3. Praktikum

Bewegungsplanung

Im zweiten Praktikum werden Kinematikberechnungen und Bewegungsplanungen für einen dreiachsigen Industrieroboter behandelt.
Eine klassische Aufgabe ist es, Gegenstände mit einem am Endpunkt des Roboters montierten Greifer zu transportieren. Dabei ist zunächst lediglich die Anfangslage sowie das Ziel des Transportguts und bei geeigneter Greifstrategie die zugehörige Position des Endpunktes des Roboters in Weltkoordinaten ${}_I\boldsymbol{r}_E = (x_E, y_E, z_E)^T$ bekannt. Die Bewegung die der Roboter dazwischen durchführen soll, ist jedoch völlig frei, und kann in der Bahnplanung günstig gewählt werden. Die Bahnplanung ist klassischerweise in zwei Stufen aufgebaut. Zunächst plant sie einen geometrischen Pfad $\boldsymbol{q}(\sigma)$ oder $\boldsymbol{r}_E(\sigma)$, welche den Roboter mit dem Bahnparameter $\sigma \in [0, \sigma_E]$ kollisionsfrei, auf möglichst kurzem Weg, von Anfang zum Ziel führt. Der zweite Schritt ist es, den Verlauf $\sigma(t)$ und damit die Trajektorie $\boldsymbol{q}(t) = \boldsymbol{q}(\sigma(t))$ bzw. ${}_I\boldsymbol{r}_E(t)$ so zu bestimmen, dass der Roboter mit seinen physikalischen Grenzen dieser Bahn auch folgen kann. Je nach Aufgabe und Umgebung des Roboters ist es hin und wieder geschickter die Bewegung in Weltkoordinaten ${}_I\boldsymbol{r}_E(t)$ zu planen und sie anschließend auf Gelenkwinkel $\boldsymbol{q}({}_I\boldsymbol{r}_E(t))$ zu übersetzen, um sie der Regelung übergeben zu können. Diese Beziehung $\boldsymbol{q}({}_I\boldsymbol{r}_E)$ wird als Inverskinematik bezeichnet. Im Gegensatz zur Vorwärtskinematik ${}_I\boldsymbol{r}_E(\boldsymbol{q})$ ist die Bestimmung der Inverskinematik bei seriellen Robotern sehr herausfordernd. Der Aufbau von Robotern wird typischerweise gezielt so gestaltet, sodass diese Beziehung zwar Mehrfachlösungen aufweist (es gibt mehrere Konfigurationen $\boldsymbol{q}$, die dieselbe Endpunktposition ${}_I\boldsymbol{r}_E$ ergeben), jedoch an sich analytisch aufgelöst werden kann.

3.1 Vorwärtskinematik

Unter der Vorwärtskinematik versteht man die Aufgabe, zu einem Satz von Minimalkoordinaten $\boldsymbol{q}$ die zugehörige Lage des Endeffektors (z.B. Greifer oder Werkzeug) in Weltkoordinaten zu bestimmen, also ${}_I\boldsymbol{r}_E, \boldsymbol{\varphi}_E$. Im Praktikum werden lediglich die ersten drei Freiheitsgrade des Roboters $\boldsymbol{q} = (q_1, q_2, q_3)^T$ behandelt und die Orientierung $\boldsymbol{\varphi}_E$ des Endeffektors nicht weiter berücksichtigt.

Aufgabe 3.1 *Berechnen Sie die Vorwärtskinematik des Laborroboters nach Abbildung 3.1. Erstellen Sie eine Funktion „VorKin“ mit Hilfe des SimCode C-Package in Matlab/Simulink und testen Sie den Block mit verschiedenen Winkeln.*

Hinweis 3.1 *Bei der Codegenerierung ist zu beachten, dass der so erzeugte Block keinen dynamischen Zustand, sondern lediglich Eingänge und Ausgänge besitzt.*

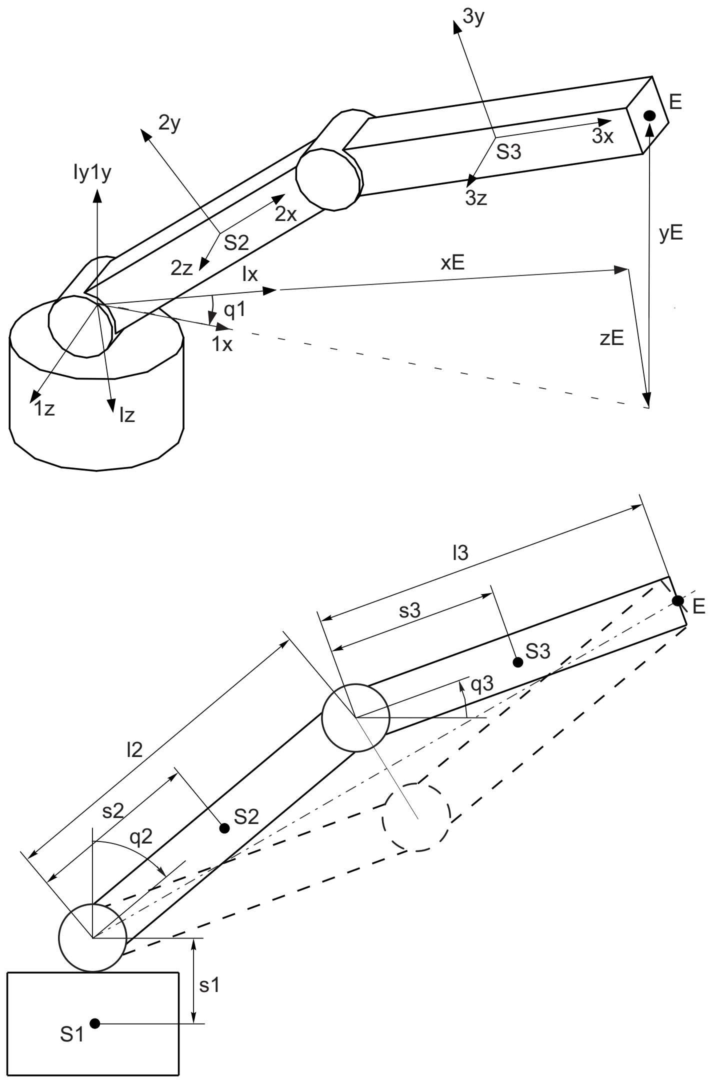


Abbildung 3.1: CRS CataLyst-5 Roboter - Freiheitsgrade

3.2 Inverskinematik

Die Inverskinematik stellt das inverse Problem zur Vorwärtskinematik dar. Eine Lösungsmöglichkeit ist es, die bei der Vorwärtskinematik entstandenen Gleichungen nach den Minimalkoordinaten aufzulösen. Dieser analytische Weg ist jedoch oft der schwierigere. Alternativ kann die Inverskinematik direkt mit geometrischen Überlegungen ausgehend von Abb. 3.1 aufgelöst werden.

Aufgabe 3.2 *Berechnen Sie die Inverskinematik des Laborroboters allgemein in Maple. Berücksichtigen Sie dabei beide möglichen Stellungen, die in Abb. 3.1 durchgezogen und strichliert dargestellt sind. Erstellen Sie wieder eine Funktion (SimCode C-Package) „InvKin“ in Matlab/Simulink, wobei die Wahl der „Up“ oder „Down“ Konfiguration über einen Parameter oder Eingang verändert werden kann. Testen Sie den Block für verschiedene Weltkoordinaten.*

Hinweis 3.2 *Sehen Sie sich für die Berechnung von q_1 die Dokumentation der Maple-Funktion atan2 an (Befehl* `?atan2`*) und machen Sie sich zur Berechnung der Armkinematik mit dem* Kosinussatz *vertraut.*

Aufgabe 3.3 *Testen Sie die Blöcke „VorKin“ und „InvKin“ in Serienschaltung, indem Sie für alle 3 Winkel eine Erregung von*

$$q_1 = q_2 = q_3 = 30°\frac{\pi}{180}\sin(t)$$

auf die Vorwärtskinematik, und die so berechneten Weltkoordinaten (x_E, y_E, z_E) anschließend der Inverskinematik aufschalten. Sie müssen wieder die ursprünglichen Winkel erhalten.

3.3 Bahnplanung

Eine in der Industrie häufig zu findende, wenn auch nicht optimale Bahn ist in Abb. 3.2 dargestellt. Um von einem Startwert 0 aus den Endwert $\Delta\sigma$ zu erreichen, wird bei dieser Bahn mittels eines symmetrischen Rechteckverlaufes beschleunigt bzw. verzögert. Die Geschwindigkeit stellt sich dadurch dreiecksförmig ein. Da bei diesem Bahnverlauf immer mit maximaler Beschleunigung a_{max} beschleunigt bzw. verzögert wird, muss die Dauer der Beschleunigungsphasen T_a so gewählt werden, dass die gewünschte Strecke zurückgelegt wird.

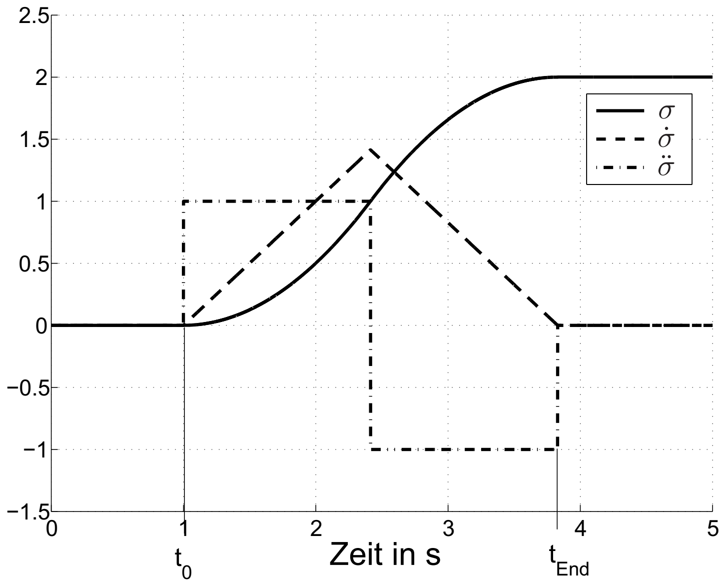


Abbildung 3.2: – Position, - - Geschwindigkeit und -·- Beschleunigung der Dreiecksbahn

Aufgabe 3.4 *Berechnen Sie T_a. Implementieren Sie anschließend einen Bahngenerator in Matlab/Simulink. Der Bahngenerator soll als Eingang die aktuelle Zeit t, die Startzeit der Beschleunigungsphase t_0, den Startpunkt σ_0, die maximale Beschleunigung a_{max} und die Wegdifferenz $\Delta\sigma$ besitzen. Die Ausgänge sind die aktuelle Position σ, die Geschwindigkeit $\dot{\sigma}$ und die Beschleunigung $\ddot{\sigma}$ sowie die Trajektorienendzeit t_{End} (zu der $\Delta\sigma$ erreicht wird).*

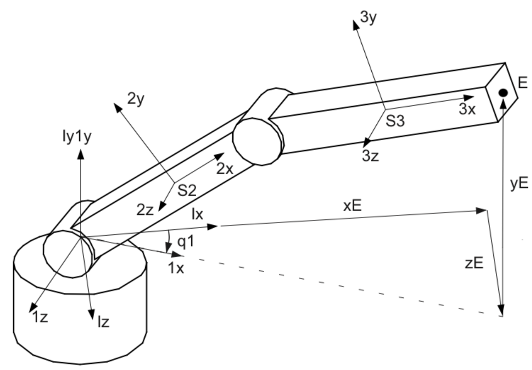
Hinweis 3.3 *Eine stückweise definierte Funktion kann in Maple über den Befehl* `piecewise` *definiert werden. Außerdem kann der Befehl* `assume` *bei der Vereinfachung von Integralen nützlich sein. Die* `plot` *Funktion ist für diese Aufgabe ebenfalls sehr hilfreich.*

Hinweis 3.4 *Verwenden Sie zur Implementierung des Bahngenerators eine MATLAB Function (Simulink Library Browser: User-Defined Functions/MATLAB Function). Die Beispieldatei* sim_bahn_dreieck.mdl *beinhaltet eine (leere) Vorlage.*

Aufgabe 3.5 *Verwenden Sie den Bahngenerator um eine Bahn vom Punkt* 0.5 *zum Punkt* 0.2 *zu planen.*

3.1 Vorwärtskinematik

Unter der Vorwärtskinematik versteht man die Aufgabe, zu einem Satz von Minimalkoordinaten $\boldsymbol{q}$ die zugehörige Lage des Endeffektors (z.B. Greifer oder Werkzeug) in Weltkoordinaten zu bestimmen, also ${}_I\boldsymbol{r}_E, \varphi_E$. Im Praktikum werden lediglich die ersten drei Freiheitsgrade des Roboters $\boldsymbol{q} = (q_1, q_2, q_3)^T$ behandelt und die Orientierung φ_E des Endeffektors nicht weiter berücksichtigt.

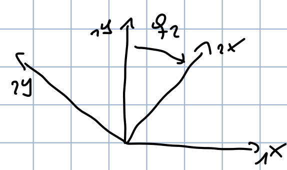


Aufgabe 3.1 *Berechnen Sie die Vorwärtskinematik des Laborroboters nach Abbildung 3.1. Erstellen Sie eine Funktion „VorKin“ mit Hilfe des SimCode C-Package in Matlab/Simulink und testen Sie den Block mit verschiedenen Winkeln.*

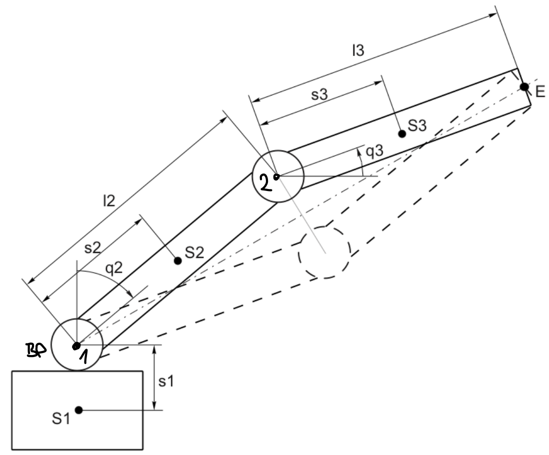
Hinweis 3.1 *Bei der Codegenerierung ist zu beachten, dass der so erzeugte Block keinen dynamischen Zustand, sondern lediglich Eingänge und Ausgänge besitzt.*

$$R_{1I} = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 \\ 0 & 1 & 0 \\ -\sin q_1 & 0 & \cos q_1 \end{bmatrix}$$

$$R_{21} = \begin{bmatrix} \sin q_2 & \cos q_2 & 0 \\ -\cos q_2 & \sin q_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$R_{31} = \begin{bmatrix} \cos q_2 & \sin q_2 & 0 \\ -\sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{aligned} {}_E\mathbf{r}_{0E} &= \sum_{j=1}^{E} [\mathbf{R}_{Ep}\ {}_p\mathbf{r}_{pj}]|_{p=p(j)} = \mathbf{R}_{E0}\ {}_0\mathbf{r}_{01} + \mathbf{R}_{E1}\ {}_1\mathbf{r}_{12} + \mathbf{R}_{E2}\ {}_2\mathbf{r}_{23} + \dots \\ &= ..\Big(\mathbf{R}_{32}\big(\mathbf{R}_{21}\left(\mathbf{R}_{10}\ {}_0\mathbf{r}_{01} + {}_1\mathbf{r}_{12}\right) + {}_2\mathbf{r}_{23}\big) + {}_3\mathbf{r}_{34}\Big).. \\ \mathbf{R}_{0E} &= \prod_{j=1}^{E} \mathbf{R}_{pj}|_{p=p(j)} = \mathbf{R}_{01}\mathbf{R}_{12}\mathbf{R}_{23}\mathbf{R}_{34}\dots \end{aligned}$$

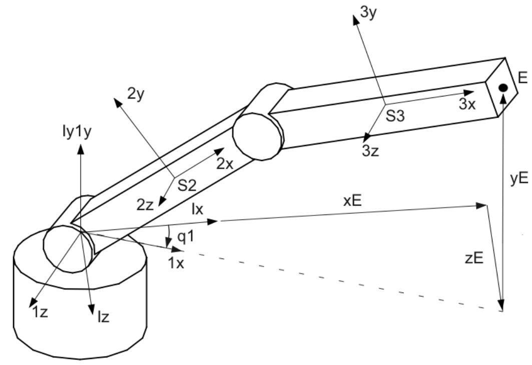
$${}_2r_{12} = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \quad {}_3r_{2E} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix} \quad {}_2r_{AS2} = \begin{bmatrix} s_2 \\ 0 \\ 0 \end{bmatrix} \quad {}_3r_{BS3} = \begin{bmatrix} s_3 \\ 0 \\ 0 \end{bmatrix} \quad {}_3r_{S3E} = \begin{bmatrix} l_3 - s_3 \\ 0 \\ 0 \end{bmatrix}$$

$${}_Ir_E = R_{I2}\,{}_2r_{12} + R_{I3}\,{}_3r_{2E}$$

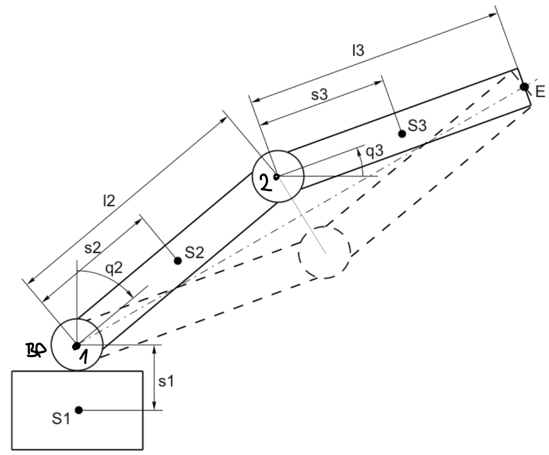
$${}_Er_E = {}_3r_E = R_{32}\,{}_2r_{12} + {}_3r_{2E}$$

3.2 Inverskinematik

Die Inverskinematik stellt das inverse Problem zur Vorwärtskinematik dar. Eine Lösungsmöglichkeit ist es, die bei der Vorwärtskinematik entstandenen Gleichungen nach den Minimalkoordinaten aufzulösen. Dieser analytische Weg ist jedoch oft der schwierigere. Alternativ kann die Inverskinematik direkt mit geometrischen Überlegungen ausgehend von Abb. 3.1 aufgelöst werden.

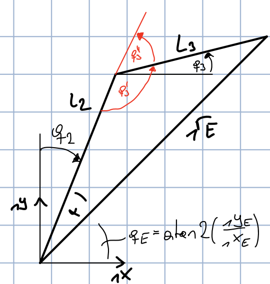


Aufgabe 3.2 *Berechnen Sie die Inverskinematik des Laborroboters allgemein in Maple. Berücksichtigen Sie dabei beide möglichen Stellungen, die in Abb. 3.1 durchgezogen und strichliert dargestellt sind. Erstellen Sie wieder eine Funktion (SimCode C-Package) „InvKin“ in Matlab/Simulink, wobei die Wahl der „Up“ oder „Down“ Konfiguration über einen Parameter oder Eingang verändert werden kann. Testen Sie den Block für verschiedene Weltkoordinaten.*



$$\varphi_1 = \operatorname{atan2}\left(\frac{z_E}{x_E}\right)$$

up



$$\varphi_3' = \arccos\left(\frac{l_2^2 + l_3^2 - r_E^2}{2 l_2 l_3}\right)$$

$$\frac{r_E}{\sin\varphi_3'} = \frac{l_3}{\sin\alpha} \rightarrow \alpha = \arcsin\left(\frac{l_3}{r_E}\sin\varphi_3'\right)$$

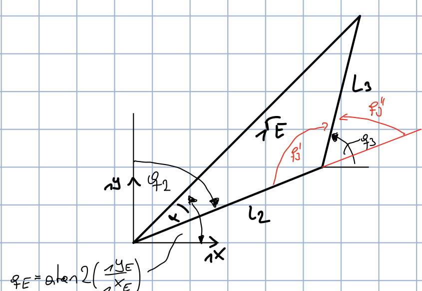
$$\varphi_2 = \frac{\pi}{2} - \varphi_E - \alpha$$

$$\varphi_3'' = \pi - \varphi_3'$$

$$\varphi_3 = \frac{\pi}{2} - \varphi_2 - \varphi_3''$$

down:

$\varphi_3', \varphi_3'', \alpha$ gleich



$$\varphi_2 = \frac{\pi}{2} - \varphi_E + \alpha$$

$$\varphi_3'' = \pi - \varphi_3'$$

$$\varphi_3 = \frac{\pi}{2} - \varphi_2 + \varphi_3''$$

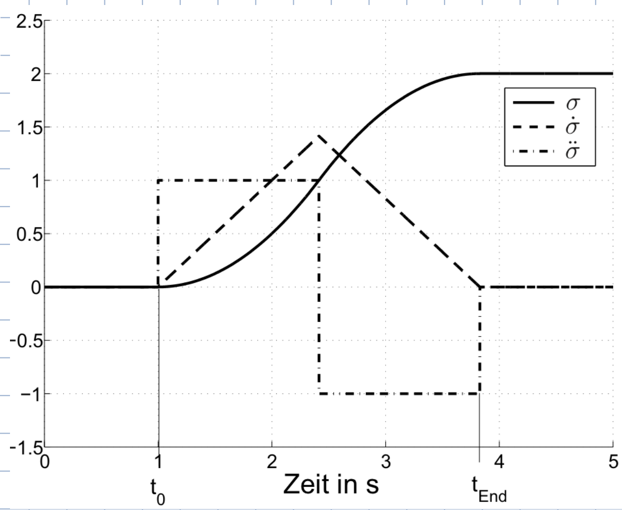
Ergebnis:

$$\varphi_1 = \operatorname{atan2}\left(\frac{z_E}{x_E}\right)$$

$$\varphi_2 = \frac{\pi}{2} - \varphi_E \pm \alpha \quad \begin{cases} \text{"+" für down} \\ \text{"-" für up} \end{cases}$$

$$\varphi_3 = \frac{\pi}{2} - \varphi_2 \mp \varphi_3'' \quad \begin{cases} \text{"+" für down} \\ \text{"-" für up} \end{cases}$$

Aufgabe 3.4 *Berechnen Sie T_a. Implementieren Sie anschließend einen Bahngenerator in Matlab/Simulink. Der Bahngenerator soll als Eingang die aktuelle Zeit t, die Startzeit der Beschleunigungsphase t_0, den Startpunkt σ_0, die maximale Beschleunigung a_{max} und die Wegdifferenz $\Delta\sigma$ besitzen. Die Ausgänge sind die aktuelle Position σ, die Geschwindigkeit $\dot{\sigma}$ und die Beschleunigung $\ddot{\sigma}$ sowie die Trajektorienendzeit t_{End} (zu der $\Delta\sigma$ erreicht wird).*



$$a = \begin{cases} 0 & t < t_0 \\ a_{max} & t_0 \le t < t_0 + \frac{T_a}{2} \\ -a_{max} & t_0 + \frac{T_a}{2} \le t < t_0 + T_a \\ 0 & t_0 + T_a \le t \end{cases}$$

$$v = \int a\,dt = at$$

$$v_{/} = a_{max}\left(t_0 + \frac{T_a}{2} - t_0\right) = a_{max}\frac{T_a}{2}$$

$$v_{\backslash} = -a_{max}\left(t_0 + T_a - t_0 - \frac{T_a}{2}\right) = -a_{max}\frac{T_a}{2}$$

$$s = \int v\,dt = a_{max}\frac{t^2}{2}$$

$$s_{/} = \frac{a_{max}}{2}\left[\left(t_0 + \frac{T_a}{2}\right)^2 - t_0^2\right] = \frac{a_{max}}{2}\left[\cancel{t_0^2} + t_0T_a + \frac{T_a^2}{4} - \cancel{t_0^2}\right] = \frac{a_{max}}{2}\left(t_0T_a + \frac{T_a^2}{4}\right)$$

$$s_{\backslash} = -\frac{a_{max}}{2}\left[(t_0 + T_a)^2 - \left(t_0 + \frac{T_a}{2}\right)^2\right] = -\frac{a_{max}}{2}\left[\cancel{t_0^2} + \cancel{2}t_0T_a - T_a^2 - \cancel{t_0^2} - \cancel{t_0T_a} - \frac{T_a^2}{4}\right] = -\frac{a_{max}}{2}\left(t_0T_a - \frac{3}{4}T_a^2\right)$$

$$s_{ges} = s_{/} + s_{\backslash} = \frac{a_{max}}{2}\left(\cancel{t_0T_a} + \frac{T_a^2}{4}\right) - \frac{a_{max}}{2}\left(\cancel{t_0T_a} - \frac{3}{4}T_a^2\right) = \frac{a_{max}}{2}\frac{T_a^2}{4}$$