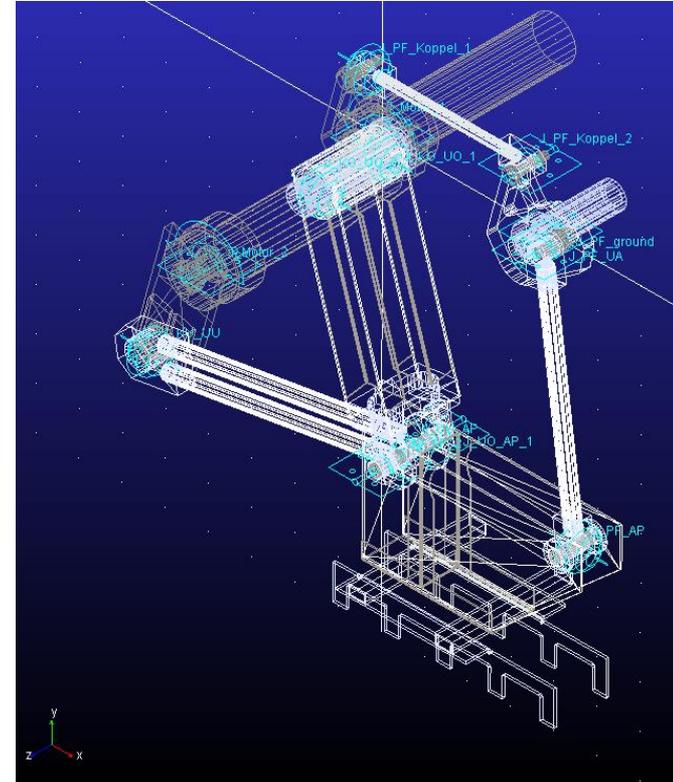
2. Aufbau *ADAMS*-Modell

- Modellierung einfacher Geometrien ist in *ADAMS* direkt mit *Primitives* möglich
- Für komplexe Bauteile → Import von CAD-Daten (*.step, *.stl, *.x_t, ...)
- Platzierung durch Angabe von Position und Orientierung
- Zuweisung von Masseigenschaften durch Angabe von Dichte oder Material oder direkte Vorgabe



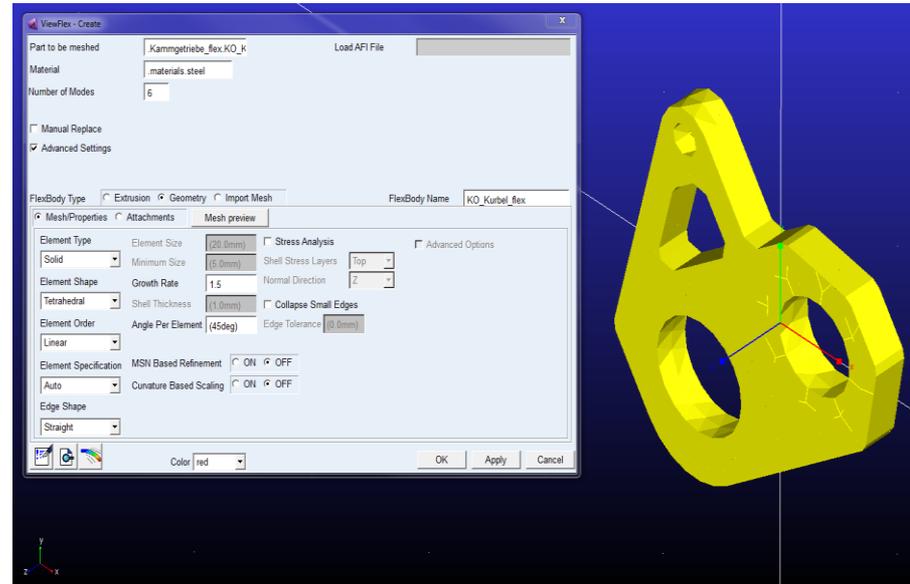
2. Aufbau *ADAMS*-Modell

- Verknüpfung der Komponenten durch einfache oder komplexe Zwangsbedingungen (*Joints, Primitives*)
- Vorgabe von gewünschten Bewegungen an ausgewählten Gelenken über *Motions*:
 1. Interne *ADAMS*-Funktionen
 - *STEP/STEP5* (Polynome)
 - *HAVSIN*
 - *SIN,COS,TANH*
 2. Einlesen von *Test Data*
→ Splineinterpolation



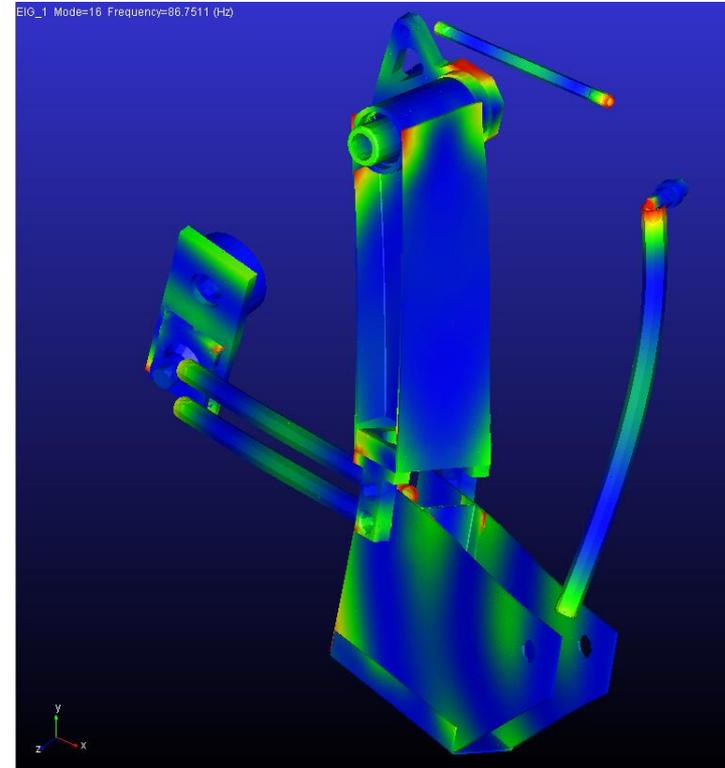
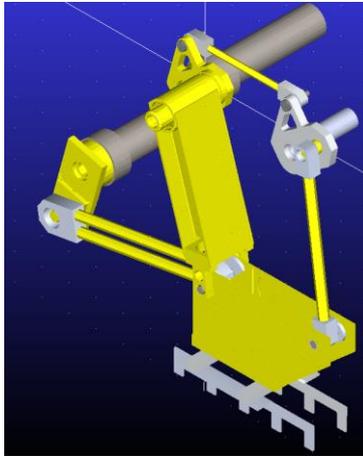
2. Aufbau *ADAMS*-Modell

- Umwandlung von starren in flexible Bauteile
- Abbildung des linearen Verformungsverhaltens
- Speicherung der Informationen in MNF-Dateien (modal neutral file) → Geometrie, Masse- und Trägheitseigenschaften, Eigenmoden und generalisierte Masse und Steifigkeiten für Eigenmoden



2. Aufbau *ADAMS*-Modell

- Beurteilung des Einflusses einzelner Elastizitäten auf das Systemverhalten
- Abgleich mittels Modalanalyse



3. Einrichten der Schnittstelle

Aktionen in ADAMS

1. Anlegen von Zustandsvariablen in ADAMS (Schnittstellenparameter)
 - Eingangsparameter sollten Kraftgrößen sein (keine kinematischen Vorgaben)
 - Ausgangsparameter können sowohl kinematische als auch kinetische Größen sein
2. Export eines ADAMS-Plants für Matlab
→ *.m, *.cmd, *.adm werden erstellt

The image shows three overlapping windows from the ADAMS software interface:

- Modify State Variable ... (top):** Name: N_Motor_1, Definition: Run-Time Expression, F(time, ...) = 0, Guess for F(t=0) = 0.0.
- Modify State Variable ... (middle):** Name: Riegel_vx, Definition: Run-Time Expression, F(time, ...) = VX(Riegel_Ist, Riegel_Referenz, Riegel_Referenz), Guess for F(t=0) = 0.0. A large blue '1.' is overlaid on this window.
- Adams Controls Plant Export (bottom):** New Controls Plant: .Kammgetriebe_flex_CPControls_Plant_4, File Prefix: CP_flex, Initial Static Analysis: No, Initialization Command: (empty), Input Signal(s): IN_Motor_1, Output Signal(s): Riegel_vx, Target Software: MATLAB, Analysis Type: non_linear, Adams Solver Choice: C++, User Defined Library Name: (empty), Adams Host Name: GOLLEE.vat.mw.tu-dresden.de. A large blue '2.' is overlaid on this window.

Below the windows, three files are listed:

- CP_flex.adm
- CP_flex.cmd
- CP_flex.m

3. Einrichten der Schnittstelle

Aktionen in Matlab:

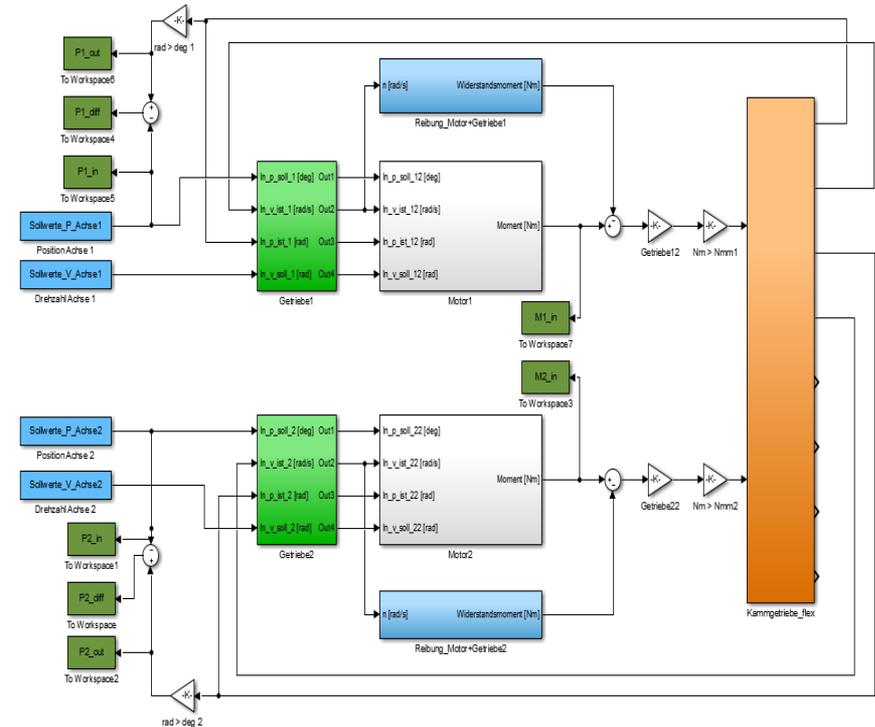
1. Ausführen der *CP_flex.m*
2. Ausführen des Befehls *adams_sys*
3. Kopieren des Blocks *adams_sub* und Einfügen in *Simulink*-Modell
4. Kommunikationsfrequenz festlegen

The image displays four screenshots illustrating the setup of the ADAMS-plant interface in MATLAB/Simulink:

- 1. Command Window:** Shows the execution of `CP_flex` and `adams_sys`. The output lists ADAMS plant actuators (IN_Motor_1, IN_Motor_2) and sensors (OUT_ANG_Motor_1, OUT_AVEL_Motor_1, OUT_ANG_Motor_2, OUT_AVEL_Motor_2).
- 2. Workspace:** Lists variables created by the `adams_sys` command, including `ADAMS_cwd`, `ADAMS_exec`, `ADAMS_host`, `ADAMS_init`, `ADAMS_inputs`, `ADAMS_mode`, `ADAMS_outputs`, `ADAMS_pinout`, `ADAMS_poutput`, `ADAMS_prefix`, `ADAMS_solver_type`, `ADAMS_static`, `ADAMS_sysdir`, `ADAMS_uj_ids`, `ADAMS_version`, `ans`, `arch`, `block`, `block_name`, `flag`, `machine`, and `topdir`.
- 3. Simulink Model:** Shows a Simulink diagram with an `adams_sub` block connected to `MSC Softwares` (S-Function) and `ANSYS/PT-CXNDU` (State Space) blocks. The `adams_sub` block has multiple input and output ports labeled `OUT_ANG`, `OUT_AVEL`, `Riege1_x`, and `Riege1_vx`.
- 4. Block Parameters: ADAMS Plant:** A dialog box for configuring the ADAMS plant. Key parameters include:
 - Adams model file prefix: `ADAMS_prefix`
 - Output files prefix: `ADAMS_prefix`
 - Adams Solver type: `C++`
 - Interprocess option: `PFPE(ODE)`
 - Animation mode: `batch`
 - Simulation mode: `discrete`
 - Plant input extrapolation order: `0`
 - Plant output extrapolation order: `0`
 - Communication interval: `1/Freq`
 - Number of communications per output step: `1`

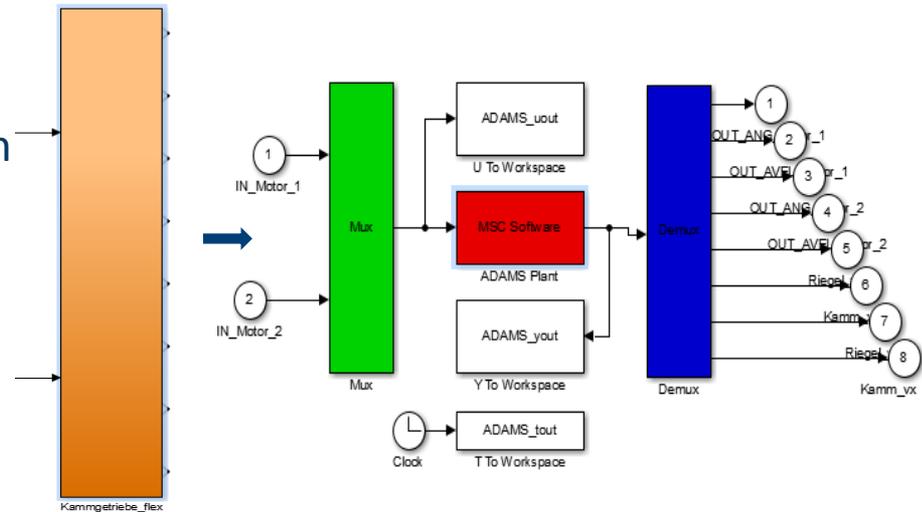
4. Aufbau *Simulink*-Modell

- Abbildung beider Motoren inklusive Getriebe
- Darstellung der Motoren über Reglermodelle
- Winkel- und Drehzahlvorgabe → Regler → Antriebsmoment → *ADAMS*
- Vorgabe eines Reibmodells
- Daten Input/Output von und in den *Matlab-Workspace* für weitere Verarbeitung und Darstellung



4. Aufbau *Simulink*-Modell

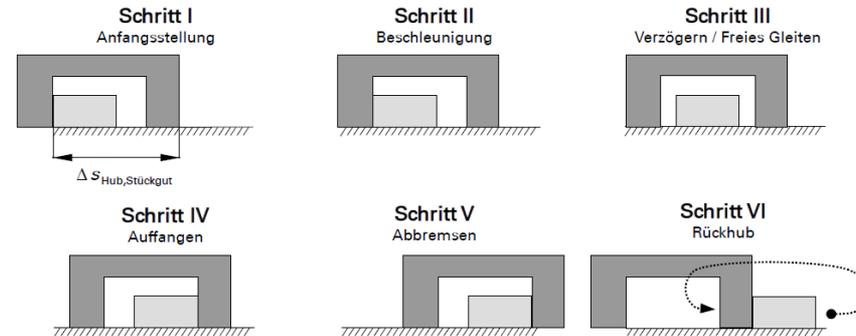
- Integration des *ADAMS*-Modells in Form eines *ADAMS-Plants*
- Verknüpfung des *ADAMS-Plants* mit den entsprechenden Signalleitungen
- *ADAMS* kann interaktiv (erhöhte Rechenzeit) oder im Batch-Modus aufgerufen werden
- Alle Größen von Interesse mit *ToWorkspace*-Block in den *Matlab-Workspace* schreiben



5. Beispielanwendung

Riegeltransport

- Intermittierender Stückguttransport
- 3 Phasen:
 - Beschleunigen
 - Gleiten
 - Abbremsen
- Taktrate: 5 Hz

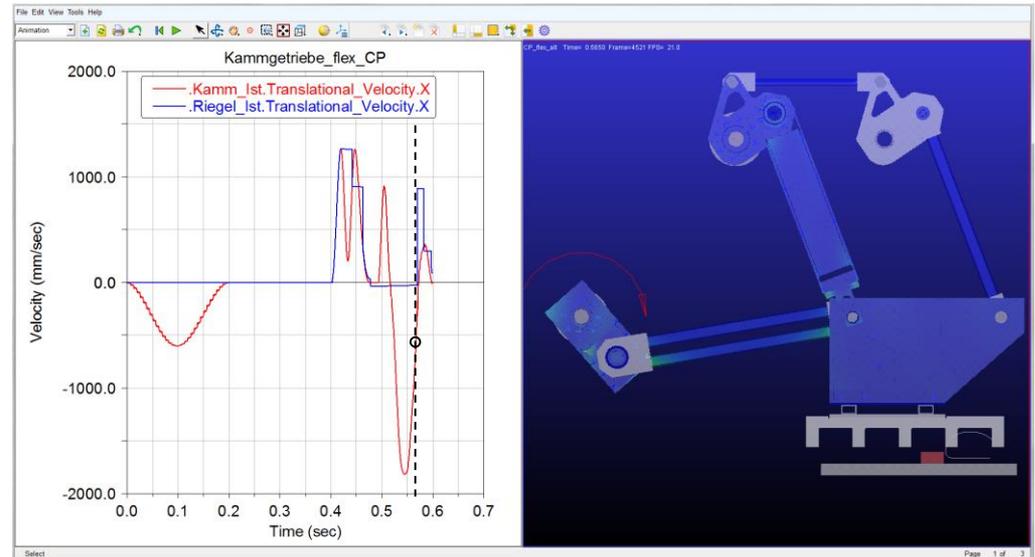


Quelle: „Wirkpaarungssimulation am Beispiel des innermaschinellen Transports von Stückgütern“, DA Clemens Troll; <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-213381>

5. Ergebnisdarstellung

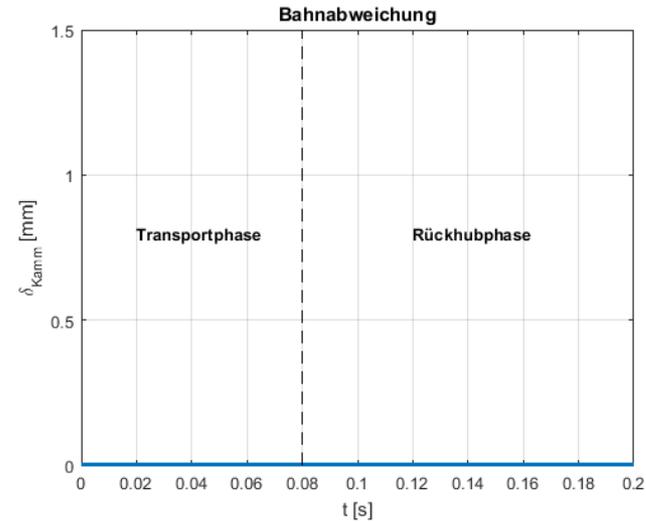
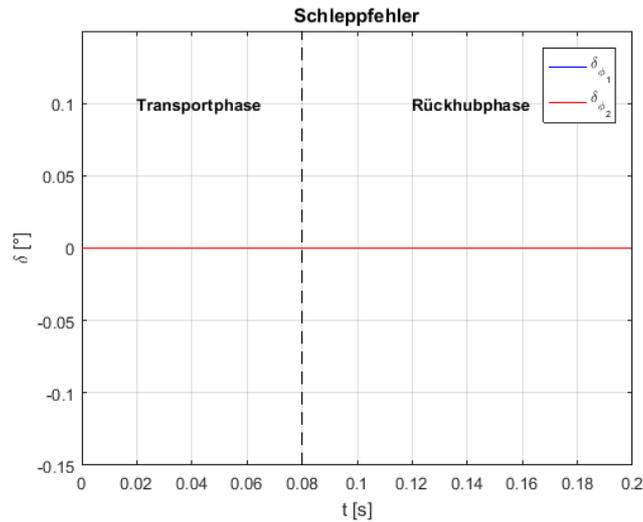
ADAMS-Postprozessor

- Auswertung als Diagramm oder Animation (oder parallel)
- Export als Report-Datei (HTML-Format)
- Erstellen von Video-Dateien
- Statistische Auswerte-Tools
 - ➔ Min, Max, Mittel, FFT, Filter, ...



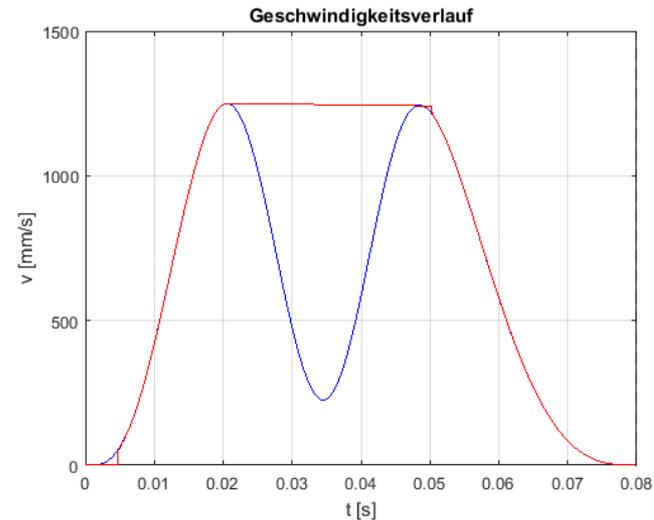
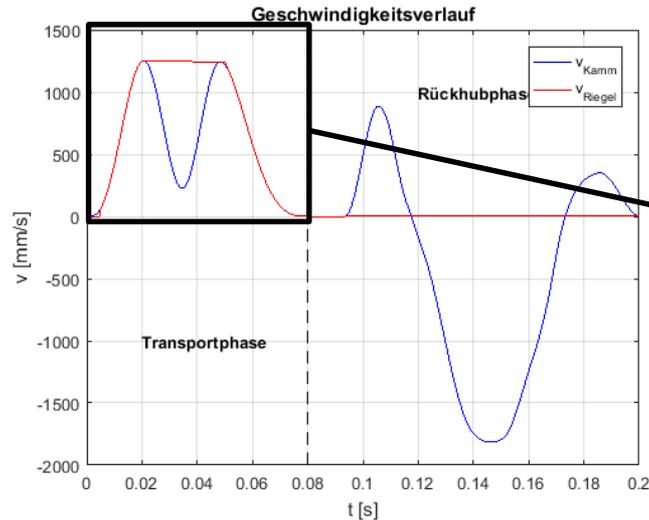
6. Ergebnisdarstellung

Modellstufe 1: Starrkörpermodell ohne Regler und ohne Reibung



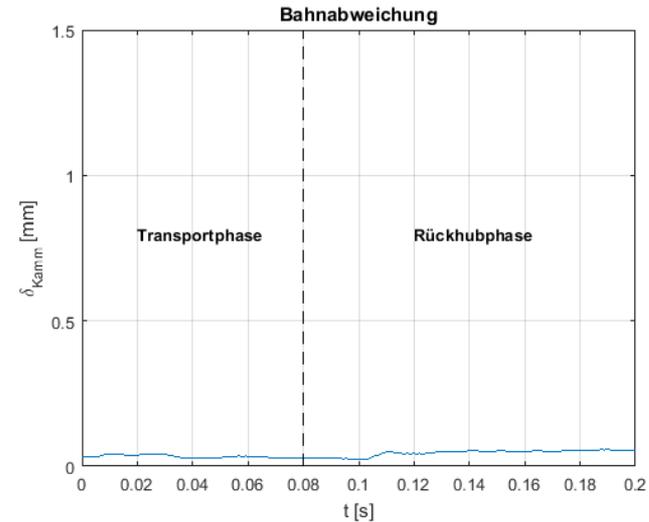
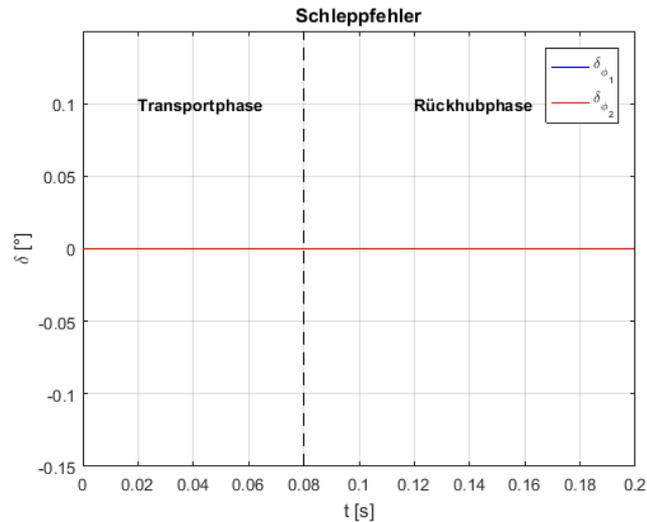
6. Ergebnisdarstellung

Modellstufe 1: Starrkörpermodell ohne Regler und ohne Reibung



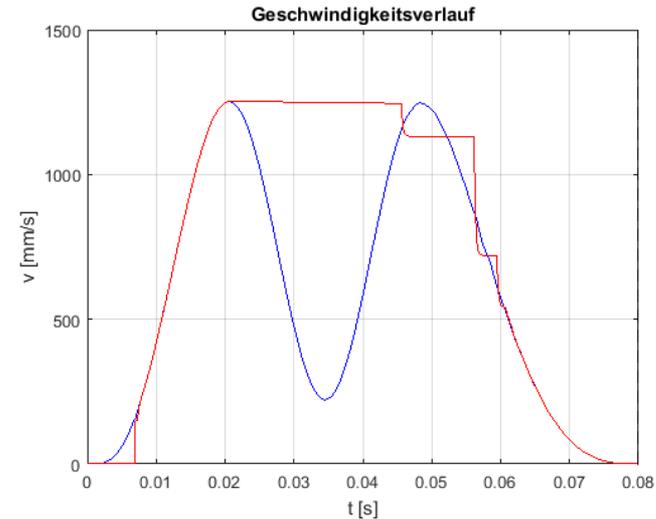
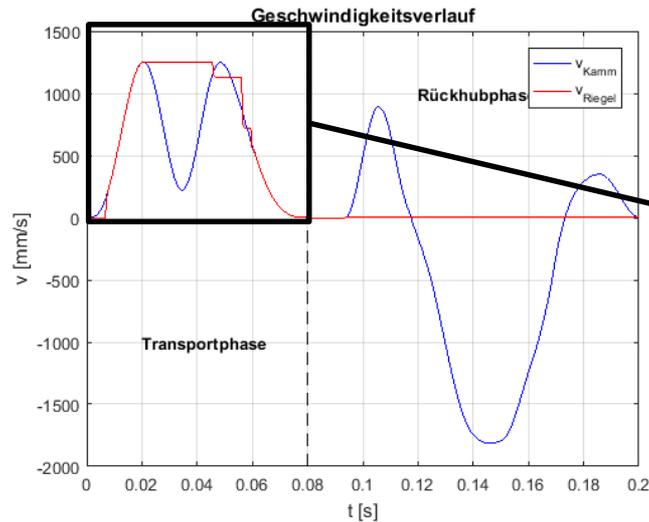
6. Ergebnisdarstellung

Modellstufe 2: flexibles Modell ohne Regler und ohne Reibung



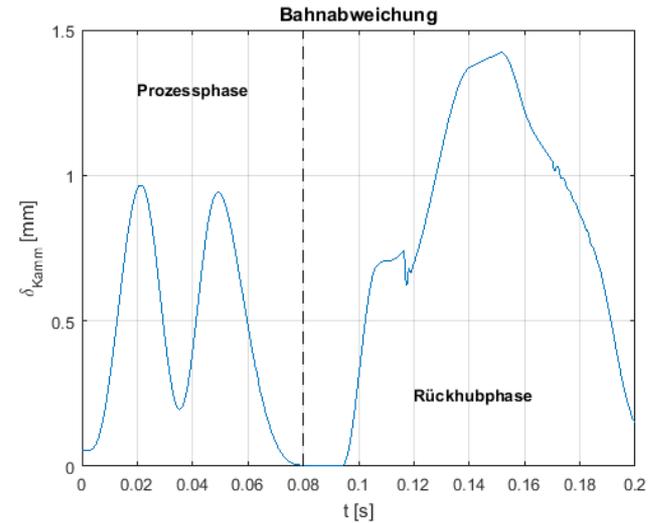
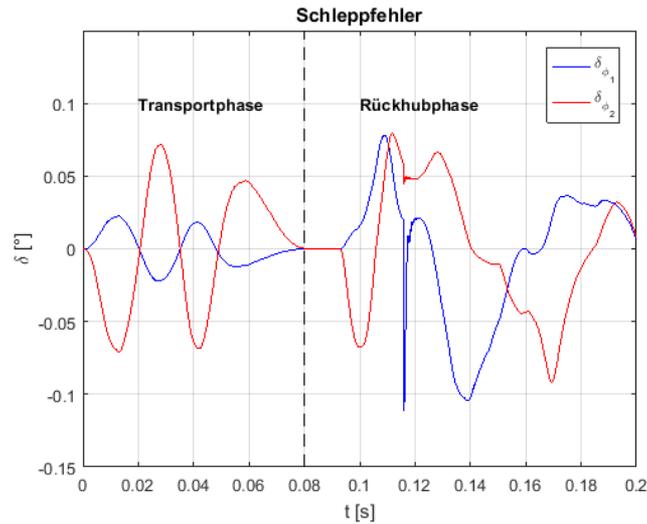
6. Ergebnisdarstellung

Modellstufe 2: flexibles Modell ohne Regler und ohne Reibung



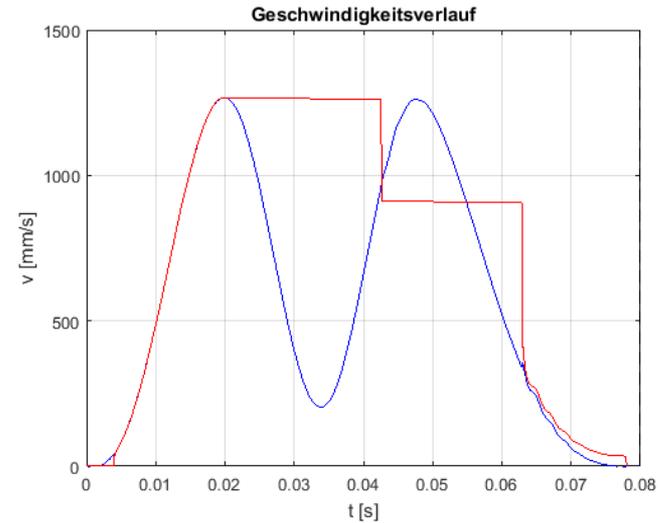
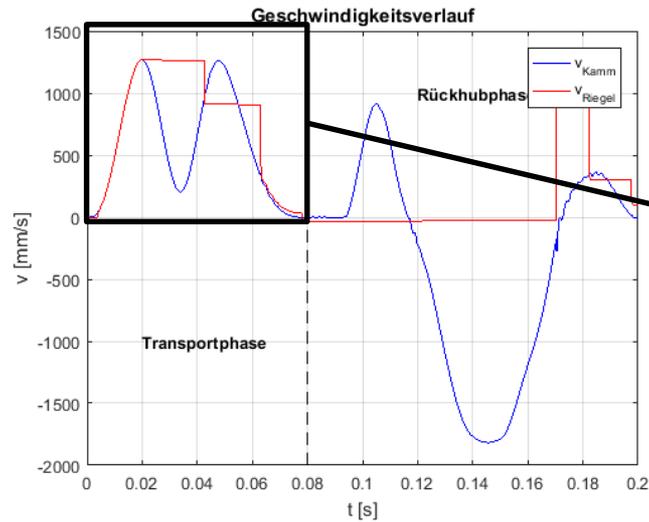
6. Ergebnisdarstellung

Modellstufe 3: flexibles Modell mit Regler und mit Reibung



6. Ergebnisdarstellung

Modellstufe 3: flexibles Modell mit Regler und mit Reibung



6. Ergebnisdarstellung

Zusammenfassung

- Gegenüber einer reinen kinetostatischen Analyse in Matlab ermöglicht das Einbinden flexibler Bauteile die Berücksichtigung des elastischen Verhaltens des Systems
- Gegenüber einer reinen MKS-Analyse in *ADAMS* ermöglicht die Co-Simulation die Berücksichtigung des Reglerverhaltens
- Co-Simulation kann eingesetzt werden für:
 1. Virtuelle Inbetriebnahme (Einstellen der Reglerparameter)
 2. Steifigkeitsoptimierte Dimensionierung des Mechanismus
 3. Ermitteln der Stabilitätsgrenzen des Transportprozesses



FAKULTÄT
MASCHINEN
WESEN

TECHNIK ÜBERWINDET GRENZEN