

Bericht

für die Manz Automation AG, Reutlingen

Dynamische Modellierung einer Delta-Kinematik und Implementierung von Computed-Torque-Algorithmen

20.09.2007

Inhaltsverzeichnis

I	Einleitung	2
1	Verkoppelte Bewegungen	2
II	Modellierung	4
2	Kinematik	4
2.1	Zwangsbedingungen	4
2.2	Inverse Kinematik	5
2.3	Differentialkinematik	6
2.4	Jacobimatrix	6
3	Dynamik	8
3.1	Lagrange-Euler-Ansatz	8
3.1.1	Potentielle Energie	9
3.1.2	Kinetische Energie	9
3.1.3	Lösung der Dynamikgleichungen	10
III	Implementierung	12
4	Validierung	13
4.1	Numerisches MKS-Modell	13
4.2	Simulationsergebnisse	15
4.2.1	Reduziertes Modell	15
4.2.2	Fehlparametrisierung der Nutzlast	16
IV	Anhang	17
A	Modellierung der vierten Achse	17
B	Hilfsgrößen der dynamischen Modellierung	18
C	Computeralgebra-Skripte	19
D	Parametrisierung des dynamischen Modells	21

Teil I

Einleitung

1 Verkoppelte Bewegungen

Um die Leistungsfähigkeit von Roboterregelungen zu erhöhen, kann von der Integration von Modellen der zu regelnden Strecke profitiert werden. Dies ist in der Regel umso erfolgreicher, je stärker die reale Strecke von der bei der Reglerauslegung als linear angenommenen theoretischen Strecke abweicht. Abbildung 1 zeigt den Aufbau einer konventionellen Regelung ohne Berücksichtigung der Verkopplungen im Modell.

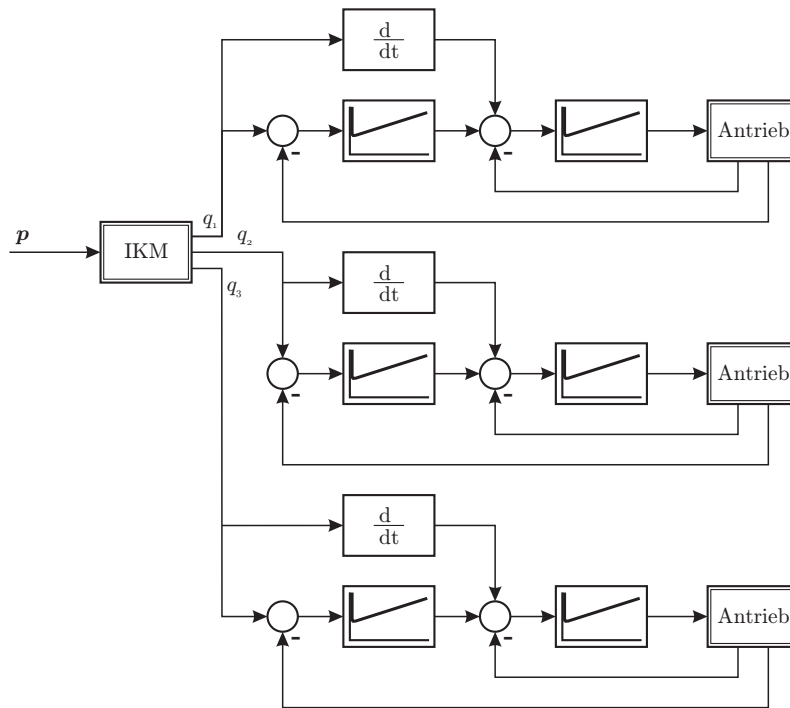


Abbildung 1: Regelungsschema ohne Berücksichtigung von Verkoppelungen

Hier sind alle Achsen des Roboters unabhängig voneinander geregelt. Die kartesische Pose \mathbf{p} aus dem Interpolator wird durch ein inverses kinematisches Modell (IKM) in Solllagen \mathbf{q} für die Antriebe umgerechnet. Durch die Struktur einkoppelnde Kräfte von einer jeweils anderen Achse werden als Störgröße aufgefasst und es muss zu einem Bahnfehler kommen, bis der geschlossene Regelkreis den Roboter dem theoretischen, linearen Verhalten annähert. Dies kann nur mit zeitlicher Verzögerung erfolgen und führt zu Bahnabweichungen am Endeffektor. Die durch Ableitung von \mathbf{q} gewonnenen Geschwindigkeitsvorsteuerungen berücksichtigen nicht, dass die nötige Geschwindigkeit eines Antriebs um der Bahn zu folgen, nicht nur von der Lage des Antriebs selbst abhängt, sondern auch von den durch die Bewegung der Struktur auferlegten Zwangsbedingungen. Werden diese Zwangsbedingungen verletzt, kommt es zu Störkräften in der Struktur, die den zuvor beschriebenen Effekt auf die Bahntreue zeigen.

In einem ersten Schritt liegt es also nahe, die kinematischen Zwangsbedingungen bei der Berechnung der Geschwindigkeitsvorsteuerung zu berücksichtigen. Dies ist die einfachste Ausprägung einer modellbasierten Regelung. Ihr Aufbau ist in Abbildung 2 dargestellt. Hier wird ein erweitertes IKM genutzt und die Achsregler von der Aufgabe befreit, die kinematischen Kopplungen zu kompensieren. Diese können daher mit erhöhter Bandbreite dazu eingesetzt werden, den Roboter zu linearisieren. Das Resultat ist eine erhöhte Bahntreue und eine bessere Dynamik des Roboters.

Dazu müssen aus dem Interpolator nicht nur die kartesische Pose \mathbf{p} , sondern auch die Bahngeschwindigkeiten $\dot{\mathbf{p}}$ gewonnen werden. Steht $\dot{\mathbf{p}}$ dort nicht zur Verfügung, so können diese Größen durch numerische Differentiation von \mathbf{p} mit hinreichender Genauigkeit gewonnen werden.

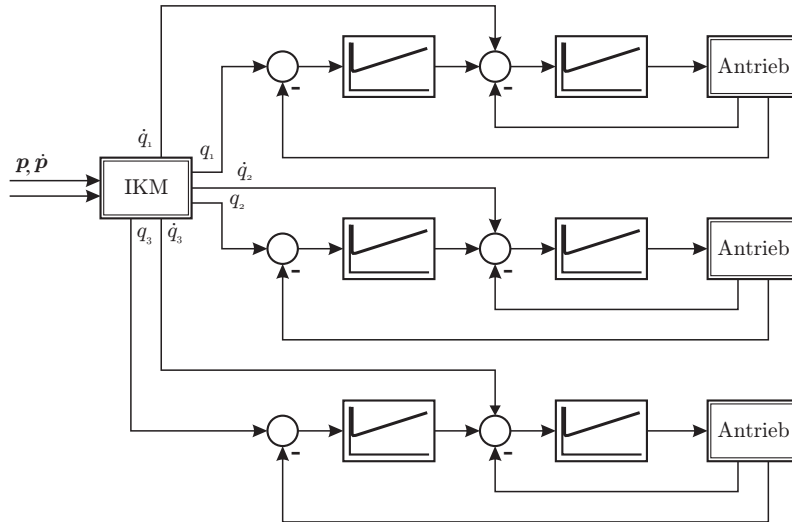


Abbildung 2: Regelungsschema mit modellbasierter Vorsteuerung

Die Achsregler können jedoch auch von der linearisierenden Funktion befreit werden, wenn die Stellgrößen für die Antriebe τ , die erforderlich sind, um der Bahn zu folgen, nicht eingeregelt werden müssen, sondern direkt aus einem inversen dynamischen Modell (IDM) gewonnen werden. Man spricht hier von Computed-Torque-Feedforward-Linearisierung, das entsprechende Schema ist in Abbildung 3 dargestellt. Der Achsregler wird hier nur noch dazu eingesetzt, Abweichungen im IDM zum tatsächlichen Roboter auszuregeln. Dies kann bei gutem Modell mit sehr hoher Bandbreite geschehen, das Resultat ist eine erhebliche Steigerung der Dynamik und damit auch der Bahntreue.

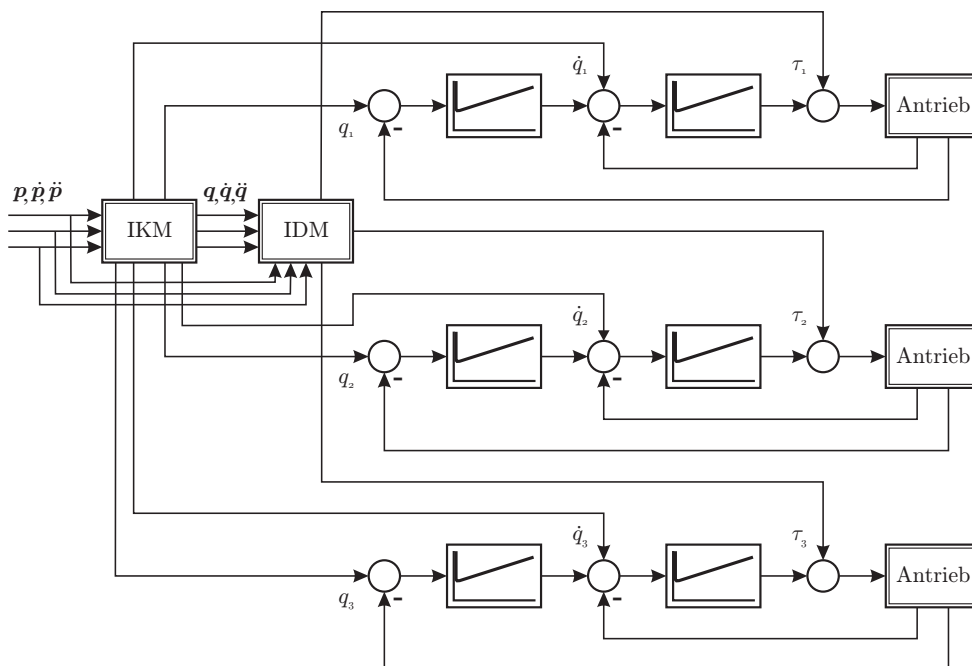


Abbildung 3: Linearisierende Regelung mit Momentenvorsteuerung

Das IKM muss erneut erweitert werden, um auch die Antriebsbeschleunigungen $\ddot{\mathbf{q}}$ für das IDM zu berechnen. IDM und IKM benötigen die durch $\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}$ vollständig beschriebene Trajektorie, deren Ableitungen $\dot{\mathbf{p}}, \ddot{\mathbf{p}}$ wie oben erwähnt auch numerisch gewonnen werden können. Der Aufbau der Modelle ist im folgenden Kapitel beschrieben.

Teil II

Modellierung

2 Kinematik

Abbildung 4 gibt eine Übersicht über die Modellierung des Roboters. Der Basiskoordinatensystem der Modellierung O liegt in der Ebene die durch die Antriebsachsen aufgespannt wird, äquidistant zu den Schnittpunkten der Antriebsachsen mit den Längsachsen der Kurbeln A . Die Schnittpunkte zwischen der Kurbellängsachse und der Parallelogrammbasis werden als B bezeichnet, die entsprechende Punkte an der Plattform als C .

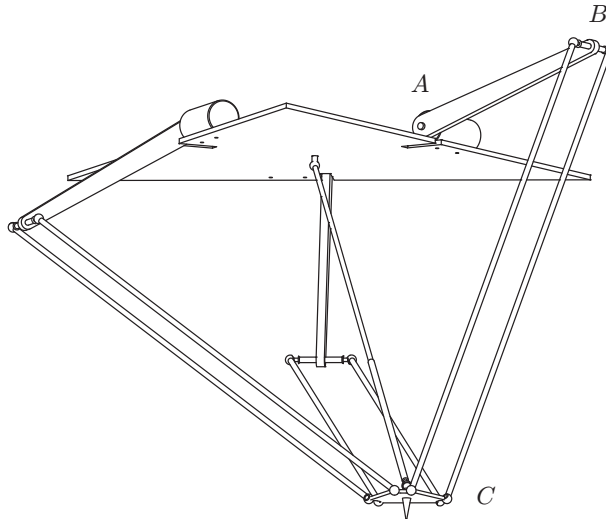


Abbildung 4: Übersicht über das Robotermodell

Die Antriebswinkel lauten q , die Position des Punktes P gegenüber dem Basiskoordinatensystem O , der mittig in der Plattform liegt, wird mit $p = [x, y, z]^T$ bezeichnet. Die vierte Achse lässt sich, da sie unabhängig von den ersten drei Achsen zu betrachten ist, einfach und getrennt modellieren. Ihr Posewinkel wird mit ϕ bezeichnet. Die entsprechenden Gleichungen finden Berücksichtigung in Anhang A.

2.1 Zwangsbedingungen

In der Abbildung 5 finden sich alle relevanten kinematischen Parameter und Formelzeichen. Die Parallelogramme sind für die Modellierung zu einem Stab zusammengefasst und der Plattform werden nur translatorische Freiheiten gewährt. Die Lage der Punkte B und C wird beschrieben durch:

$$\begin{aligned}
 B_{x,i} &= (r_a + l_a \cos q_i) \cos \theta_i \\
 B_{y,i} &= (r_a + l_a \cos q_i) \sin \theta_i \\
 B_{z,i} &= -l_a \sin q_i \\
 C_{x,i} &= x + r_b \cos \theta_i \\
 C_{y,i} &= y + r_b \sin \theta_i \\
 C_{z,i} &= z
 \end{aligned}$$

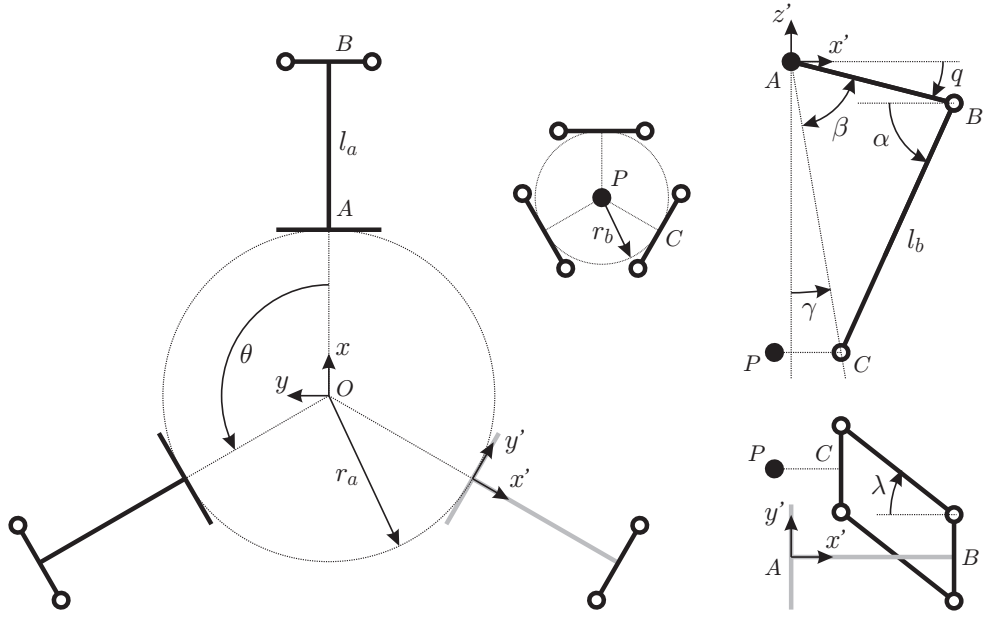


Abbildung 5: Relevante kinematische Größen

Der Verbindungsvektor zwischen den Punkten B und C ergibt sich demnach zu:

$$\begin{aligned} \overrightarrow{C_i B_i} &= \begin{bmatrix} (r_a + l_a \cos q_i) \cos \theta_i - r_b \cos \theta_i - x \\ (r_a + l_a \cos q_i) \sin \theta_i - r_b \sin \theta_i - y \\ -l_a \sin q_i - z \end{bmatrix} \\ &= \begin{bmatrix} (r + l_a \cos q_i) \cos \theta_i - x \\ (r + l_a \cos q_i) \sin \theta_i - y \\ -l_a \sin q_i - z \end{bmatrix} \end{aligned}$$

mit $r = r_a - r_b$ und $i = 1, 2, 3$.

Die Länge l_b des Parallelogramms ist konstant, somit ergibt sich mit

$$\left| \overrightarrow{C_i B_i} \right|^2 = l_b^2$$

die kinematische Zwangsbedingung:

$$\begin{aligned} f_i &= [(r + l_a \cos q_i) \cos \theta_i - x]^2 \\ &+ [(r + l_a \cos q_i) \sin \theta_i - y]^2 \\ &+ [-l_a \sin q_i - z]^2 - l_b^2 \stackrel{!}{=} 0 \end{aligned} \quad (1)$$

2.2 Inverse Kinematik

Um das Aufstellen und Lösen der kinematischen Gleichungen zu erleichtern werden die Endeffektorkoordinaten von der Roboterbasis O in die Punkte A_i so transformiert, dass die y_i -Achse in Richtung der Antriebsachse liegt und die x_i -Achse nach außen zeigt. Die Transformationsgleichungen lauten:

$$x'_i = x \cos \theta_i + y \sin \theta_i - r_a \quad (2)$$

$$y'_i = -x \sin \theta_i + y \cos \theta_i \quad (3)$$

$$z'_i = z \quad (4)$$

Im jeweiligen Bezugskordinatensystem in den Punkten A_i gilt:

$$B'_{x,i} = l_a \cos q_i \quad (5)$$

$$B'_{y,i} = 0 \quad (6)$$

$$B'_{z,i} = -l_a \sin q_i \quad (7)$$

$$C'_{x,i} = x'_i + r_b = l_a \cos q_i - l_b \cos \lambda_i \cos \alpha_i \quad (8)$$

$$C'_{y,i} = y'_i = l_b \sin \lambda_i \quad (9)$$

$$C'_{z,i} = z'_i \quad (10)$$

Berechnung der passiven Winkel λ_i über (9) und γ_i :

$$\lambda_i = \arcsin \frac{y'_i}{l_b} \quad (11)$$

$$\begin{aligned} \tan \gamma_i &= \frac{C'_{x,i}}{-C'_{z,i}} \\ \Rightarrow \gamma_i &= \arctan \frac{r_b - x'_i}{z'_i} \end{aligned} \quad (12)$$

Kosinussatz in dem Dreieck in der von O , A_i und B_i aufgespannten Ebene:

$$\begin{aligned} (l_b \cos \lambda_i)^2 &= l_a^2 + \left(\frac{C'_{x,i}}{\sin \gamma_i} \right)^2 - 2 \cdot l_a \left(\frac{C'_{x,i}}{\sin \gamma_i} \right) \cos \beta_i \\ \Rightarrow \beta_i &= \arccos \frac{l_a^2 + \left(\frac{C'_{x,i}}{\sin \gamma_i} \right)^2 - (l_b \cos \lambda_i)^2}{2 \cdot l_a \left(\frac{C'_{x,i}}{\sin \gamma_i} \right)} \end{aligned} \quad (13)$$

Damit lassen sich die Antriebswinkel q_i über die Summe der Komplementärwinkel ausdrücken:

$$q_i = \frac{\pi}{2} - \gamma_i - \beta_i \quad (14)$$

2.3 Differentialkinematik

Die Ableitung $\frac{df_i}{dt} = 0$ der kinematischen Zwangsbedingung (1) und Freistellen nach \dot{q} liefert:

$$\dot{q}_i = - \frac{(z + l_a \sin q_i) \dot{z} + [y + (-r - l_a \cos q_i) \sin \theta_i] \dot{y} + [x + (-r - l_a \cos q_i) \cos \theta_i] \dot{x}}{l_a (z \cos q_i + y \sin q_i \sin \theta_i + x \sin q_i \cos \theta_i - r \sin q_i)} \quad (15)$$

Erneutes Ableiten liefert:

$$\begin{aligned} \ddot{q}_i &= - \frac{Z}{N} \\ Z &= (z + l_a \sin q_i) \ddot{z} + \dot{z}^2 + 2l_a \cos q_i \cdot \dot{q}_i \dot{z} - l_a \sin q_i \cdot \dot{q}_i^2 z \\ &\quad + [y + (-r - l_a \cos q_i) \sin \theta_i] \ddot{y} + \dot{y}^2 + 2l_a \sin q_i \sin \theta_i \cdot \dot{q}_i \dot{y} + l_a \cos q_i \sin \theta_i \cdot \dot{q}_i^2 y \\ &\quad + [x + (-r - l_a \cos q_i) \cos \theta_i] \ddot{x} + \dot{x}^2 + 2l_a \sin q_i \cos \theta_i \cdot \dot{q}_i \dot{x} + l_a \cos q_i \cos \theta_i \cdot \dot{q}_i^2 x \\ &\quad - r l_a \cos q_i \cdot \dot{q}_i^2 \\ N &= l_a \cos q_i \cdot z + l_a \sin q_i \sin \theta_i \cdot y + l_a \sin q_i \cos \theta_i \cdot x - r l_a \sin q_i \end{aligned} \quad (16)$$

Damit lassen sich die nötigen Gelenkgeschwindigkeiten und -beschleunigungen für die Vorsteuerung und die Dynamikgleichungen aus den kartesischen Trajektorienaten errechnen. Hierzu muss zuvor die inverse Kinematik (14) berechnet werden um dann sukzessive (15) und (16) auszuwerten.

2.4 Jacobimatrix

Wird die Jacobimatrix für den Zusammenhang $\dot{\mathbf{q}} = \mathbf{J}(\mathbf{x}, \mathbf{y}, \mathbf{z})\dot{\mathbf{p}}$ benötigt, ohne zuvor die inverse Kinematik lösen zu müssen, so kann diese durch Differentiation der inversen Kinematik (14) gewonnen werden:

$$\frac{\partial q_i}{\partial \mathbf{r}'_i} = - \left(\frac{\partial \gamma_i}{\partial \mathbf{r}'_i} + \frac{\partial \beta_i}{\partial \mathbf{r}'_i} \right) \quad (17)$$

mit

$$\frac{\partial \gamma_i}{\partial \mathbf{r}'_i} = \frac{\partial}{\partial \mathbf{r}'_i} \arctan \frac{-C'_{x,i}}{z'_i} \quad (18)$$

und

$$\begin{aligned} \frac{\partial \beta_i}{\partial \mathbf{r}'_i} &= \frac{\partial}{\partial \mathbf{r}'_i} \arccos \frac{l_a^2 + \left(\frac{C'_{x,i}}{\sin \gamma_i}\right)^2 - (l_b \cos \lambda_i)^2}{2 \cdot l_a \left(\frac{C'_{x,i}}{\sin \gamma_i}\right)} \\ &= \frac{-1}{\sqrt{1 - \cos^2 \beta_i}} \cdot \left[\frac{\frac{\partial}{\partial \mathbf{r}'_i} \left[\left(\frac{C'_{x,i}}{\sin \gamma_i}\right)^2 - (l_b \cos \lambda_i)^2 \right]}{2 l_a \frac{C'_{x,i}}{\sin \gamma_i}} \right. \\ &\quad \left. - \left(l_a^2 + \left(\frac{C'_{x,i}}{\sin \gamma_i}\right)^2 - (l_b \cos \lambda_i)^2 \right) \left(\frac{\frac{\partial}{\partial \mathbf{r}'_i} \frac{C'_{x,i}}{\sin \gamma_i}}{\left(2 l_a \frac{C'_{x,i}}{\sin \gamma_i}\right)^2} \right) \right] \end{aligned} \quad (19)$$

Division des totalen Differentials dq_i durch dt liefert:

$$\dot{q}_i = \frac{\partial q_i}{\partial x'_i} \dot{x}'_i + \frac{\partial q_i}{\partial y'_i} \dot{y}'_i + \frac{\partial q_i}{\partial z'_i} \dot{z}'_i \quad (20)$$

So lassen sich durch Auswerten von (18) und (19) und anschließendem Einsetzen in (17) die Elemente für die i-te Spalte der Jacobimatrix für den Zusammenhang $\dot{\mathbf{q}} = \mathbf{J} \dot{\mathbf{p}}'$ finden. Sie ergeben sich zu

$$\frac{\partial q_i}{\partial x'_i} = \frac{z'_i}{z_i'^2 + C_{x,i}'^2} + \frac{\sin \gamma_i - C'_{x,i} \cos \gamma_i \cdot \frac{\partial \gamma_i}{\partial x'_i}}{\sqrt{1 - \cos^2 \beta_i}} \cdot \left[\frac{(l_b \sin \gamma_i \cos \lambda_i)^2 - (l_a \sin \gamma_i)^2 + C_{x,i}'^2}{2 l_a C_{x,i}'^2 \sin^2 \gamma_i} \right] \quad (21)$$

$$\frac{\partial q_i}{\partial y'_i} = \frac{\sin \gamma_i}{\sqrt{1 - \cos^2 \beta_i}} \cdot \frac{l_b^2 \cos \lambda_i \sin \lambda_i \cdot \frac{\partial \lambda_i}{\partial y'_i}}{l_a C'_{x,i}} \quad (22)$$

$$\frac{\partial q_i}{\partial z'_i} = \frac{-C'_{x,i}}{z_i'^2 + C_{x,i}'^2} + \frac{\cos \gamma_i \cdot \frac{\partial \gamma_i}{\partial z'_i}}{l_a \sqrt{1 - \cos^2 \beta_i}} \cdot \left[\frac{l_a^2 + \left(\frac{C'_{x,i}}{\sin \gamma_i}\right)^2 - (l_b \cos \lambda_i)^2}{2 C'_{x,i}} - \frac{C'_{x,i}}{\sin^2 \gamma_i} \right] \quad (23)$$

mit

$$\frac{\partial \gamma_i}{\partial x'_i} = -\frac{z'_i}{z_i'^2 + C_{x,i}'^2}, \quad \frac{\partial \lambda_i}{\partial y'_i} = \frac{1}{\sqrt{l_b^2 - y_i'^2}}, \quad \frac{\partial \gamma_i}{\partial z'_i} = \frac{C'_{x,i}}{z_i'^2 + C_{x,i}'^2}$$

Die Transformationsgleichungen von $\dot{\mathbf{p}}$ nach $\dot{\mathbf{p}}'_i$ ergeben sich durch Ableiten von (2) bis (4) nach der Zeit:

$$\dot{x}'_i = \dot{x} \cos \theta_i + \dot{y} \sin \theta_i \quad (24)$$

$$\dot{y}'_i = -\dot{x} \sin \theta_i + \dot{y} \cos \theta_i \quad (25)$$

$$\dot{z}'_i = \dot{z} \quad (26)$$

3 Dynamik

Die dynamische Modellierung erfolgt nach dem Lagrange-Euler-Ansatz, der darauf basiert, dass die Gesamtenergie des Systems sich aus potentieller und kinetischer Energie zusammensetzt. Es werden für die in Abbildung 6 abgebildeten Ersatzmassen und -trägheiten verwendet.

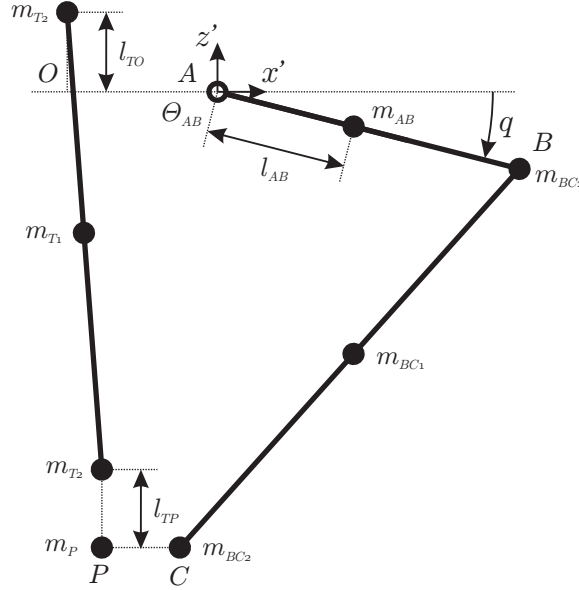


Abbildung 6: Äquivalente der dynamischen Modellierung

- m_P - Ersatzmasse für die Plattform
- m_T - Ersatzmasse für die Teleskopstange
- l_{TP} - Versatz des Teleskop-Gelenkpunktes an der Plattform in z-Richtung
- l_{TO} - Versatz des Teleskop-Gelenkpunktes zur Roboterbasis in z-Richtung
- m_{AB} - Ersatzmasse für die Kurbel
- Θ_{AB} - Trägheit der Kurbel um die Antriebsachse und des Antriebs
- l_{AB} - Lage der Ersatzmasse der Kurbel als Abstand von der Antriebsachse
- m_{BC} - Ersatzmasse für die Parallelgrammstangen

Dabei werden die Parallelgrammstäbe und die Teleskopstange mit einem Dreimassenäquivalent modelliert. Die Umrechnung erfolgt eingedenk der Vereinfachungen, dass das Parallelgramm wie ein Stab mit doppelter Masse und Trägheit aufgefasst wird und die Schwerpunkte der stabförmigen Bauteile in der Mitte der jeweiligen Hauptachse liegen:

$$m_{BC2} = \frac{4\Theta_{BC}}{l_b^2} \quad (27)$$

$$m_{BC1} = 2m_{BC} - 2m_{BC2} \quad (28)$$

$$m_{T2} = \frac{2\Theta_T}{x^2 + y^2 + (z + l_{TP} - l_{TO})^2} \quad (29)$$

$$m_{T1} = m_T - 2m_{T2} \quad (30)$$

Für die Kurbel wird bei der Modellierung der potentiellen Energie eine Ersatzmasse im Schwerpunkt berücksichtigt, bei der Modellierung der kinetischen Energie stattdessen die Trägheit der hierbei als masselos angenommenen Kurbel. Dies erlaubt, die Antriebsträgheit direkt als Summand der Kurbelträgheit zu parametrisieren.

3.1 Lagrange-Euler-Ansatz

Entgegen der intuitiven Vorgehensweise, nur drei Lagrange'sche Gleichungen den drei parallelen Freiheitsgraden der Kinematik entsprechend aufzustellen, erleichtert es erheblich den Rechenaufwand, generalisierte Koordinaten ξ zu verwenden:

So lässt sich das Verhalten der Struktur mit der Anzahl $j = 6$ generalisierter Koordinaten und der Anzahl $i = 3$ Zwangsbedingungen (siehe (1)) beschreiben.

Folglich werden die Lagrange'schen Gleichungen

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\xi}_j} \right) - \frac{\partial L}{\partial \xi_j} = \Xi_j + \sum_{i=1}^3 \mu_i \frac{df_i}{d\xi_j} \quad (31)$$

aufgestellt, wobei Ξ_j die generalisierten externen Kräfte sind, die Lagrange-Funktion

$$L = K - P \quad (32)$$

nach Euler die Differenz aus kinetischer Energie K und potentieller Energie P der Struktur darstellt und μ_i die Lagrange'schen Multiplikatoren bezeichnen.

Es existieren also 6 Gleichungen (31) für drei unbekannte Multiplikatoren μ_i und die drei gesuchten Momente der Antriebe $\tau_i = \Xi_{i+3}$, $i = 1, 2, 3$.

3.1.1 Potentielle Energie

Die potentielle Energie kann aufgrund der einfachen Zusammenhänge für den Roboter komplett modelliert werden. Als Vereinfachungen wurden angenommen, dass sich die Ersatzmassen der Parallelogrammstäbe und des Teleskopstabes mittig auf dem jeweiligen Glied befinden.

Plattform:

$$P_1 = m_P g \cdot z \quad (33)$$

Teleskopstange:

$$P_2 = m_{T2} g \cdot (z + l_{TP}) + m_{T1} g \cdot \frac{1}{2}(z + l_{TP} + l_{TO}) \quad (34)$$

Kurbel:

$$P_3 = - \sum_{i=1}^3 m_{AB} g \cdot l_{AB} \sin q_i \quad (35)$$

Parallelogramm:

$$P_4 = \sum_{i=1}^3 \left[m_{BC1} g \cdot \frac{1}{2}(z - l_a \sin q_i) - m_{BC2} g \cdot l_a \sin q_i + m_{BC2} g \cdot z \right] \quad (36)$$

Gesamtenergie:

$$\begin{aligned} P &= \sum_{n=1}^4 P_n \\ &= m_P g \cdot z + m_{T2} g \cdot (z + l_{TP}) + m_{T1} g \cdot \frac{1}{2}(z + l_{TP} + l_{TO}) \\ &\quad + \sum_{i=1}^3 \left[m_{BC1} g \cdot \frac{1}{2}(z - l_a \sin q_i) - m_{BC2} g \cdot l_a \sin q_i + m_{BC2} g \cdot z - m_{AB} g \cdot l_{AB} \sin q_i \right] \end{aligned} \quad (37)$$

3.1.2 Kinetische Energie

Die Formulierung der kinetischen Energie profitiert von der Vereinfachung durch die Massenäquivalente. So können die Massen m_{T2} und m_{BC2} and den jeweils unteren Enden der Stäbe, bzw. der Teleskopstange der Plattform zugeordnet werden.

Plattform:

$$K_1 = \frac{1}{2}(m_P + 3m_{BC2} + m_{T2})(\dot{x} \ \dot{y} \ \dot{z})^T{}^2 \quad (38)$$

Teleskopstange:

$$K_2 = \frac{1}{8}m_{T1}(\dot{x} \ \dot{y} \ \dot{z})^T{}^2 \quad (39)$$

Kurbel:

$$K_3 = \sum_{i=1}^3 \frac{1}{2}\Theta_{AB}\dot{q}_i^2 \quad (40)$$

Parallelogramm:

$$K_4 = \sum_{i=1}^3 \left[\frac{1}{2}m_{BC2}l_a^2\dot{q}_i^2 + \frac{1}{8}m_{BC1}(\mathbf{v}_{B,i} + \mathbf{v}_{C,i})^2 \right] \quad (41)$$

Ableiten der Gleichungen (5) bis (7) nach der Zeit liefert

$$\mathbf{v}_{B,i} = \begin{bmatrix} -l_a \sin q_i \cdot \dot{q}_i \\ 0 \\ -l_a \cos q_i \cdot \dot{q}_i \end{bmatrix} \quad (42)$$

und $\mathbf{v}_{C,i}$ ist gleich der Plattformgeschwindigkeit und damit analog zu (24) bis (26):

$$\mathbf{v}_{C,i} = \begin{bmatrix} \dot{x} \cos \theta_i + \dot{y} \sin \theta_i \\ -\dot{x} \sin \theta_i + \dot{y} \cos \theta_i \\ \dot{z} \end{bmatrix} \quad (43)$$

Gesamtenergie:

$$\begin{aligned} K &= \sum_{n=1}^4 K_n \\ &= \frac{1}{2}(m_P + m_{BC2} + m_{T2})(\dot{x} \ \dot{y} \ \dot{z})^T{}^2 + \frac{1}{8}m_{T1}(\dot{x} \ \dot{y} \ \dot{z})^T{}^2 \\ &\quad + \sum_{i=1}^3 \left[\frac{1}{2}\Theta_{AB}\dot{q}_i^2 + \frac{1}{2}m_{BC2}l_a^2\dot{q}_i^2 + \frac{1}{8}m_{BC1}(\mathbf{v}_{B,i} + \mathbf{v}_{C,i})^2 \right] \end{aligned} \quad (44)$$

3.1.3 Lösung der Dynamikgleichungen

Mit der Definition

$$\Lambda_j := \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\xi}_j} \right) - \frac{\partial L}{\partial \xi_j} \quad (45)$$

lässt sich über Gleichung (31) für $j = 1, \dots, 3$ ein Gleichungssystem unter Berücksichtigung von $\Xi_1 = \Xi_2 = \Xi_3 = 0$ (Keine Kräfteinleitung über diese Koordinaten) für die gesuchten Multiplikatoren aufstellen:

$$\begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial x} & \frac{\partial f_3}{\partial x} \\ \frac{\partial f_1}{\partial y} & \frac{\partial f_2}{\partial y} & \frac{\partial f_3}{\partial y} \\ \frac{\partial f_1}{\partial z} & \frac{\partial f_2}{\partial z} & \frac{\partial f_3}{\partial z} \end{bmatrix} \cdot \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} \quad (46)$$

Aus diesem System lassen sich die μ_i mittels LR-Zerlegung berechnen. Da $\frac{\partial f_i}{\partial \xi_j} = 0$ für $j \neq i + 3$ lassen sich dann über Gleichung (31) für $j = 4, \dots, 6$ die gesuchten Antriebsmomente τ_i errechnen:

$$\tau_1 = \Xi_4 = \Lambda_4 - \mu_1 \frac{\partial f_1}{\partial q_1} \quad (47)$$

$$\tau_2 = \Xi_5 = \Lambda_5 - \mu_2 \frac{\partial f_2}{\partial q_2} \quad (48)$$

$$\tau_3 = \Xi_6 = \Lambda_6 - \mu_3 \frac{\partial f_3}{\partial q_3} \quad (49)$$

Die für die Berechnung von Gleichung (46) und (49) nötigen Terme Λ_i und $\frac{\partial f_i}{\partial \xi_j}$ sind im Anhang B dokumentiert.

Teil III

Implementierung

Die Implementierung des Roboter-Modells erfolgte in der Programmiersprache C++. Die Berechnungen wurden in der Klasse `DeltaRobot` gekapselt, welche über den Quellcode dokumentiert ist. Das dynamische Modell ist in drei Reduktionsstufen verfügbar implementiert. So kann in der Stufe 1 die Teleskopstange unberücksichtigt bleiben, wenn ein Delta-Roboter ohne diese Stange betrieben wird. Ist die exakte Modellierung zu rechenzeitintensiv, kann die Reduktionsstufe 2 gewählt werden, in der die Masse m_{BC1} unberücksichtigt bleibt, was zu einem erheblich geringeren Rechenaufwand führt. Die Wahl der Reduktionsstufe kann der Funktion `calcDynamicModel(int reduction)` als Parameter übergeben werden. Dazu muss der Aufruf in der Funktion `computeTorques()` entsprechend angepasst werden. Die Parametrisierung (siehe Anhang D) erfolgt direkt in der Funktion `initParameters()` und kann später entsprechend erweitert oder angepasst werden.

In der Tabelle 1 sind Abschätzungen für den Berechnungsaufwand der einzelnen Implementierungsteile wiedergegeben. Sie können als Entscheidungshilfe dienen, in welchem Umfang das Modell in eine Steuerung integriert werden kann.

Berechnung	Funktionsaufrufe	Aufwand
Inverse Kinematik (nur $\dot{\mathbf{q}}$)	<code>calc_q()</code> <code>calc_q_dot()</code>	18 \angle , 78 \pm , 72 \times , 12 \div
Vollständige inverse Kinematik (Vorraussetzung für Dynamik)	<code>calcInvKinematic()</code>	18 \angle , 138 \pm , 150 \times , 15 \div
Momentenberechnung (inklusive Gleichungssystem lösen)	<code>computeTorques()</code> ohne <code>calcDynamicModel()</code>	28 \pm , 32 \times , 23 \div
Dynamisches Modell	<code>calcDynamicModel(0)</code>	84 \pm , 166 \times , 7 \div
Dynamisches Modell (reduziert)	<code>calcDynamicModel(2)</code>	46 \pm , 92 \times , 4 \div

Tabelle 1: Berechnungsaufwand der Implementierungsteile

4 Validierung

4.1 Numerisches MKS-Modell

Zur Validierung der Implementierung wurde mit dem SimMechanics-Werkzeugkasten für die Software MATLAB/Simulink ein numerisches Mehrkörper-Simulations-Modell des Delta-Roboters aufgebaut. Abbildung 7 zeigt das grafische Programm.

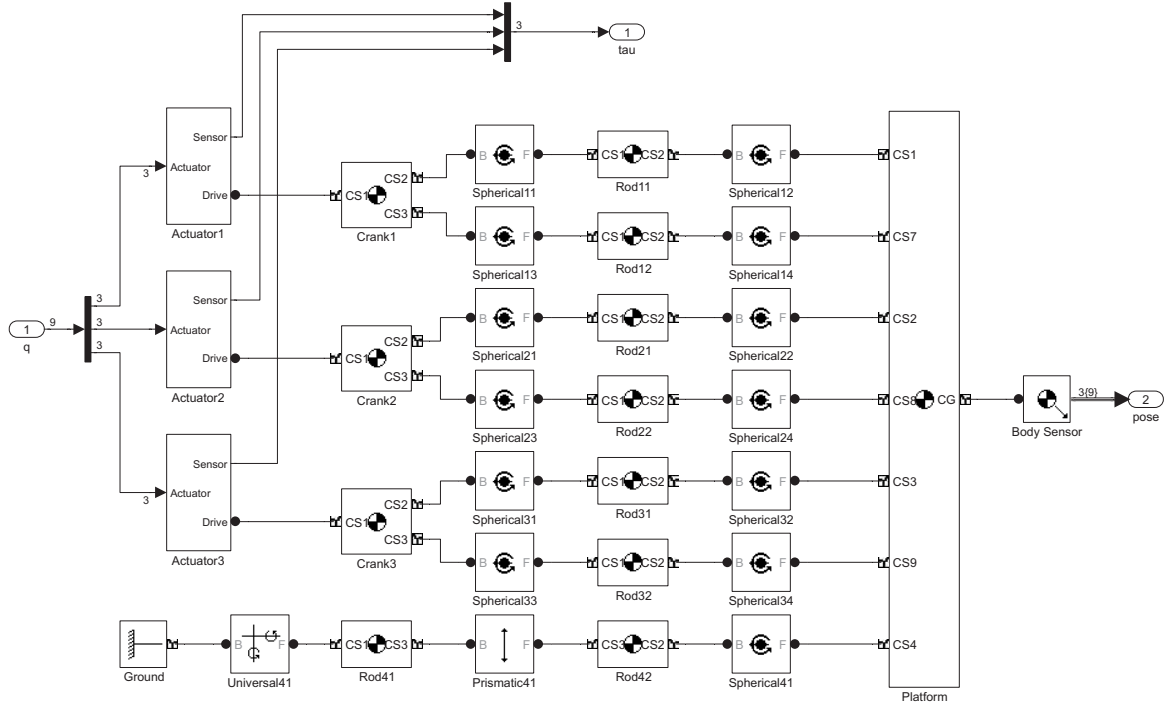


Abbildung 7: MKS-Modell des Delta-Roboters mit SimMechanics für MATLAB/Simulink

Die Einbindung der `DeltaRobot`-Klasse erfolgte über einen S-Function-Wrapper, der mit dem Borland C/C++-Compiler Version 5.5 übersetzt wurde. Die Testtrajektorie weist, um eventuelle Fehler in den einzelnen Termen des dynamischen Modells gut identifizieren zu können, recht hohe Geschwindigkeiten (bis zu $8 \frac{m}{s}$) und Beschleunigungen (bis $24 \frac{m}{s^2}$) auf. Sie ist durch die Funktionen

$$x(t) = 0,25m \cdot \sin\left(\frac{1}{2} \cdot \frac{(5-0,1) \cdot 2\pi}{10s^2} \cdot t^2 + 0,1 \cdot \frac{2\pi}{s} \cdot t\right) \quad (50)$$

$$y(t) = 0,25m \cdot \sin\left(\frac{1}{2} \cdot \frac{(0,1-5) \cdot 2\pi}{10s^2} \cdot t^2 + 5 \cdot \frac{2\pi}{s} \cdot t\right) \quad (51)$$

$$z(t) = 0,1m \cdot \sin\left(5 \cdot \frac{2\pi}{s} \cdot t + \frac{\pi}{2}\right) - 0,7556m \quad (52)$$

beschrieben. Abbildung 8 zeigt die Bahn im Raum, die so gewählt wurde, dass möglichst grosse Teile des Arbeitsraumes berücksichtigt werden.

Mit der gewählten Parametrisierung (siehe Anhang D) lässt sich dem analytischen Modell ein hervorragender Gleichlauf zum numerischen Modell attestieren, wie Abbildung 9 belegt. Die Ausgaben des numerischen Modells sind mit dem Index n gekennzeichnet. Als Fehlermaß werden die aufsummierten Beträge der Fehler der Einzel-Antriebsmomente gemäß

$$E(t) = \frac{\sum_{i=1}^3 |\tau_i(t) - \tau_{i,n}(t)|}{\max(\tau)}$$

gewählt. Da der Fehler E jederzeit unter 1 Promille bleibt, ist die Implementierung als fehlerfrei zu bewerten.

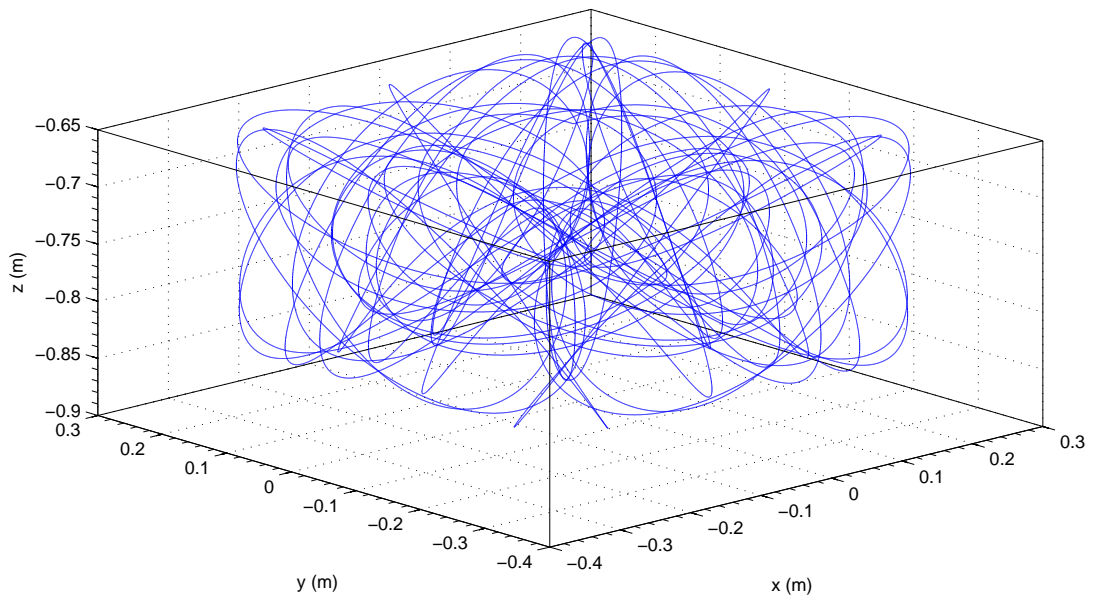


Abbildung 8: Testtrajektorie

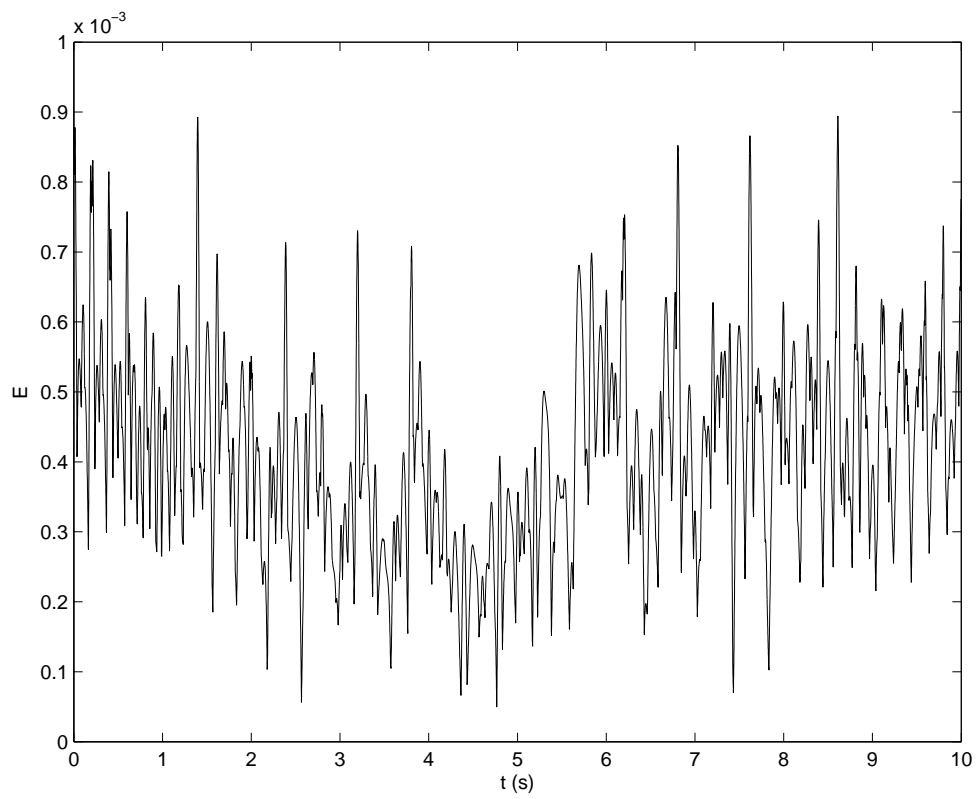


Abbildung 9: Relative Abweichung E zwischen den Modellen auf der Testtrajektorie

4.2 Simulationsergebnisse

Im folgenden werden Simulationsergebnisse zur Güte der reduzierten Modelle mit und ohne Nutzlast wiedergegeben. Für das vollständige Modell wurde zudem der Einfluss einer Fehlparametrisierung der Nutzlast simuliert.

4.2.1 Reduziertes Modell

Für die Reduktionsstufe 2, in der nur die Bewegung der Plattform, der Teleskopstange und der Kurbeln berücksichtigt sind, werden die errechneten Momente für den ersten Antrieb mit den tatsächlich benötigten verglichen. Das Resultat zeigen Abbildung 10 und 11, gemessen am geringeren Berechnungsaufwand ist ein guter Gleichlauf mit dem exakten Modell zu erkennen.

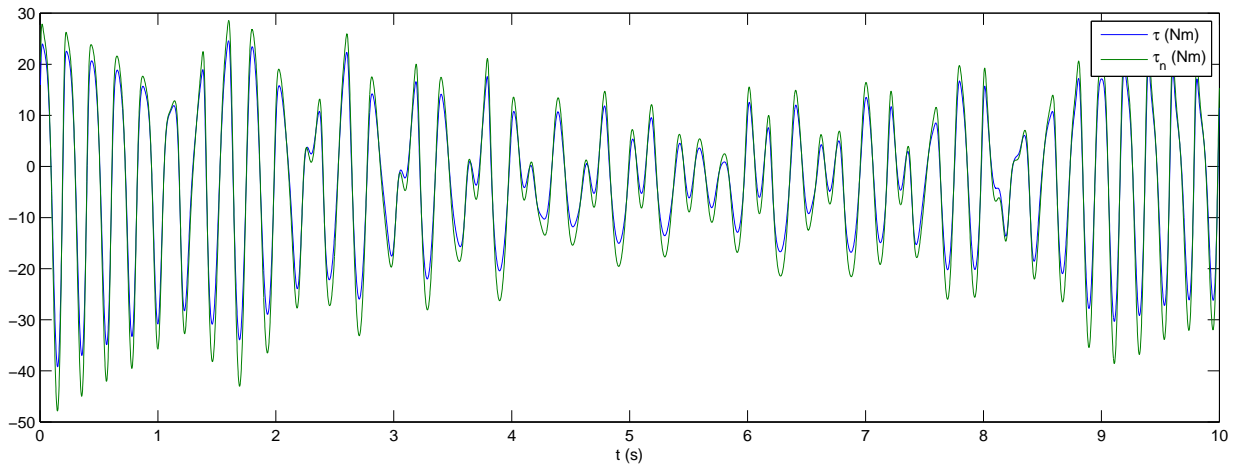


Abbildung 10: Vergleich der errechneten Momente für die Reduktionsstufe 2 ohne Nutzlast

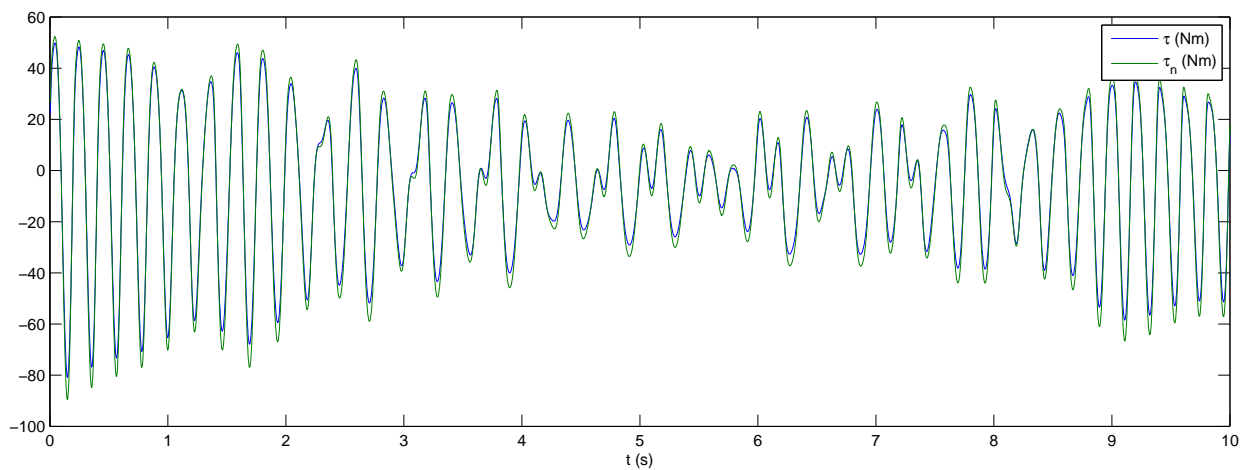


Abbildung 11: Vergleich der errechneten Momente für die Reduktionsstufe 2 mit einer Nutzlast von 500g

4.2.2 Fehlparametrisierung der Nutzlast

Für das vollständige Modell wurde bezüglich der Nutzlast statt mit 500g mit 450g Nutzlast fehlparametrisiert. Die Abbildungen 12 und 13 zeigen das Ergebnis.

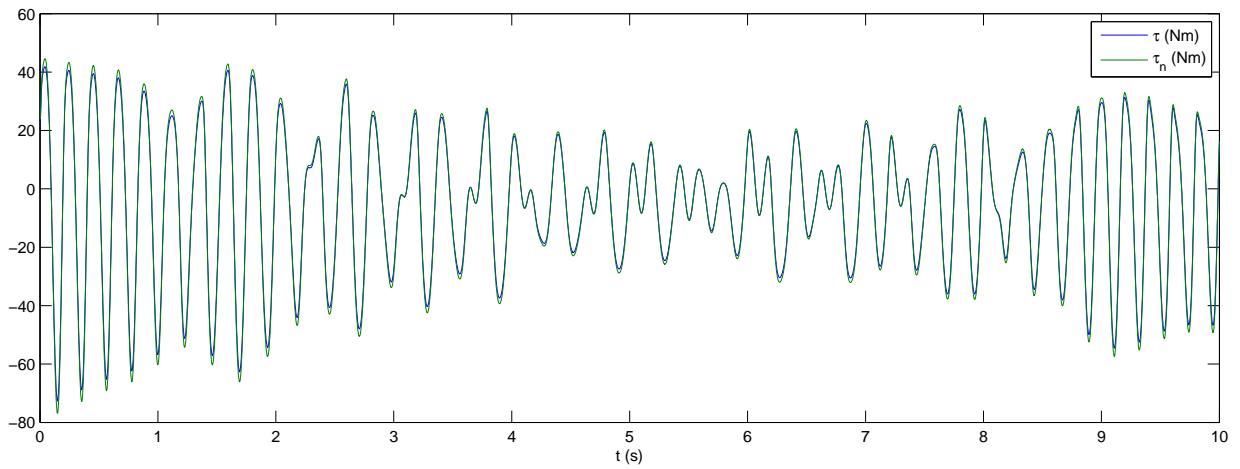


Abbildung 12: Einfluss einer Fehlparametrisierung der Nutzlast von -10%

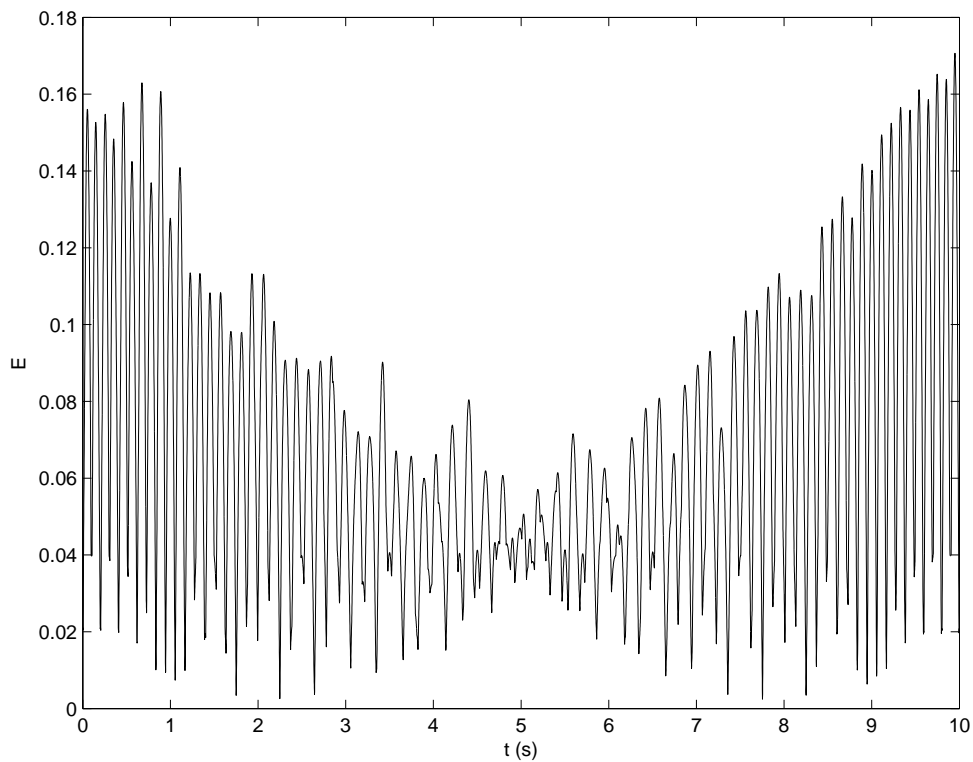


Abbildung 13: Relative Abweichung E bei einer Fehlparametrisierung der Nutzlast von -10%

Teil IV

Anhang

A Modellierung der vierten Achse

Die inverse Kinematik für die vierte Achse, die nicht mit den translatorischen Freiheitsgraden verkoppelt ist, kann über die Identität beschrieben werden:

$$\begin{bmatrix} q_4 & \dot{q}_4 & \ddot{q}_4 \end{bmatrix}^T = \mathbf{I}_{3 \times 3} \begin{bmatrix} \phi & \dot{\phi} & \ddot{\phi} \end{bmatrix}^T \quad (53)$$

Bei der inversen Dynamik ist nur die Rotation um ihre Längsachse zu berücksichtigen, da die translatorischen Bewegungen und die übrigen Rotationen über die Antriebe an den parallelen Ketten erzeugt werden. Folglich gilt:

$$\tau_4 = \Theta_{T\phi} \ddot{\phi} \quad (54)$$

mit $\Theta_{T\phi}$ als Trägheitsmoment des Antriebs, der Teleskopstange und des Greifers.

B Hilfsgrößen der dynamischen Modellierung

Größen $\frac{\partial f_i}{\partial \xi_j}$ für $i = 1, \dots, 3$:

$$\frac{\partial f_i}{\partial x} = 2x - 2r \cos \theta_i - 2l_a \cos q_i \cos \theta_i \quad (55)$$

$$\frac{\partial f_i}{\partial y} = 2y - 2r \sin \theta_i - 2l_a \cos q_i \sin \theta_i \quad (56)$$

$$\frac{\partial f_i}{\partial z} = 2z + 2l_a \sin q_i \quad (57)$$

$$\frac{\partial f_i}{\partial q_i} = 2l_a z \cos q_i + 2l_a y \sin q_i \sin \theta_i + 2l_a x \sin q_i \cos \theta_i - 2l_a r \sin q_i \quad (58)$$

Größen Λ_j :

$$\begin{aligned} \Lambda_1 &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} \\ &= \left\{ (4m_{T2} + m_{T1} + 4m_P + 4m_{BC2}) \ddot{x} + \left[3\ddot{x} + \left((-\ddot{q}_3 \sin q_3 - \dot{q}_3^2 \cos q_3) \cos \theta_3 \right. \right. \right. \\ &\quad \left. \left. \left. + (-\ddot{q}_2 \sin q_2 - \dot{q}_2^2 \cos q_2) \cos \theta_2 + (-\ddot{q}_1 \sin q_1 - \dot{q}_1^2 \cos q_1) \cos \theta_1 \right) l_a \right] m_{BC1} \right\} / 4 \quad (59) \end{aligned}$$

$$\begin{aligned} \Lambda_2 &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}} \right) - \frac{\partial L}{\partial y} \\ &= \left\{ (4m_{T2} + m_{T1} + 4m_P + 4m_{BC2}) \ddot{y} + \left[3\ddot{y} + \left((\ddot{q}_3 \sin q_3 + \dot{q}_3^2 \cos q_3) \sin \theta_3 \right. \right. \right. \\ &\quad \left. \left. \left. + (\ddot{q}_2 \sin q_2 + \dot{q}_2^2 \cos q_2) \sin \theta_2 + (\ddot{q}_1 \sin q_1 + \dot{q}_1^2 \cos q_1) \sin \theta_1 \right) l_a \right] m_{BC1} \right\} / 4 \quad (60) \end{aligned}$$

$$\begin{aligned} \Lambda_3 &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}} \right) - \frac{\partial L}{\partial z} \\ &= \left\{ (4\ddot{z} + 4g)m_{T2} + (\ddot{z} + 2g)m_{T1} + (4\ddot{z} + 4g)m_P + (12\ddot{z} + 12g)m_{BC2} \right. \\ &\quad \left. + \left[3\ddot{z} + (-\ddot{q}_3 \cos q_3 + \dot{q}_3^2 \sin q_3 - \ddot{q}_2 \cos q_2 + \dot{q}_2^2 \sin q_2 \right. \right. \\ &\quad \left. \left. - \ddot{q}_1 \cos q_1 + \dot{q}_1^2 \sin q_1 \right) l_a + 6g \right] m_{BC1} \right\} / 4 \quad (61) \end{aligned}$$

Für $i = 1, \dots, 3$:

$$\begin{aligned} \Lambda_{i+3} &= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \\ &= - \left[4gl_a m_{BC2} \cos q_i + l_a \ddot{z} m_{BC1} \cos q_i - l_a \ddot{y} m_{BC1} \sin q_i \sin \theta_1 + l_a \ddot{x} m_{BC1} \sin q_i \cos \theta_1 \right. \\ &\quad \left. - \ddot{q}_i l_a^2 m_{BC1} + 2gl_a m_{BC1} \cos q_i + 4gl_{AB} m_{AB} \cos q_i - 4\ddot{q}_i \Theta_{AB} \right] / 4 \quad (62) \end{aligned}$$

C Computeralgebra-Skripte

Die symbolischen Manipulationen wurden mit dem Computeralgebra-Paket MAXIMA in der Version 5.12.0 durchgeführt. Die Skripte zur Ausführung sind in den folgenden Abschnitten wiedergegeben.

Berechnung der Differentialkinematik:

```
kill(all);
depends(x,t,y,t,z,t,q,t);

((r+la*cos(q))*cos(the)-x)^2 +
((r+la*cos(q))*sin(the)-y)^2 +
(-la*sin(q)-z)^2 -lb^2;

expand(diff(%o2,t));
solve([%], [q]);
trigsimp(%);

expand(diff(diff(%o2,t),t));
solve([%], [q]);
trigsimp(%);
```

Berechnung der Elemente der Jacobi-Matrix:

```
kill(all);
depends(x,t,y,t,z,t,lam,t,gam,t,bet,t,q,t);

depends(lam,y);
depends(gam,z,gam,x);
depends(C_x,x);
lam(y) := asin(y/lb);
C_x(x) := x-rb;
gam(C_x,z) := atan(C_x/-z);

bet(C_x,gam,lam) := acos(
  (la^2 + (C_x /sin(gam))^2 - (lb*cos(lam))^2 )
  / ( 2*la* C_x / sin(gam))
);

bet(C_x(x),gam(C_x(x),z),lam(y));
bet(C_x(x),gam,lam);

diff(bet(C_x,gam,lam) ,x);
diff(bet(C_x,gam,lam) ,y);
diff(bet(C_x,gam,lam) ,z);
```

Berechnung der Lagrange-Funktion und ihrer Ableitungen:

```
kill(all);
depends(x,t,y,t,z,t,q,t);

P_1: m_P * g * z;

P_2: m_T2 * g * (z+l_TP) + m_T1 * g * 1/2*(z+l_T0+l_TP);

P_3: -sum( m_AB * g * l_AB * sin(q[i]) ,i,1,3);

P_4: sum(m_BC1*g*1/2*(z-l_a*sin(q[i]))-m_BC2*g*l_a*sin(q[i])+ m_BC2*g*z ,i,1,3);

P: P_1 + P_2 + P_3 + P_4;
```

```

v_B(i):= matrix( [ -l_a*sin(q[i])*('diff(q[i],t)) ],
[ 0 ],
[ -l_a*cos(q[i])*('diff(q[i],t)) ]]);

v_C(i):= matrix( [ ('diff(x,t))*cos(the[i]) + ('diff(y,t))*sin(the[i]) ],
[ -('diff(x,t))*sin(the[i]) + ('diff(y,t))*cos(the[i]) ],
[ ('diff(z,t)) ]);

v(i):= v_C(i) + v_B(i);

K_1: 1/2*(m_P+3*m_BC2)*('diff(x,t)^2+'diff(y,t)^2+'diff(z,t)^2);

K_2: (1/8*m_T1 + 1/2*m_T2)*('diff(x,t)^2+'diff(y,t)^2+'diff(z,t)^2);

K_3: sum(1/2*The*'diff(q[i],t)^2,i,1,3);

K_4: sum(1/2*m_BC2*l_a^2*'diff(q[i],t)^2 + 1/8*m_BC1*v(i).v(i) ,i,1,3);

K: K_1 + K_2 + K_3 + K_4;

L: K-P;

L: trigsimp(%);

ratsimp(diff(diff(L,'diff(x,t)),t) - diff(L,x));
ratsimp(diff(diff(L,'diff(y,t)),t) - diff(L,y));
ratsimp(diff(diff(L,'diff(z,t)),t) - diff(L,z));

factor(diff(diff(L,'diff(q[1],t)),t) - diff(L,q[1]));
factor(diff(diff(L,'diff(q[2],t)),t) - diff(L,q[2]));
factor(diff(diff(L,'diff(q[3],t)),t) - diff(L,q[3]));

```

Ableitungen der kinematischen Zwangsbedingungen:

```

kill(all);
depends(x,t,y,t,z,t,q,t);

f(i):= ((r+l_a*cos(q[i]))*cos(the[i])-x)^2 +
((r+l_a*cos(q[i]))*sin(the[i])-y)^2 +
(-l_a*sin(q[i])-z)^2 -l_b^2;

trigsimp(diff(f(i),x));
trigsimp(diff(f(i),y));
trigsimp(diff(f(i),z));
trigsimp(diff(f(i),q[i]));

```

D Parametrisierung des dynamischen Modells

Die Parametrisierung des dynamischen Modells wurde wie folgt ermittelt:

m_{BC}	$= 71,477 \text{ g}$	(aus CAD-Daten)
Θ_{BC}	$= 6,5928 \text{ gm}^2$	(aus CAD-Daten)
m_{BC2}	$= 0,033299 \text{ kg}$	(nach Glg. 27)
m_{BC1}	$= 0,10965 \text{ kg}$	(nach Glg. 28)
m_P	$= 0,13107 \text{ kg}$	(aus CAD-Daten, Nutzlast noch nicht addiert)
m_T	$= 0,42140 \text{ kg}$	(aus CAD-Daten)
Θ_T	$= 20,445 \text{ gm}^2$	(aus CAD-Daten)
$\Theta_{T\phi}$	$= 0,7144 \text{ gm}^2$	(aus CAD-Daten, Motorträgheit und Nutzlast noch nicht addiert)
m_{AB}	$= 0.9109892\text{kg}$	(aus CAD-Daten)
l_{AB}	$= 0.1069968\text{m}$	(aus CAD-Daten)
Θ_{AB}	$= m_{AB} \cdot l_{AB}^2$	(Motorträgheit noch nicht addiert)