

Formatabhängige hochdynamische Bewegungen mit Servoantrieben

Dipl.-Ing. Dipl-Inform. Rainer Nolte
Nolte NC-Kurventechnik GmbH, Bielefeld

Rückblick, 1996: Buchstapler mit 10 Servoantrieben

Ziel: maximale Taktzahl für alle Formate

Formatvorgaben:

Buchhöhe 100 bis 480 mm, min. 9 Varianten
Buchdicke 5 bis 80 mm, min. 16 Varianten
Breite 70 bis 360 mm, min. 7 Varianten
Übermaß Breite: min. 3 Varianten
Anzahl Produkte auf Teilstapel: 20 Varianten
Anzahl Teilstapel: 10 Varianten
Mit/ohne Criss-Cross-Versatz: 2 Varianten
Teilstapel drehen ja/nein: 2 Varianten
Lücke bei Abgabe: min. 3 Varianten

Insgesamt $9 \times 16 \times 7 \times 3 \times 20 \times 10 \times 2 \times 2 \times 3$
= 7257600 Formatvarianten

Wie sollen die optimierten Bewegungsdiagramme in der Maschinensteuerung abgebildet werden?

Realisierung mit festen Stützpunkttabellen
(1996 üblich) in einer Datenbank nicht möglich
wegen Rechenzeit und Speicherbedarf



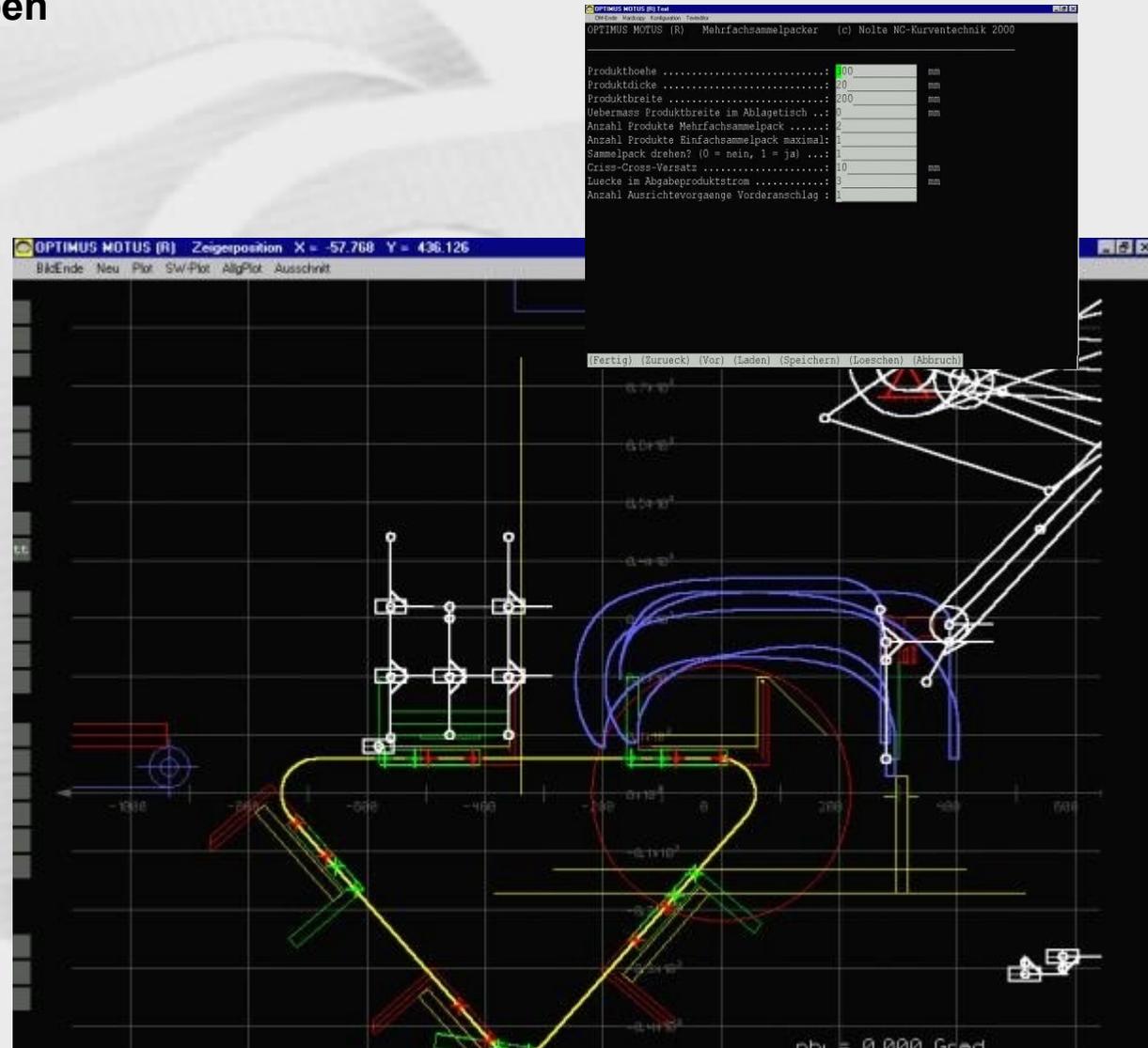
Buchstapler mit 10 Servoantrieben

Lösung 1996:

Spezialprogramm berechnet
bei Formatwechsel
auf dem PC neben der Maschine
Bewegungstabellen für ein neues
Produktformat

- + Bewegungsqualität gut, aber ...
- PC an der Maschine erforderlich
- keine Integration in das SPS-Projekt
- manuelle Übertragung der Tabellen
- Entwicklung aufwändig
- Spezialprogramm schlecht änderbar
- Zum Testen der Bewegungen muss
zusätzlich eine Simulation programmiert
werden

**Wunsch damals:
Grafisches Entwicklungswerkzeug!**

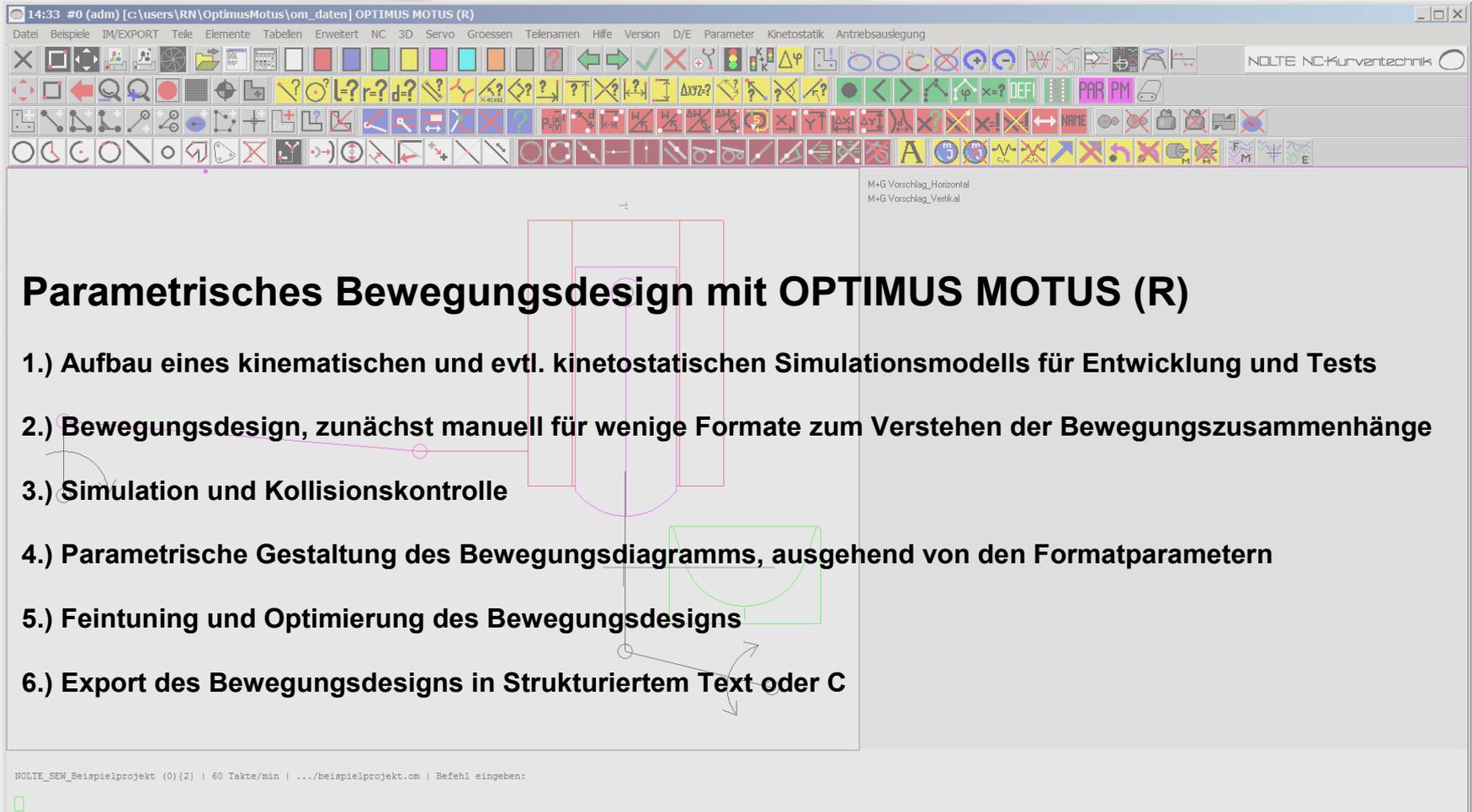


Typischer Entwicklungsansatz heute:

Spezialfunktionen im SPS-Projekt berechnen
bei Formatwechsel oder zur Laufzeit
Stützpunkte für ein neues Produktformat,
die von den Servocontrollern interpoliert werden, meist mit Polynomen 5. Grades

- Bewegungsqualität mittelprächtig (ruckfrei, aber ohne Rücksicht auf Schwingungen)
- + Berechnung in SPS-Projekt integriert
- Entwicklung der Bewegungsberechnungen aufwändig wegen der Abhängigkeiten der Stützpunkte
- Zum Testen der Bewegungen muss zusätzlich eine Simulation programmiert werden
- Die SPS-Berechnung der Stützpunkte und Bewegungen muss für die Simulation nochmal entwickelt werden
- Spezialprogramm schlecht änderbar

Wunsch immer noch: Grafisches Entwicklungswerkzeug!



Parametrisches Bewegungsdesign mit OPTIMUS MOTUS (R)

- 1.) Aufbau eines kinematischen und evtl. kinetostatischen Simulationsmodells für Entwicklung und Tests
- 2.) Bewegungsdesign, zunächst manuell für wenige Formate zum Verstehen der Bewegungszusammenhänge
- 3.) Simulation und Kollisionkontrolle
- 4.) Parametrische Gestaltung des Bewegungsdiagramms, ausgehend von den Formatparametern
- 5.) Feintuning und Optimierung des Bewegungsdesigns
- 6.) Export des Bewegungsdesigns in Strukturiertem Text oder C

2. Bewegungsdesign

Bewegungsvorgabe am Abtrieb während der Synchronfahrt und am Antrieb in den Umkehrbereichen

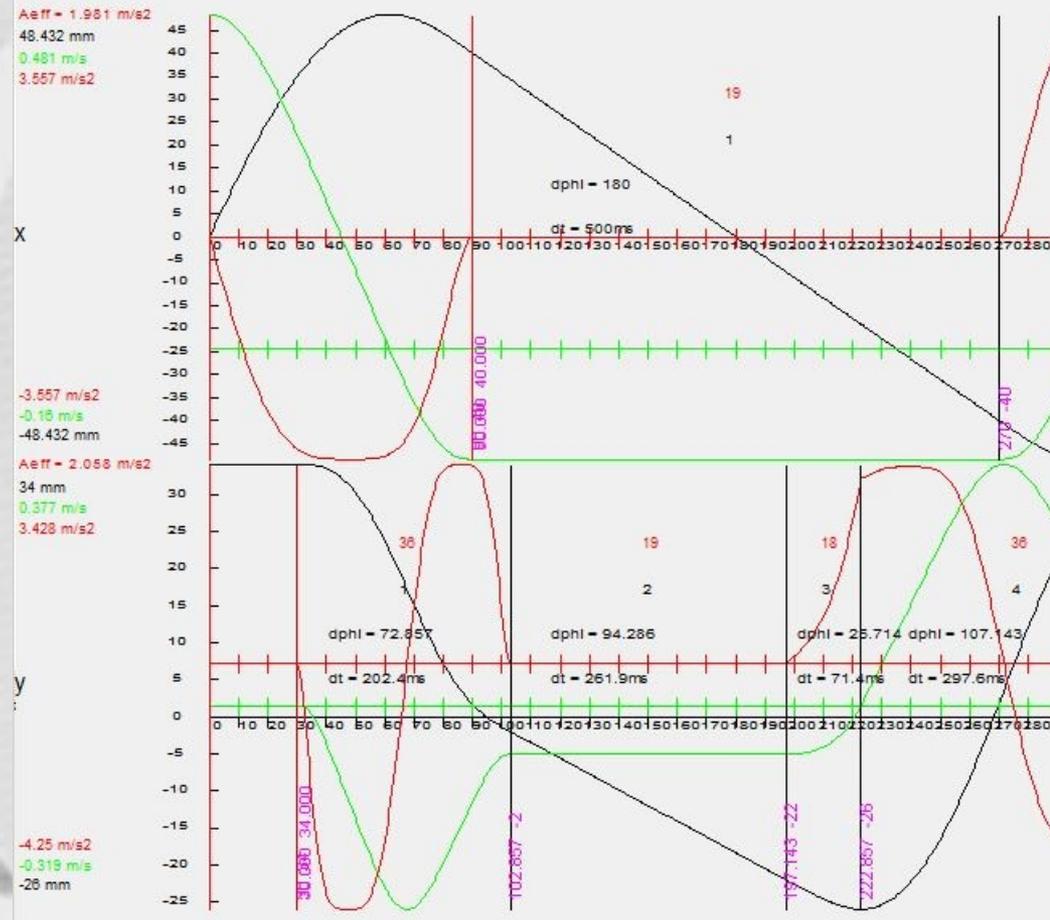
Grobe zeitliche Verteilung der Stützpunkte

Gestaltung der Bewegungen hin auf kleine Beschleunigungen, Antriebsmomente, Gestellkräfte, Schwingungsanregung usw.

mit

- Ruckfreien Bewegungsgesetzen
- Wendepunktverschiebungen, Geradenanteilen
- Allgemeinen Sinuskombinationen
- Stetiger dritter oder vierter Ableitung
- Frei gestalteten Polynomen
- Polynomsplines
- HS-Profilen
- Polydyn-Funktionen

Die Qualität der Bewegungsgestaltung entscheidet mit über die erreichbare Taktzahl!



2. Bewegungsdesign (Fortsetzung)

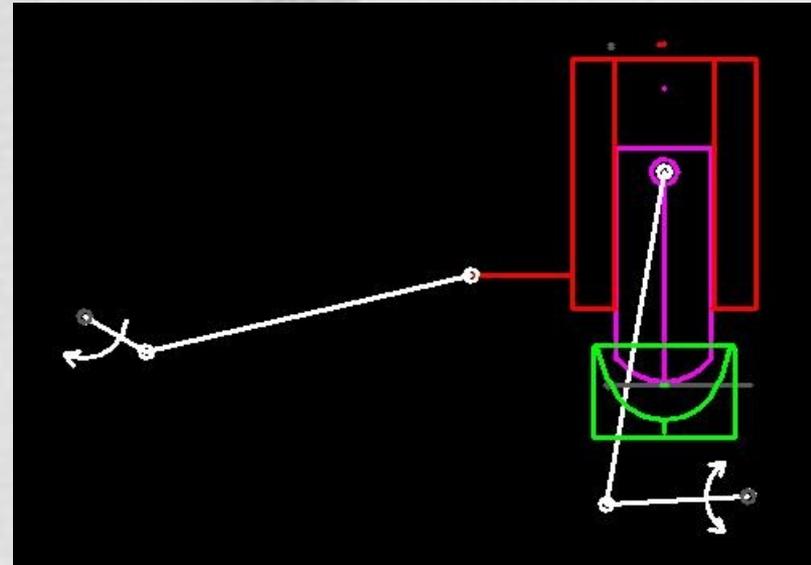
Bewegungsvorgabe: $x(\varphi)$ = Horizontalbewegung, $y(\varphi)$ = Vertikalbewegung des Stempels

Ermittlung der Kurbelwinkel $\psi_{\text{horizontal}}(\varphi)$ und $\psi_{\text{vertikal}}(\varphi)$:

$$\psi_{\text{horizontal}}(\varphi) = \arctan(-y_1 / (x(\varphi) - x_1)) - \arccos((h_5 + h_1 + (x(\varphi) - x_1)^2 - h_2) / (2 l_1 \sqrt{h_1 + (x(\varphi) - x_1)^2}))$$

$$z(\varphi) = l_1 \cos(\psi_{\text{horizontal}}(\varphi)) + \sqrt{h_2 - (l_1 \sin(\psi_{\text{horizontal}}(\varphi)) + y_1)^2} + x_1$$

$$\psi_{\text{vertikal}}(\varphi) = \arccos(((z(\varphi) - x_2)^2 + (y(\varphi) - y_2)^2 + h_3 - h_4) / (2 l_2 \sqrt{(z(\varphi) - x_2)^2 + (y(\varphi) - y_2)^2})) + \pi/2 - \arctan((z(\varphi) - x_2) / (y(\varphi) - y_2))$$



3. Simulation und Kollisionskontrolle

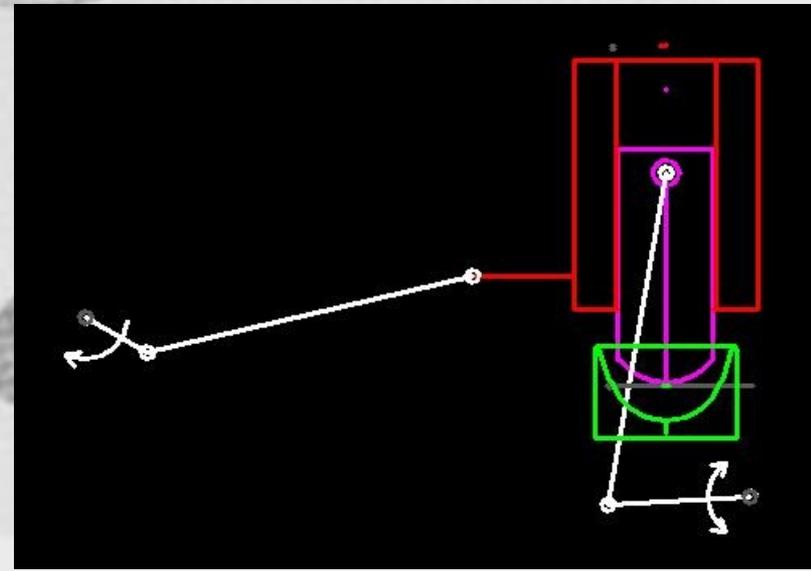
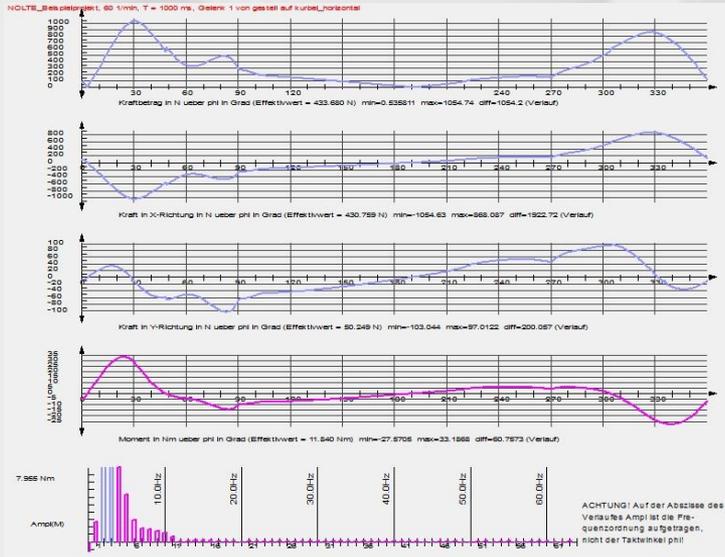
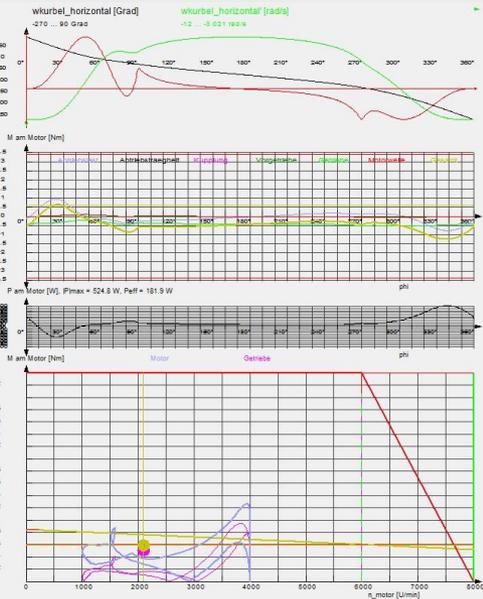
Kinematische Simulation des Bewegungsablaufs

Kinetostatische Berechnung der Gelenklasten, Antriebsmomente, dynamischen Gestellkräfte

Servoantriebsbewertung oder -auslegung

Erweitern des Bewegungsdiagramms um Stützpunkte oder Gestaltungsmerkmale zur Kollisionsvermeidung und Lastverlaufsoptimierung

OPTIMUS MOTUS ® Mechanismensoftware: Antriebsbewertung
 Datensatzname: E_NOLTE_Beispielprojekt (G1: Vorschlag_Horizontal)
 Bewegungsverlauf: w_kurbel_horizontal, Winkel in Grad (am Antrieb)
 Taktfrequenz: 50 Taktschritte
 Zykluszeit: 1000 ms
 Motor-Code: MOT_siemens-19/7022-5ak71_bremse
 Getriebe-Code: GTR_alpha-sp_100s-sp100s_mc2_35_0e1_05
 Motor und Getriebe passen vielleicht zusammen (prüfen!)
 Antrieb:
 J_antrieb = 0 kg*m²
 M_antrieb_const = 0 Nm
 Getriebe: GTR_alpha-sp_100s-sp100s_mc2_35_0e1_05 (Katalogauswahl)
 J_getriebebewegung = 0.000072 kg*m²
 M_const_getriebe = 14 Nm
 L_getriebe = 1.35
 M_nenn_getriebe = 140 Nm
 M_max_getriebe = 240 Nm
 N_max_getriebe = 8000 U/min
 eta_getriebe = 99.5 Prozent
 stossfaktor_getriebe = 2
 Motor: MOT_siemens-19/7022-5ak71_bremse (Katalogauswahl)
 J_motor = 0.000055 kg*m²
 M_nenn_motor = 0.6 Nm
 M_max_motor = 3.4 Nm
 N0_motor = 0.05 Nm
 N_nenn_motor = 8000 U/min
 N_max_motor = 8000 U/min
 Momentenverhältnis = 1 : 13.3 (Moment durch Motorgetriebegetriebe: Fremdmoment)
 Mittlere Drehzahl am Motor: 2100.0 U/min
 Max. Drehzahl am Motor: 4010.7 U/min
 Eff. Motormoment: 0.589 Nm
 Max. Motormoment: 1.267 Nm
 Maximalle Antriebsleistung: 524.8 W
 Effektive Antriebsleistung: 181.9 W
 M_eff = 129.5 Nm
 Eff-Moment Getriebeausgang: 17.708 Nm
 Max. Gesamtmoment Getriebeausgang: 33.187 Nm
 Legende zur Grafik am Ende der Tabelle!
 Lastspalte: SDATEINom_dyn_max.txt erzeugt!
 Lastspalte: SDATEINom_dyn_max.txt erzeugt!



4. Parametrische Gestaltung des Bewegungsdiagramms

Vorgabewerte (Formatparameter):

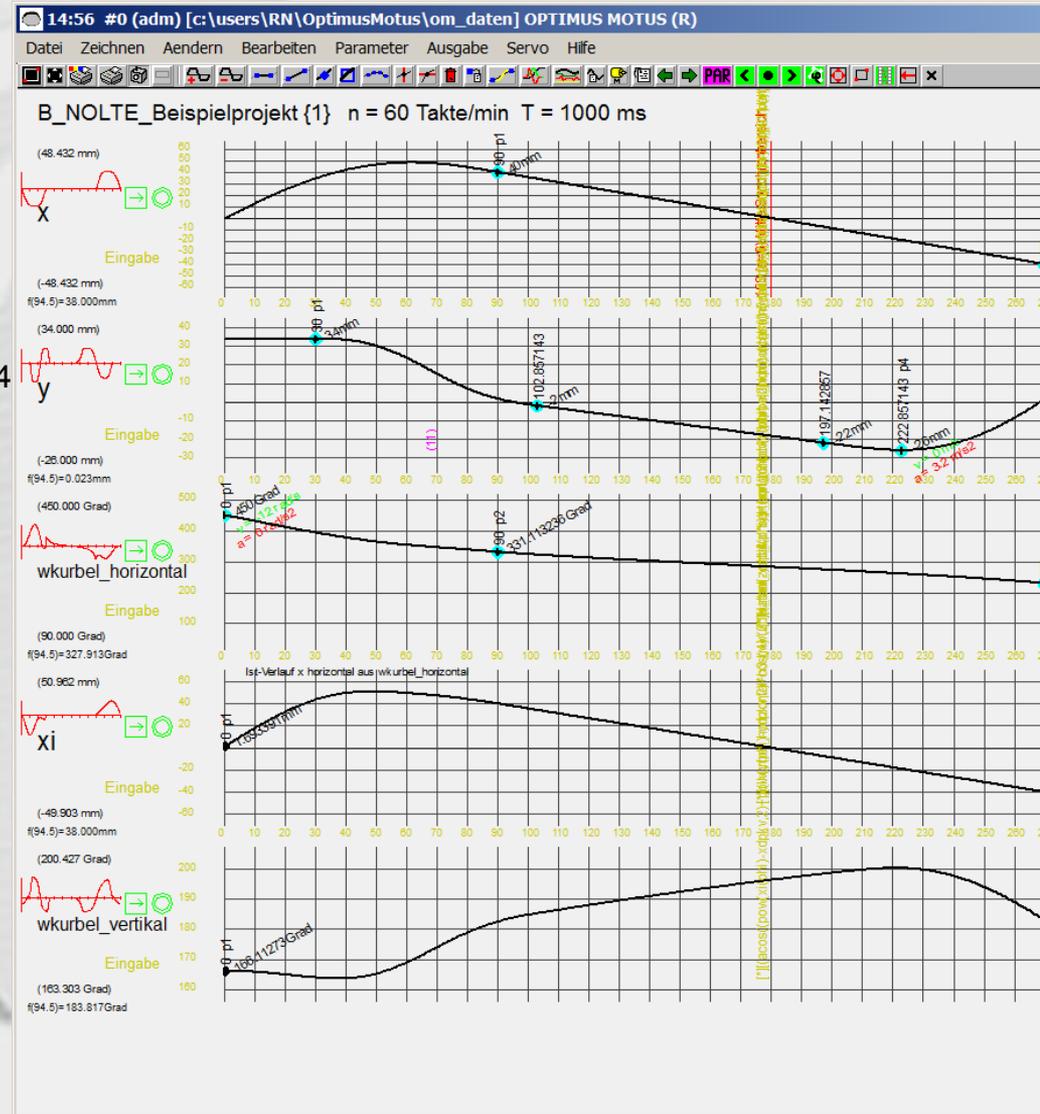
- becherabstand_auf_kette = 160
- hub_startwert = 20
- taktzahl = 60
- synchronweg = 80
- vertikalhub = 80
- ...

Abhängige Parameter:

- $c_kette = becherabstand_auf_kette / 360 = 0.444$
- $h1 = pow(ydp_kurbel_horizontal, 2) = 900$
- $h2 = pow(lkoppel_horizontal, 2) = 55225$
- $h3 = pow(lkurbel_vertikal, 2) = 10000$
- $h4 = pow(lkoppel_vertikal, 2) = 57600$
- $h5 = pow(lkurbel_horizontal, 2) = 2500$
- $y_{max} = (204 - d_tcp_koppelanlenkpunkt) = 54$
- $y_{min} = y_{max} - vertikalhub = -26$
- ...

Definition der Wegkoordinaten, Taktwinkel, Geschwindigkeitswerte usw. von Stützpunkten abhängig von allen Parameterwerten

Definition von Übertragungsfunktionen, inversen Übertragungsfunktionen durch Nullstellensuche, Kennlinien, Kraft-, Momenten- und sonstigen Zwischengrößenverläufen für Zwischenrechnungen



5. Feintuning und Optimierung des Bewegungsdesigns

Parametrische Verallgemeinerung des Know Hows, das in den manuellen Simulationen gewonnen wurde

Erweiterung um „Tuning-Parameter“ (z.B. Wegkoordinate, Geschwindigkeit, Beschleunigung in Stützpunkten) zur Kollisionsvermeidung oder zur Optimierung beliebiger Zielgrößen (z.B. maximales Antriebsmoment)

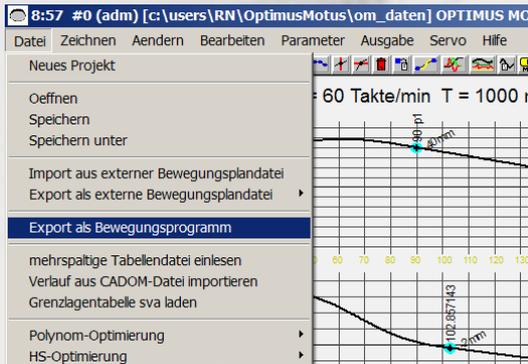
Verwendung von Kennlinien, Kennfeldern und Ähnlichkeiten zur formatabhängigen Festlegung der Tuningparameter

Abwägung zwischen

Analytischer Herleitung und **empirischer Ermittlung** guter Tuningparameterwerte

6. Export des Bewegungsdesigns in Strukturiertem Text oder C

Der letzte Schritt: das Programmieren der Bewegungsberechnung: vollautomatisch, wie ein Postprozessorlauf

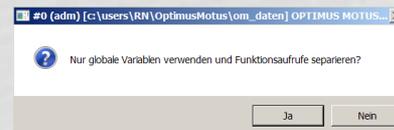


```
vereinfachtes C
vereinfachtes C fuer Dateizerlegung
strukturiertes Text (*.ST)
strukturiertes Text fuer CoDeSys 2.3 (*.EXP)
strukturiertes Text fuer Simotion
strukturiertes Text fuer Elau (mit CoDeSys)
strukturiertes Text fuer Rockwell RSLogix5000
```

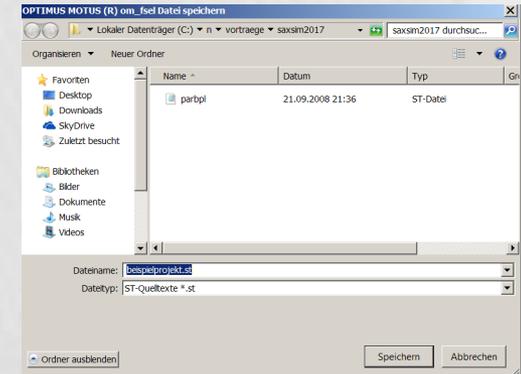
Sprachdialekt wählen

```
FUNCTION cpt_bewegungsdiagramm : DINT
VAR_IN_OUT
stab_kette:ARRAY [0..720] OF LREAL;
stab_x:ARRAY [0..720] OF LREAL;
stab_y:ARRAY [0..720] OF LREAL;
stab_wkurbel_horizontal:ARRAY [0..720] OF LREAL;
stab_xi:ARRAY [0..720] OF LREAL;
stab_wkurbel_vertikal:ARRAY [0..720] OF LREAL;
END_VAR
VAR_INPUT
FNPARG_inpar_becherabstand_auf_kette:LREAL;
FNPARG_inpar_hub_startwert:LREAL;
FNPARG_inpar_ntakte:LREAL;
FNPARG_inpar_synchronweg:LREAL;
FNPARG_inpar_vertikalhub:LREAL;
FNPARG_phi1:LREAL;
FNPARG_phi2:LREAL;
FNPARG_ntab:DINT;
END_VAR
VAR
```

Exportierter SPS-Funktionsbaustein mit schmaler, einheitlicher Schnittstelle (API)



Rücksicht auf ST-Dialekte ohne lokale Variablen und Funktionsparameter



Zieldatei wählen

Ergebnis

OPTIMUS MOTUS (R) ist ein grafischer Editor, um komplexe Bewegungsabläufe zu modellieren, zu optimieren, zu testen und schließlich als Funktionsbausteine für die SPS-Welt zu exportieren.

So können SPS-Bewegungsprogramme erheblich schneller entwickelt und geändert werden als bei manueller Programmentwicklung.

Die aus der Kurventechnik bekannte Bewegungsqualität kommt auch bei Servoantrieben zum Tragen.

Das Debugging entfällt, weil die Quelltexte maschinell erzeugt werden.