Tarzan-Plotter

Das Gerät heisst Tarzan-Plotter, weil sich der Plotkopf der beiden Halteseile entlanghangelt, wie Tarzan an zwei Lianen. Deshalb heissen die Seile bei mir „Lianen".

Verwendete Variablen

$\vec{e}_x$	ist der Einheitsvektor in Richtung der x-Koordinate des ortsfesten Koordinatensystems
$\vec{e}_y$	ist der Einheitsvektor in Richtung der y-Koordinate des ortsfesten Koordinatensystems
$\vec{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$	„Plotpunkt", der Vektor vom Ursprung des ortsfesten Koordinatensystems zum Punkt, an dem die Zeichenspitze die Zeichenfläche berührt. p_x ist die x-, p_y die y-Komponente dieses Vektors.
L	„Basislänge", derAbstand zwischen den Achsen der beiden Antriebsmotoren
$L\vec{e}_x = \begin{pmatrix} L \\ 0 \end{pmatrix}$	„Plotbasis", der Vektor von der Achse des linken zu der des rechten Antriebsmotors.
$\vec{m}_1$	Vektor zur Achse des linken Antriebsmotor
l_1	Länge der linken Liane
$\vec{e}_1$	Einheitsvektor in Richtung von der Achse des linken Antriebsmotors zum Plotpunkt
$\vec{m}_2$	Vektor zur Achse des rechten Antriebsmotor
l_e	Länge der rechten Liane
$\vec{e}_2$	Einheitsvektor in Richtung vom Plotpunkt zur Achse des rechten Antriebsmotors
H , B	Höhe bzw, Breite der Fläche, in der der Plotter mit der gewünschten Genauigkeit arbeitet („Plotfläche"). Die obere, horizontale Seite dieser Fläche liegt etwas unterhalb der Plotbasis (abhängig von der maximal zulässigen Zugspannung der Lianen). Die Breite ist geringer als die Länge der Plotbasis (abhängig vom maximal zulässigen Durchhang der längeren Liane).
F	„Plotmassstab", der Faktor, um den beim Plotten alle Längen im Originalbild verkleinert oder vergrössert werden sollen.
$\vec{p}_M = \begin{pmatrix} 0 \\ p_{M,y} \end{pmatrix}$	Mittelpunkt der Plotfläche. Weil die Plotfläche unterhalb der Plotbasis liegt, ist $p_{M,y} < 0$.
$\vec{r} = \begin{pmatrix} x \\ y \end{pmatrix}$	Vektor zu einem Punkt des Originalbildes, gemessen im dessen Koordinatensystem.
M	Masse des Plotterkopfes
μ_1 , μ_2	Masse pro Längeneinheit der linken, bzw. der rechten Liane
$\vec{s}_1$, $\vec{s}_2$	Vektoren zu beliebigen Punkten entlang der „Seele" der linken bzw. rechten Liane, gemessen im Koordinatensystem des Plotters.

1. Einen Punkt plotten

Der Ursprung des Plotter-Koordinatensystems soll in der Mitte zwischen den Achsen der Antriebsmotoren liegen. Der Einheitsvektor von der Achse des linken zu der des rechten Antriebsmotors ist gleichzeitig der Einheitsvektor in Richtung der x-Achse des Plotter-Koordinatensystems. Die Vektor zu den Achsen der beiden Antriebsmotoren sind deshalb

$$\vec{m}_1 = -\frac{L}{2}\vec{e}_x \quad \text{und} \quad \vec{m}_2 = \frac{L}{2}\vec{e}_x \quad .$$

Das drückt auch aus, dass

$$\vec{m}_1 + \vec{m}_2 = 0$$

sich beide Motoren gleich weit und in entgegengesetzen Richtungen vom Ursprung entfernt sind. Andererseits bilden sie gemeinsam die Plotbasis

$$-\vec{m}_1 + \vec{m}_2 = L\vec{e}_x \quad .$$

Der Durchmesser der Zahnscheiben an beiden Motoren werden vernachlässigt, weil sie viel kleiner sind, als die Plotbasis. Genauso wird der Durchmesser des Plotterkopfes vernachlässigt. Der Vektor zum aktuellen Plotpunkt lässt sich damit auf zwei verschiedene Weisen darstellen, nämlich durch

(1_1) $$\vec{p} = \vec{m}_1 + l_1\vec{e}_1 = -\frac{L}{2}\vec{e}_x + l_1\vec{e}_1$$

und durch

(1_2) $$\vec{p} = \vec{m}_2 - l_2\vec{e}_2 = \frac{L}{2}\vec{e}_x - l_2\vec{e}_2 \quad .$$

Einmal links herum, vom linken Antriebsmotor, und einmal rechts herum, vom rechten Antriebsmotor aus.

So umgestellt, dass die gesuchten Grössen auf der linken Seite stehen, sehen die Gleichungen folgendermassen aus

$$l_1\vec{e}_1 = \vec{p} + \frac{L}{2}\vec{e}_x$$
$$l_2\vec{e}_2 = -\vec{p} + \frac{L}{2}\vec{e}_x$$

Durch Quadrieren verschwinden die Einheitsvektoren $\vec{e}_1$ und $\vec{e}_2$,

(1_3) $$l_1^2 = \left(\vec{p} + \frac{L}{2}\vec{e}_x\right)^2 = \left(p_x + \frac{L}{2}\right)^2 + p_y^2$$

(1_4) $$l_2^2 = \left(-\vec{p} + \frac{L}{2}\vec{e}_x\right)^2 = \left(p_x - \frac{L}{2}\right)^2 + p_y^2$$

Mit diesen Formeln kann man für jeden Punkt unterhalb der Plotbasis ($p_y < 0$) und im Intervall $-\frac{L}{2} < p_x < +\frac{L}{2}$ die zugehörigen Lianenlängen ausrechnen. Das Vorzeichen von l_1 und l_2 ist immer positiv.

2. Eine Gerade plotten

Angenommen, eine Gerade vom Punkt $\vec{p}_0$ zum Punkt $\vec{p}_1$ soll gezeichnet werden. Beide Punkte sollen in dem Bereich liegen, der dem Plotter zugänglich ist.

Die Gerade hat die Länge l

$$l = |\vec{p}_1 - \vec{p}_0|$$

und wird mit Hilfe des Richtungsvektor $\vec{q}$ der Geraden

$$\vec{q} = \vec{p}_1 - \vec{p}_0 = \begin{pmatrix} p_{1,x} - p_{0,x} \\ p_{1,y} - p_{0,y} \end{pmatrix}$$

in der Form

$$\vec{p}(s) = \vec{p}_0 + s \cdot \vec{q} = \begin{pmatrix} p_{0,x} + s\,q_x \\ p_{0,y} + s\,q_y \end{pmatrix} = \begin{pmatrix} p_x(s) \\ p_y(s) \end{pmatrix}$$

dargestellt. s ist die Laufvariable, mit der die Reihe der Punkte auf der Geraden dargestellt wird, angefangen mit

$\vec{p}(s=0) = \vec{p}_0$ und endend bei $\vec{p}(s=l) = \vec{p}_1$.

Die Lianenlängen l_1 und l_2 für jeden Punkt auf der Geraden werden jetzt auch zu Funktionen von s :

(2_1) $$l_1(s)^2 = \left(x_0 + s\,q_x + \frac{L}{2}\right)^2 + (y_0 + s\,q_y)^2$$

(2_2) $$l_2(s)^2 = \left(x_0 + s\,q_x - \frac{L}{2}\right)^2 + (y_0 + s\,q_y)^2 \; .$$

Mit diesen Formeln kann man die Gerade Punkt für Punkt in jeder gewünschten Auflösung plotten.

Umsetzung in ein Steuerprogramm (eine mögliche Variante)

Angenommen wird, dass ein Steuerrechner periodisch (z.B. Timer-gesteuert) die Lianenlängen berechnet, die zum Anfahren des nächsten (Zwischen-)Punktes erforderlich sind. Diese werden dann in Schrittzahlen und Drehrichtungen für die beiden Motoren umgerechnet und an die Steuerelektronik der Motoren weitergegeben.

Der Steuerrechner verwendet folgende Variablen:

P0_X, P0_Y	Koordinaten des Anfangspunktes der aktuellen Geraden
Q_X, Q_Y	Koordinaten aktuellen Geraden-Richtungsvektors
S_MAX	Maximalwert der Laufvariablen (Minimalwert ist Null). Zweckmässigerweise wird er so gewählt, dass er die Anzahl Zwischenpunkte angibt, die entlang der Geraden angefahren werden sollen. Wenn die Gerade 10cm lang werden soll, und alle Zentimeter ein Punkt angefahren werden soll, dann wäre SMAX := 10.
S_IST	aktueller Wert der Laufvariablen
LL1_SOLL	neu berechnete Länge der linken Liane
LL2_SOLL	dasselbe für die rechte Liane

für die kumulierten Schrittzahlen beider Schrittmotoren
NSCHRITT1_IST aktuelle kumumlierte Schrittzahl des linken Motors
NSCHRITT2_IST dasselbe für den rechten Motor

für die neu berechneten kumulierten Schrittzahlen beider Schrittmotoren
NSCHRITT1_SOLL neu berechnete, kumumlierte Schrittzahl des linken Motors
NSCHRITT2_SOLL dasselbe für den rechten Motor

für die Drehrichtungen der Motoren

DIR1_SOLL	neu berechneter Drehsinn des linken Motors. Der Wert von DIR1_SOLL ist +1, wenn sich der Motor im Gegenuhrzeigersinn drehen soll; er ist -1, wenn der Motor sich im Uhrzeigersinn drehen soll.
DIR2_SOLL	dassselbe für den rechten Motor. Bezüglich des Drehsinss gilt dieselbe Festlegung, wie beim linken Motor.
SCHRITTZAHL1	Schrittzahl die der linke Motor zurücklegen soll, um den neu berechneten Punkt auf der Geraden anzufahren
SCHRITTZAHL2	dasselbe für den rechten Motor

Konstanten für die Plotterparameter:
SCHRITTWEG1 Weglänge pro Schritt, linker Antriebsmotor, z.B. in cm/Schritt.
SCHRITTWEG2 dasselbe für den rechten Motor

Prozedur DRAW_LINE:

```
// Abfragen, ob das Ende der Geraden erreicht wurde
        ? S_IST  >  S_MAX                  // prüfen, ob Endpunkt erreicht wurde
        SPRUNG -> DRAW_LINE_END     // Sprung, wenn Endpunkt erreicht

// Endpunkt der Geraden ist noch nicht erreicht
// jetzt die kumulierten Schrittzahlen für den nächsten Zwischenpunkt auf der Geraden berechnen
        S_IST := S_IST + 1                   // Laufvariabel aufzählen
        LL1_SOLL := SQRT(Formel (2_1)) // neue Soll-Länge linke Liane berechnen
        LL2_SOLL := SQRT(Formel(2_2))  // dito rechts
        NSCHRITT1_SOLL := INT(LL1_SOLL / SCHRITTWEG1 // kumulierte Schrittzahl
                                                       // für den neuen Zwischenpunkt berechnen
        NSCHRITT2_SOLL := INT(LL2_SOLL / SCHRITTWEG2         // dito für rechts

// jetzt die neue Schrittzahldifferenz und Drehrichtung für linken Motor berechnen
```

```
// Der Motor muss im Uhrzeigersinn drehen, damit die linke Liane länger wird!!!
        SCHRITTZAHL1 := NSCHRITT1_IST - NSCHRITT1_SOLL
        SCHRITTZAHL1_IST := SCHRITTZAHL1_IST + SCHRITTZAHL1
        DIR1 := SIGN(SCHRITTZAHL1)
        SCHRITTZAHL1 := ABS(SCHRITTZAHL1)

// jetzt die neue Schrittzahldifferenz und Drehrichtung für rechten Motor berechnen
// Der Motor muss im Gegen-Uhrzeigersinn drehen, damit die rechte Liane länger wird!!!
        SCHRITTZAHL2 := NSCHRITT2_SOLL - NSCHRITT2_IST
        SCHRITTZAHL1_IST := SCHRITTZAHL2_IST + SCHRITTZAHL2
        DIR2 := SIGN(SCHRITTZAHL2)
        SCHRITTZAHL2 := ABS(SCHRITTZAHL2)

// jetzt Werte an die Motorsteuerung weitergeben
        ...
        SPRUNG -> DRAW_LINE_EXIT

DRAW_LINE_END:
// Linienende wurde erreicht
// jetzt anschliessende Gerade laden ohne den Stift anzuheben oder
// Stift anheben und nächsten Geradenanfang anfahren
        ...

// Prozedurende
DRAW_LINE_EXIT:
```

Das ist nur als Muster gedacht; auch um auf die Feinheiten mit den Drehrichtungen hinzuweisen. In diesem Muster ist die gespeicherte Länge der linken Liane immer negativ oder Null. Das ist zwar widersinnig, aber praktisch, weil dadurch die Drehrichtung sehr einfach zu bestimmen ist. Die Diskrepanz zwischen berechnetem und tatsächlich angefahrenem Plotpunkt (bedingt durch die konstanten Schrittwege) wird nach diesem Verfahren mit jedem neuen Zwischenpunkt automatisch bis auf Differenzen ausgeglichen, die kleiner als ein Schrittweg sind. Ein Summieren der Abweichungen, wie bei Bresenham, ist hier überflüssig.

Deshalb sollte auch die Verzerrung in Richtung des linken Motors nicht auftreten, die im „Kritzler"-Programm auftritt.

Die trapezförmige Verzeichnung entlang der y-Achse (entlang der x-Achse äquidistante Plotpunkte rücken umso stärker zusammen, je weiter unten sie geplottet werden) kommt vom Durchhängen der Lianen. Wie das zu kompensieren ist, versuche ich, in Abschnitt 20 (Plotgenauigkeit) herauszubekommen.

3. Einen Spline plotten

Splines sind parametrische Kurven mit bis zu zwei Wendepunkten. Sie lassen sich nach bekannten Verfahren (Methode der kleinsten Fehlerquadrate) leicht stückweise an kompliziert verlaufende Originalkurven anpassen. Von der Originalkurve müssen die x- und y-Koordinaten von mindestens vier Punkten bekannt sein.

Ein zweidimensionaler Spline wird durch

$$(3\_1) \qquad \vec{p}(s) = \begin{pmatrix} a_x s^3 & +b_x s^2 & +c_x s & +d_x \\ a_y s^3 & +b_y s^2 & +c_y s & +d_y \end{pmatrix}$$

gegeben. Die acht Parameter werden durch Anpassen des Splines an die Originalkurve bestimmt. Die Parameter können so bestimmt werden, dass Laufvariable s zwischen Anfang und Ende des Splines einen beliebig vorgegebenen Wertebereich, z.B. $[0 \leq s \leq s_{max}]$ durchläuft.

Um den Spline mit dem Tarzanplotter zu zeichnen, wird der Wertebereich der Laufvariable s mit einer konstanten Schrittlänge durchlaufen:

$$s_k = k \cdot \Delta s \quad \text{mit} \quad \Delta s = \left(\frac{s_{max}}{K}\right) \quad \text{und} \quad 0 \leq k \leq K \quad .$$

K legt fest, mit welcher Auflösung, d.h. mit wievielen Schritten, der Spline geplottet wird. Die Parameter rechnet man zweckmässigerweise um:

$$\begin{aligned} A_x &= a_x \cdot \Delta s^3 \;;\; B_x = b_x \cdot \Delta s^2 \;;\; C_x = c_x \cdot \Delta s \\ A_y &= a_y \cdot \Delta s^3 \;;\; B_y = b_y \cdot \Delta s^2 \;;\; C_y = c_y \cdot \Delta s \end{aligned}$$

Der Punkt auf dem Spline bei s_k , z.B. ist dann

$$\vec{p}(s_k) = \begin{pmatrix} A_x k^3 & +B_x k^2 & +C_x k & +d_x \\ A_y k^3 & +B_y k^2 & +C_y k & +d_y \end{pmatrix} ,$$

Geplottet wird der Spline, indem man gemäss Gleichungen (1_3) und (1_4) die entsprechenden Lianenlängen berechnet:

$$(3\_2) \qquad l_1(s_k)^2 = \left(p_x(s_k) + \frac{L}{2}\right)^2 + p_y(s_k)^2$$

$$(3\_3) \qquad l_2(s_k)^2 = \left(p_x(s_k) - \frac{L}{2}\right)^2 + p_y(s_k)^2$$

Der Spline wird dann mit der Schrittweite Δs durch Geradenstücke angepasst.

10. Umrechnen der Bildkoordinaten ins Plotter-Koordinatensystem
11. Verschieben und Skalieren des Bildes in das Plotfeld (ohne Drehung)

Um einen vorgegebenen Punkt

$$\vec{r} = \begin{pmatrix} x \\ y \end{pmatrix}$$

in dem Zeichenfeld abzubilden, muss das Koordinatensystem des Originalbildes in das des Plotters umgerechnet werden (von Drehungen ersteinmal abgesehen). Angenommen, das Original beansprucht die Fläche der Höhe h und der Breite b . Die linke untere Ecke des Originals soll, in dessen eigenen Koordinatensystems gemessen, im Punkt $\vec{r}_0$ liegen. Der Mittelpunkt des Originals liegt bei $\vec{r}_M = \begin{pmatrix} h+x_0 \\ b+y_0 \end{pmatrix}$ Die x-Achse soll in eine horizontale Linie abgebildet werden.

Beispiel:
Der Ursprung des Originals liegt im Mittelpunkt seiner Bildfläche. Dann ist

$$\vec{r}_0 = -\frac{1}{2}\begin{pmatrix} h \\ b \end{pmatrix} \text{ und } \vec{r}_M = \begin{pmatrix} h+x_0 \\ b+y_0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} h \\ b \end{pmatrix}$$

Die Fläche des Originals muss auf die Plotfläche der Höhe H und Breite B abgebildet werden. Die gewünschte Vergrösserung oder Verkleinerung des Originals wird mit dem Plotfaktor F eingestellt. Der Mittelpunkt des Originalbildes soll in den Mittelpunkt $\vec{M}$ des Plotfeldes abgebildet werden.

Die Abbildungsgleichung, die die Punkte $\vec{r}$ des Originals in die Punkte $\vec{p}$ im Plotfeld abbildet, lautet

(10_1) $\vec{p} = F\left(\vec{r}-\vec{r}_M\right) + \vec{p}_M$

20. Plotgenauigkeit
21. Genauigkeit in x-Richtung

Die Genauigkeit, mit der ein bestimmter Punkt $\vec{p}$ angefahren werden kann, verschlechtert sich, je stärker die Lianen unter ihrem eigenen Gewicht durchhängen. Je länger l_1 bzw. l_2 sind und umso mehr Masse pro Längeneinheit die Lianen haben, desto stärker werden sie durchhängen. Dementsprechend ist der Abstand des tatsächlich angefahrenen Punktes vom Ursprung kleiner, als gemäss Gleichungen (1_3) bzw. (1_4) berechnet. Der Fehler der x-Komponente verschwindet für alle Punkte entlang der Mittelachse der Plotfläche, weil sich hier der Durchhang beider Lianen kompensiert.

Jede Liane für sich folgt der „Kettenkurve“ (Ableitung siehe Anhang):

DIESER ABSCHNITT IST NOCH IN ARBEIT

Anhang:

A_1 Ableitung der Kettenkurve (aus Wikipedia.de)

Die Berechnung der Kettenlinie ist ein klassisches Problem der Variationsrechnung. Man denkt sich ein Seil von gewisser Masse und Länge, das an seinen Enden aufgehängt ist. Die Seilkurve ist das Ergebnis der kleinst möglichen potentiellen Energie des Seils. Das versucht man rechnerisch nachzuvollziehen.

Dazu benötigt man den mathematischen Ausdruck für die potentielle Energie. Er ist eine Verfeinerung des bekannten „Gewicht mal Höhe" mgh. Die Verfeinerung besteht darin, dass die Energie für „alle Teile" des Seils getrennt ausgewertet und zum Schluss aufsummiert wird. Das ist notwendig, weil die Teile des Seils sich auf unterschiedlichen Höhen befinden. Die gedankliche Zerlegung des Seils in immer kleinere Teile macht aus der Summe ein Integral. Die Höhe h aus mgh wird durch die gesuchte Funktion y(x) ersetzt, die Masse m durch die Masse dm des Seilstücks über dem Intervall [x,x + dx]; nach Pythagoras ist dies:

$$dm \;=\; \mu\sqrt{dx^2+dy^2} \;=\; \mu\sqrt{1+\left(\frac{dy}{dx}\right)^2}\,dx \;=\; \mu\sqrt{1+y'^2}\,dx$$

wobei µ die Masse je Meter ist. Wenn das Seil an den Stellen x1, x2 aufgehängt ist, ergibt sich demnach die Energie („Gewicht mal Höhe") als

$$E_p \;=\; \mu g\int_{x_1}^{x_2} dx\, y\sqrt{1+y'^2}$$

Eine ähnliche Überlegung führt auf den Ausdruck für die Länge des Seils:

$$l \;=\; \int_{x_1}^{x_2} dx\sqrt{1+y'^2}$$

Die Energie ist zu minimieren, die Länge ist jedoch vorgegeben. Man bringt dies unter einen Hut durch einen Lagrange-Multiplikator $\mu g y_0$, d.h. man minimiert nun den Ausdruck

$$E_p-\mu g y_0 l \;=\; \mu g\int_{x_1}^{x_2} dx\, y\sqrt{1+y'^2}\left(y-y_0\right) \quad ,$$

also

$$\delta E_p \;=\; \delta\left\{\mu g y_0 l \;+\; \mu g\int_{x_1}^{x_2} dx\, y\sqrt{1+y'^2}\left(y-y_0\right)\right\} \;=\; 0$$

Die Variation ergibt

$$\mu g y_0\sqrt{1+y'^2} \;=\; \mu g y\sqrt{1+y'^2}\left(y-y_0\right)$$

die Differentialgleichung (Euler-Lagrange-Gleichung):

$$(y-y_0)y''-y'^2 = 1$$

Interessanterweise sind in diesem Schritt sowohl die Massengröße μ als auch die Schwerebeschleunigung g herausgefallen. Ein schweres Seil nimmt somit dieselbe Form an wie ein leichtes, und auf dem Mond ergibt sich trotz anderer Fallbeschleunigung dieselbe Form wie auf der Erde.

Die Lösungen der Gleichung sind die Funktionen

(A_1_1) $$y(x) = a \cdot \cosh\left(\frac{x-x_0}{a}\right)+y_0$$

Es handelt sich um vergrößerte und verschobene Cosinus-Hyperbolicus-Funktionen. a ist der Krümmungsradius im Scheitelpunkt (siehe Abbildung) und zugleich der Vergrößerungsfaktor. x_0 ist die Verschiebung in x-Richtung, y_0 die Verschiebung in y-Richtung.

Die konkrete Form, die das Seil letztendlich annimmt, errechnet man, indem man x_0 , y_0 und a so anpasst, dass die Kurve durch die Aufhängepunkte geht und die vorgegebene Länge l hat.

Beispiel:

Als Beispiel sei ein zwischen zwei Pfosten (Abstand w) aufgehängtes Seil der Länge l gegeben (siehe Abbildung oben). Die Pfosten sind gleich hoch und befinden sich bei $x = -\frac{w}{2}$ und $x = +\frac{w}{2}$, es gilt also $x_0 = 0$.

Um den Krümmungsradius a zu berechnen, schreiben wir die Seillänge l als Funktion von a

$$l = \int_{-\frac{w}{2}}^{\frac{w}{2}} \sqrt{1+y'^2}\,dx = 2a\sinh\left(\frac{w}{2a}\right)$$

Diese Beziehung legt a in Abhängigkeit von w und l eindeutig fest. Da man keinen geschlossenen Ausdruck für a angeben kann, muss der Wert mit einem numerischen Verfahren zur Lösung nichtlinearer Gleichungen approximativ berechnet werden.

Zuletzt liest man aus der Abbildung die Bedingung $y\left(\frac{w}{2}\right) = 0$ ab, aus der man y_0 erhält. Des Weiteren gelten die Beziehungen

$$\frac{b}{a} = \cosh\left(\frac{w}{2\mathrm{a}}\right) \quad \text{und} \quad b = h+a$$

wobei h der "Durchhang" ist.

Die potentielle Energie dieses Systems beträgt

$$E = -\left(\frac{\mu g}{2}\right)(l b - w a)$$

Genauer ist dies die Energiedifferenz gegenüber dem Fall, dass sich das Seil komplett auf der Höhe der Aufhängepunkte $y = 0$ befindet.

Symmetrisch aufgehängtes Seil mit Umlenkrolle

Mit Hilfe der Energie kann man die Kraft F in den Aufhängepunkten berechnen. Hierzu stellt man sich vor, dass das Seil in einem Aufhängepunkt über eine Umlenkrolle läuft, die die Kraft in horizontale Richtung umlenkt. Um das Seil wie abgebildet um eine sehr kleine Strecke ds hinauszuziehen, muss man die Energie $dE = F\,ds$ aufwenden. Diese kann man berechnen und erhält so die Kraft $F = \frac{dE}{ds}$. Zur Berechnung von dE vergleicht man die Energie des ursprünglichen Seils mit der des um ds verkürzten Seiles. Das Ergebnis ist überraschend einfach, nämlich

$$F = \mu g b = \mu g (h+a) = \left(\frac{m g}{2}\right) \coth\left(\frac{w}{2\mathrm{a}}\right)$$

Dieselbe Formel kann man auch auf Teilstücke des Seils anwenden. Da die Teilstücke alle denselben Krümmungsradius a haben, aber für kleine Teilstücke (unten im Tal) der Durchhang h vernachlässigbar wird, besteht im Tal des Seiles die Seilspannung $\mu g a$.

Stellt man die Pfosten nah beisammen, dann dominiert der Durchhang h , der dann recht genau die halbe Seillänge ist. Die Kraft ist dann erwartungsgemäß die halbe Gewichtskraft des Seiles, $\frac{m g}{2}$ (man beachte, dass zwei Aufhängepunkte sich die Last teilen).

Die Formel zeigt auch, wie die Kraft bei zunehmender Seilspannung die halbe Gewichtskraft um den Faktor $\coth\left(\frac{w}{2a}\right)$ übersteigt. Der Faktor ist praktisch 1 für sehr kleine Krümmungsradien a , aber ungefähr $\frac{2a}{w}$ oder auch $\frac{2a}{l}$ für sehr große Krümmungsradien.

Im Alltag beträgt der Faktor etwa 2 bis 4. Im Aufhängepunkt wirkt dann das ganze oder doppelte Gewicht des Seiles.

A_2 Unsymmetrisch aufgehängtes Seil

Das Seil der Länge l soll zwischen zwei Pfosten (Abstand w) aufgehängt sein. Der linke Pfosten soll die Höhe y_1 , der rechte y_2 haben. Der linke Pfosten steht bei $x = L - w$ der rechte bei $x = L$.

DIESER ABSCHNITT IST NOCH IN ARBEIT!!!