

Technische Universität Dresden

Fakultät Elektrotechnik und Informationstechnik

Institut für Regelungs- und Steuerungstheorie

Masterarbeit

Über den Einfluss hochfrequenter mechanischer Oszillationen auf das Schaltverhalten supraleitender PID-Regler auf Quantenbasis

Eine Fallstudie unter besonderer Berücksichtigung
stochastischer Einflüsse

vorgelegt von: Martin P. Mustermann
geboren am: 1. Januar 1912 in Dresden

zum Erlangen des akademischen Grades

Master of Science

(M. Sc.)

Betreuer:

Betreuer 1

Betreuer 2

Verantwortlicher Hochschullehrer:

Prof. Dr.-Ing. habil. Dipl.-Math. K. Röbenack

Tag der Einreichung:

2. Februar 2222

Bitte ersetzen Sie diese Seite vor dem Binden mit der Aufgabenstellung.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tage dem Prüfungsausschuss der Fakultät Elektrotechnik und Informationstechnik eingereichte Masterarbeit zum Thema

Über den Einfluss hochfrequenter mechanischer Oszillationen auf das Schaltverhalten supraleitender PID-Regler auf Quantenbasis

selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen sind, wurden als solche kenntlich gemacht.

Pirna, 1. Januar 2016

Martin P. Mustermann

Kurzfassung

An dieser Stelle fügen Sie bitte eine deutsche Kurzfassung ein.

Abstract

Please insert the English abstract here.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Levenberg-Maquadrat-Methode | 1 |
| 1.1 | Methode zur Lösung des Quadratmittelproblems | 1 |
| 1.1.1 | Gauß-Newton-Verfahren | 1 |
| 1.1.2 | Levenberg-Marquadt-Algorithmus | 2 |
| 1.2 | Erweiterung von PyTrajectory | 2 |
| 1.2.1 | Optimale Überführungszeit | 3 |
| 1.2.2 | Bereichgrenzung von k | 13 |
| 1.2.3 | Gütefunktion von k | 13 |
| 2 | Brockett-Bedingung für unteraktuiertes mechanisches System | 15 |
| 2.1 | Erläuterung zur Brockett-Bedingung | 15 |
| 2.2 | Ausgang: Doppelintegrator | 17 |
| 2.3 | Unteraktuiertes mechanisches System | 18 |
| 2.4 | Ausgang: Zwei-Gelenke-Manipulator | 18 |

Kapitel 1

Levenberg-Maquadrat-Methode

1.1 Methode zur Lösung des Quadratmittelproblems

Das Quadratmittelproblem einer Funktionsvektor ist wie folgt definiert[?]:

Definition 1.1 (Quadratmittelproblem). *Gegeben ist eine Funktionsvektor $\mathbf{F}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Eine Vektor \mathbf{x}^* ist zu finden, dass die euklidische Norm der Funktionsvektor minimiert wird:*

$$\mathbf{x}^* = \min_{\mathbf{x}} |\mathbf{F}(\mathbf{x})|_2^2 = \min_{\mathbf{x}} \left(\sum_{i=1}^m (f_i(\mathbf{x}))^2 \right) \quad (1.1)$$

mit $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^1$.

1.1.1 Gauß-Newton-Verfahren

Sogenanntes Problem kann man mit verschiedenen Verfahren lösen. Eine typische Methode ist das Gauß-Newton-Verfahren, das durch die lineare Annäherung die nichtlineare Probleme löst. Mit der Ableitung 1. Ordnung der Funktion $\mathbf{F}(\mathbf{x})$ wird das Iterationsverfahren mit linearer Konvergenzordnung konvergiert. Im Folgenden wird mit der Taylorentwicklung von \mathbf{F} und f angefangen[?]:

$$\begin{aligned} f(\mathbf{x} + \mathbf{h}) &= f(\mathbf{x}) + f'(\mathbf{x}) \cdot \mathbf{h} \stackrel{f' \equiv \mathbf{J}}{=} f(\mathbf{x}) + \mathbf{J}(\mathbf{x}) \cdot \mathbf{h} + o(\mathbf{h}^2) \approx \mathbf{g}(\mathbf{h}) \\ \mathbf{F}(\mathbf{x} + \mathbf{h}) &= f(\mathbf{x} + \mathbf{h})^T \cdot f(\mathbf{x} + \mathbf{h}) = (f + \mathbf{J}\mathbf{h})^T (f + \mathbf{J}\mathbf{h}) \\ &= f^T f + 2\mathbf{h}^T \mathbf{J}^T f + \mathbf{h}^T \mathbf{J}^T \mathbf{J} \mathbf{h} \approx \mathbf{G}(\mathbf{h}) \end{aligned} \quad (1.2)$$

Das Symbol \mathbf{J} steht für die Jacobian-Matrix. Nach der Ableitung des vorletzten Terms erhält man die 1.- und 2. Ableitung von \mathbf{G} :

$$\mathbf{G}'(\mathbf{h}) = 2\mathbf{J}^T \mathbf{f} + 2\mathbf{J}^T \mathbf{J} \mathbf{h} \quad (1.3)$$

$$\mathbf{G}''(\mathbf{h}) = 2\mathbf{J}^T \mathbf{J} \quad (1.4)$$

Angenommen, dass \mathbf{G}'' positiv definit ist¹. Sei $\mathbf{G}'(\mathbf{h}_*) = \mathbf{0}$, ist \mathbf{h}_* lokal Minimizer. Zur Berechnung die Schrittweite \mathbf{h} lässt sich Gl. 1.3 verschwinden:

$$(\mathbf{J}^T \mathbf{J}) \cdot \mathbf{h} = -\mathbf{J}^T \mathbf{f} \quad (1.5)$$

Der stationäre Punkt \mathbf{x}^* wird aus der vorherige Punkt \mathbf{x}_- und \mathbf{h}_* berechnet:

$$\mathbf{x}^* = \mathbf{x}_- + \mathbf{h}_* = \mathbf{x}_- - (\mathbf{J}^T \mathbf{J})^{-1} \cdot \mathbf{J}^T \mathbf{f} \quad (1.6)$$

Linear Konvergenzordnung bedeutet, dass $\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq \alpha \|\mathbf{x}_k - \mathbf{x}_*\|$ mit $0 \leq \alpha \leq 1$. Eine weitere Beschränkung liegt darin, dass der Anfangsschätzwert \mathbf{x}_0 in der Nähe von \mathbf{x}_* sein muss. Aber manchmal fällt das GN-Verfahren solche Bedingungen. Daher wird in nächsten Abschnitt eine andere Methode vorgestellt, die obige Bedingungen erfüllt kann.

1.1.2 Levenberg-Marquadt-Algorithmus

Der nach Kenneth Levenberg und Donald Marquardt benannte Algorithmus ist tatsächlich eine Mischung von Methode des steilsten Abstiegs und Gauß-Newton-Verfahren. Gl. 1.7 zeigt die detaillierte Form von Schrittweite $\mathbf{h}[?][?]$:

$$(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}) \cdot \mathbf{h} = -\mathbf{J}^T \mathbf{f} \quad (1.7)$$

\mathbf{I} ist die Einheitsmatrix, μ ist der Dämpfungsparameter. Aufgrund des Dämpfungsparameters ist die Positivdefinitheit der Matrix $\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}$ gesichert. Falls μ groß ist (das passiert am Anfang der Iteration), ist die Darstellung von Schrittweite ähnlich wie Methode des steilsten Abstiegs, damit die Funktion bei \mathbf{x} weit von Zielpunkt schnell konvergiert wird. Andernfalls drückt der LM-Algorithmus bei kleinem μ oder in der Umgebung von \mathbf{x}_* wie NM-Verfahren aus. Das heißt, in den letzten Schritte wird die Funktion auch schnell konvergiert.

1.2 Erweiterung von PyTrajectory

Pytrajectory ist ein Python Paket zum Trajektorie-Entwurf mit gegebenen Randwertbedingungen (Anfangs- und Endwerte der Systemzuständen sowie Eingängen) für nicht

¹Das ist ein Nachteil von Gauß-Newton-Verfahren. Die Positivdefinitheit kann nicht immer gesichert werden

lineares System. Das Paket ist auf Python 2-Version entwickelt. Für weitere Informationen kann man [Pytrajectory](#) klicken.

1.2.1 Optimale Überführungszeit

Eine Aufgabe bei der Erweiterung von PyTrajectory liegt darin, die Überführungszeit der Trajektorienplanung eines Systems zu berücksichtigen. Es ist gehofft, die Überführungszeit T nicht vom Nutzer explizit vorgegeben wird, sondern anhand der Systemgleichungen mit Levenberg-Marquadt-Methode automatisch optimiert wird. Anfangspunkt ist die Ansetzung der Zeittransformation von $t = t$ zum $t = k\tau$, womit k eine zusätzliche freie Parameter (ohne Einheit) ist. Dann folgt die untere Gleichungstransformation:

$$\dot{x} = \frac{dx}{dt} = F(x, t) \quad (1.8)$$

$$\dot{x}_{new} = \frac{dx}{d\tau} = \frac{dx}{dt} \cdot \frac{dt}{d\tau} = k \cdot \frac{dx}{dt} = k \cdot F(x, t) = F_{new}(x, t) \quad (1.9)$$

Bei $k > 1$ kostet das System mehrere Zeit die Endposition zu erreichen. Zur Überprüfung der Rationalität werden zwei Beispiele gebietet. Für jedes Beispiel vergleicht die unter der Berücksichtigung von Zeit entworfene Trajektorie (abgekürzt als “Trajektorie mit k ”) mit dem zum originalen System geeigneten Trajektorie (Kurz als “Trajektorie ohne k ”).

Beispiel 1.1 (Doppelintegrator). *Das einfache Beispiel zielt auf die Darstellung der Vor- und Nachteile oberer Idee. Betracht man einen auf der X-Achse laufenden Wagen mit zwei Systemzuständen der Verschiebung x_1 und der Geschwindigkeit x_2 . Eine aufgeprägte Kraft F wirkt auf den Wagen ein. Die Systemgleichung lässt sich wie Gl. 1.10 darstellen.*

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \quad (1.10)$$

Es ist eingeplant, dass der Wagen von 0 bis zu 1m bewegen kann und am Anfang und Ende soll er ruhig sein. Mit anderen Worten ist der Anfangs- und Endwert von x_1 und x_2 jeweils (0,1) sowie (0,0). Die Überführungszeit ist 1s oder k Sekunde für die zwei Systeme.

Die Anfangswerte der Parameter wie folgt eingestellt: die Anfangsschätzwerte der Parameter für die Polynome sind eine Liste all 0.1, der Anzahl der Spline-Abschnitt für beide x und u ist 2, das Vielfache der Iteration ist auch 2. Jedes System benötigt 2 mal Iteration (nämlich 4 Spline-Abschnitte), um eine exakte Lösung zu finden.

Abb. 1 und Abb. 2 zeigt die Simulationsergebnisse des Systems mit fester Überführungszeit (1s). Unter Wirkung eines fast sinusförmigen Eingang mit maximalen ungefähr 6N

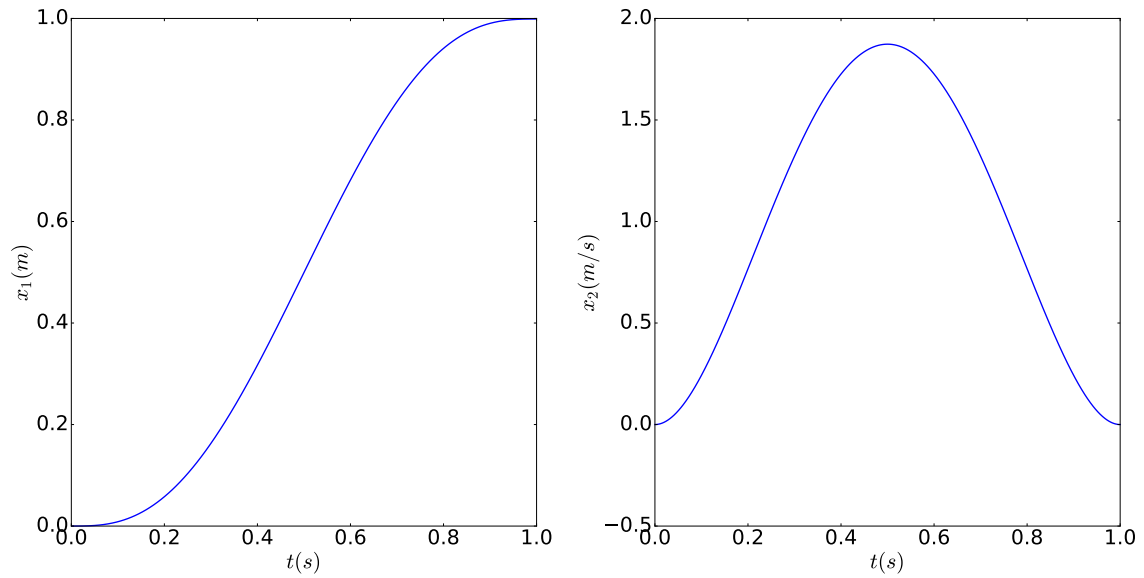


Abbildung 1 – Zeitverlauf der Systemzustände vom System ohne der Wirkung von k . x_1 und x_2 steht für Position und Geschwindigkeit.

läuft der Wagen entlang der geplanten Weg. Das heißt, der Wagen beschleunigt sich in die ersten Halbzeit, nach der Erreichung der maximale Geschwindigkeit (circa 1.8m/s) bremsst er bis zum Stoppen.

Andererseits ist die Trajektorie für das System mit “ k ”. Aus Abb. 3 und Abb. 4 beobachtet man darauf, zwar die maximale Kraft und die Geschwindigkeit weit kleiner als die im oberen System, ist der Wert von k hier ziemlich groß (ungefähr 1662,53s)! Der Wagen in diesem System muss 1662s lang bewegen, bevor er die Endposition angekommen. Daher ist die Lösung von k **richtig** aber **nicht rational**.

Zur Verbesserung des Ergebnisses wird eine andere Anfangswert der Polynomparameter gewählt. Statt aller Werte 0.1 sind die Parameter jetzt gleich die Rechenergebnisse von Polynomparameter nach der ersten Iteration in dem System ohne “ k ”. Die Kurven in Abb. 5 und 6 sehen ähnlich wie Abb. 1 und Abb. 2 mit einem kleinen $k = 1.21$. Endlich stoppt der Wagen in 1.21s am 1m.

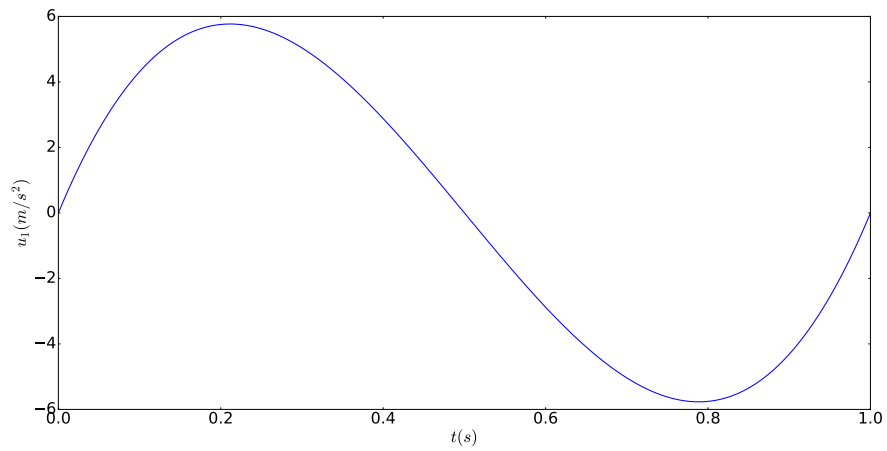


Abbildung 2 – Verlauf des Systemeingangs vom System ohne der Wirkung von k . u_1 steht für Wagenbeschleunigung.

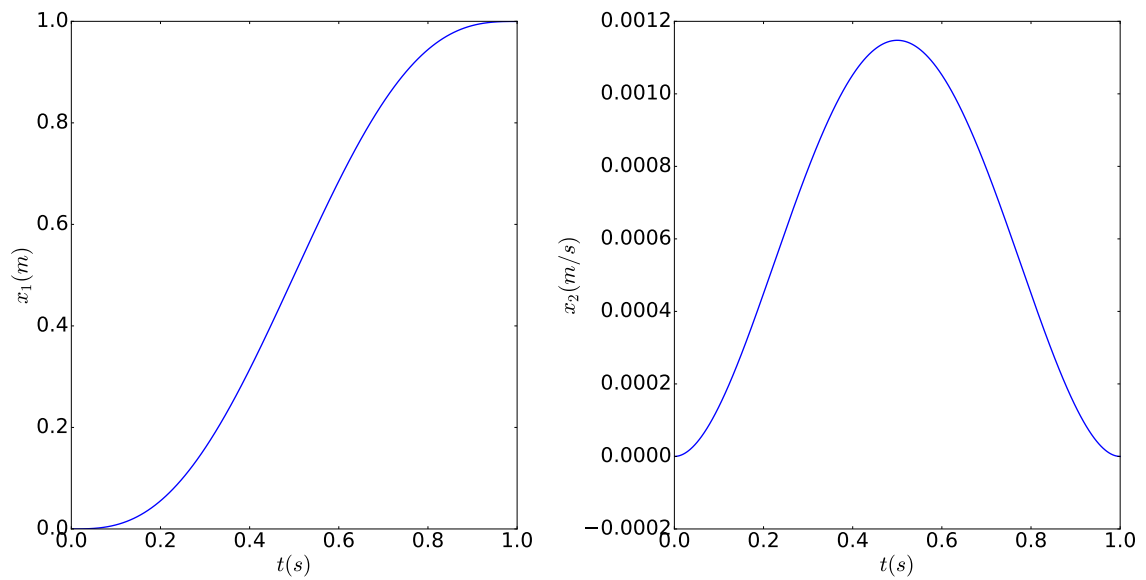


Abbildung 3 – Verlauf des Systemeingangs vom System mit der Wirkung von k . u_1 steht für Wagenbeschleunigung.

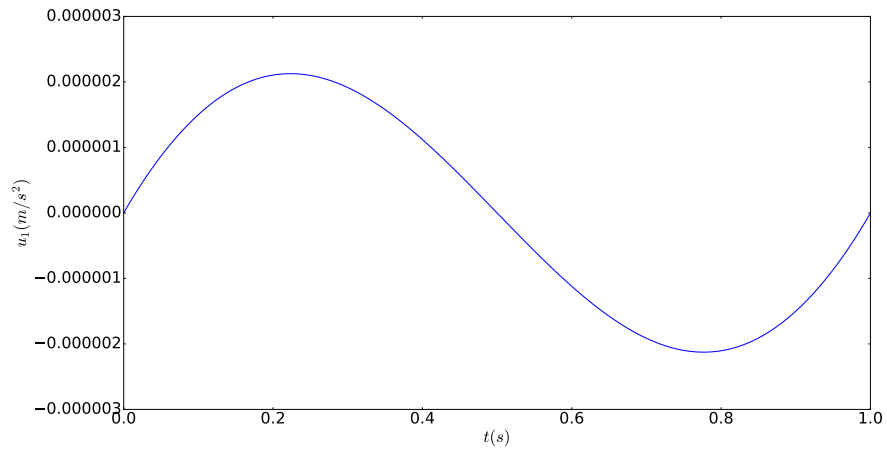


Abbildung 4 – Verlauf der Systemzustände vom System mit der Wirkung von k . x_1 und x_2 steht für Position und Geschwindigkeit.

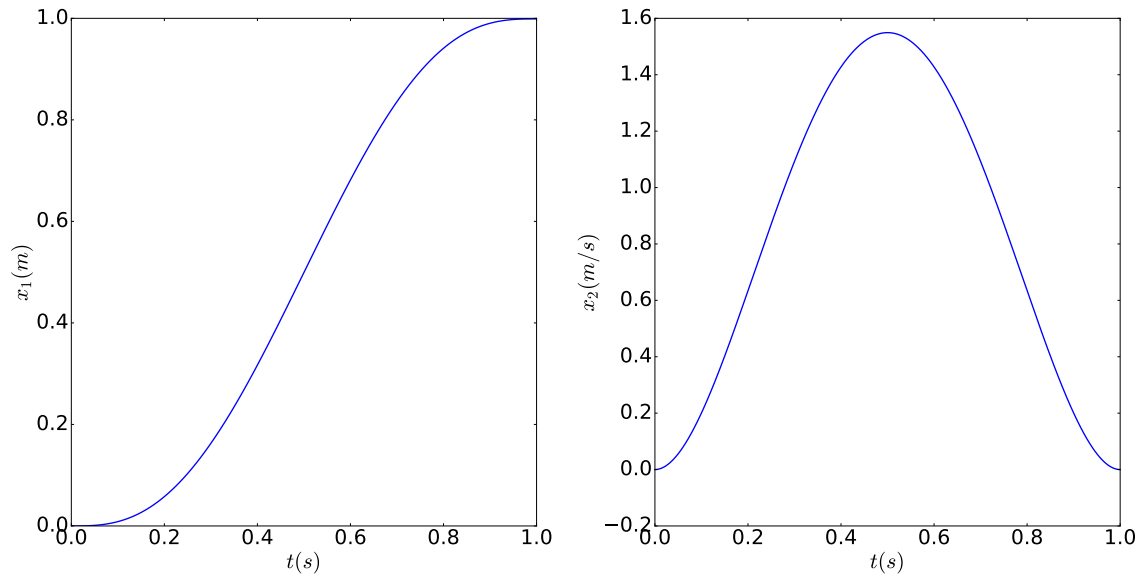


Abbildung 5 – Verlauf des Systemeingangs vom System mit der Wirkung von k . u_1 steht für Wagenbeschleunigung.

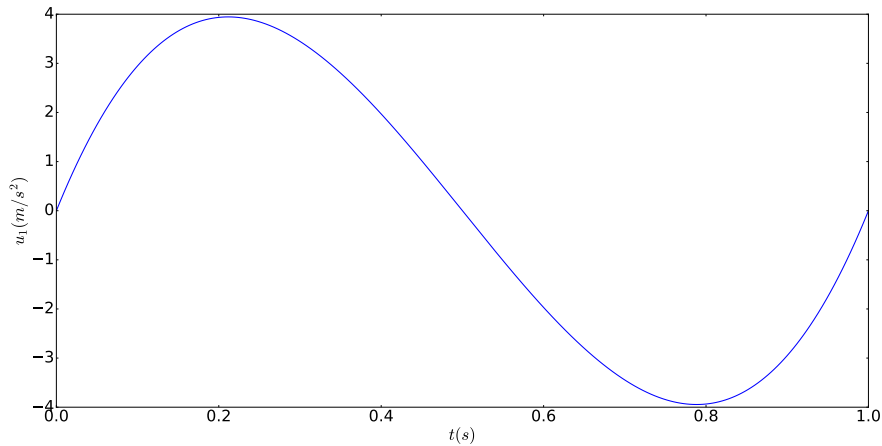


Abbildung 6 – Verlauf des Systemeingangs vom System mit der Wirkung von k . u_1 steht für Wagenbeschleunigung.

Dieses Beispiel zeigt die starke Abhängigkeit der Konvergenzmöglichkeit von Levenberg-Marquadt-Algorithmus und des Anfangsschätzwerts. Wenn die Anfangsschätzwerte der Parameter zu weit vom Minimierer eingestellt sind, kann die Methode das System nicht konvergieren oder zu einem lokalen, nicht globalen optimierten Punkt richten.

Das folgende Beispiel stellt die Anwendung der LM-Methode in einem verbreiteten Benchmark System vor.

Beispiel 1.2 (System-Inverses-Pendel). *Abb. 7 zeigt das Schema des inversen Pendels. Der Pendelarm ist auf einem horizontal bewegten Wagen montiert. Der Wagen bringt daher eine horizontale Kraft auf das Pendel auf. Beim Aufprägen der Kraft F auf den Wagen bewegt sich das Pendel von oben nach unten. Die instabile Ruhelage ist der Punkt, in dem das Pendel genau senkrecht zum Wagen steht und der Drehwinkel 180° beträgt. Systemvariable in diesem Modell sind die Wagenposition x_1 , dessen Geschwindigkeit x_2 , der Pendeldrehwinkel x_3 , die Drehgeschwindigkeit davon x_4 . Der einzige Systemeingang ist die auf den Wagen aufgeprägte Kraft f_w .*

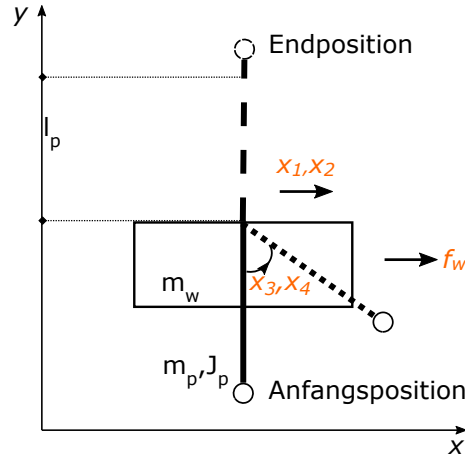


Abbildung 7 – Struktur des inversen Pendels: in dem System gibt es vier Zustände x_1 bis x_4 . Das Pendel folgt einer gegebenen Trajektorie und bewegt sich von 180° bis zur instabilen Ruhelage (nämlich vom ganz unten zu ganz oben). m_w und m_p sind jeweils die Masse des Wagens und des Pendels, J_p ist das Trägheitsmoment und l_p ist der Abstand zwischen dem Stützpunkt und dem Pendelschwerpunkt. Die Pfeile neben Zuständen geben die Vektorrichtung an.

Dazugehörige Differentialgleichungen des Modells sind wie folgt beschrieben[?]/[?]:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= \frac{(J_p + m_p l_p^2) (-f_w x_2 + m_p l_p \sin(x_3) x_4^2 + x_5) + m_p l_p f_p \cos(x_3) x_4}{(m_w + m_p) J_p + m_w m_p l_p^2 + m_p^2 l_p^2 \sin^2(x_3)} \\
 &\quad + \frac{m_p l_p f_p \cos(x_3) z_4 + m_p^2 l_p^2 g \cos(x_3) \sin(x_4)}{(m_w + m_p) J_p + m_w m_p l_p^2 + m_p^2 l_p^2 \sin^2(x_3)} \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{-m_p l_p \cos(x_3) x_5 + f_w m_p l_p \cos(x_3) x_2 - m_p^2 l_p^2 \cos(x_3) \sin(x_3) x_4^2}{(m_w + m_p) J_p + m_w m_p l_p^2 + m_p^2 l_p^2 \sin^2(x_3)} \\
 &\quad - \frac{(m_w + m_p) f_w x_4 + (m_w + m_p) g m_p l_p \sin(x_3)}{(m_w + m_p) J_p + m_w m_p l_p^2 + m_p^2 l_p^2 \sin^2(x_3)} \quad (1.11)
 \end{aligned}$$

Vereinfachen die Gleichungen und Ignorieren kleiner Teiler erhält man eine kompakte Zustandsdarstellung des inversen Pendels[?]:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= \frac{m_p \sin(x_3) (-l_p x_4^2 + g \cos(x_3))}{m_w l_p + m_p \sin^2(x_3)} + \frac{\cos(x_3)}{m_w l_p + m_p \sin^2(x_3)} u \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{\sin(x_3) (-m_p l_p x_4^2 \cos(x_3) + g(m_w + m_p))}{m_w l_p + m_p \sin^2(x_3)} + \frac{\cos(x_3)}{m_w l_p + m_p \sin^2(x_3)} u \quad (1.12)
 \end{aligned}$$

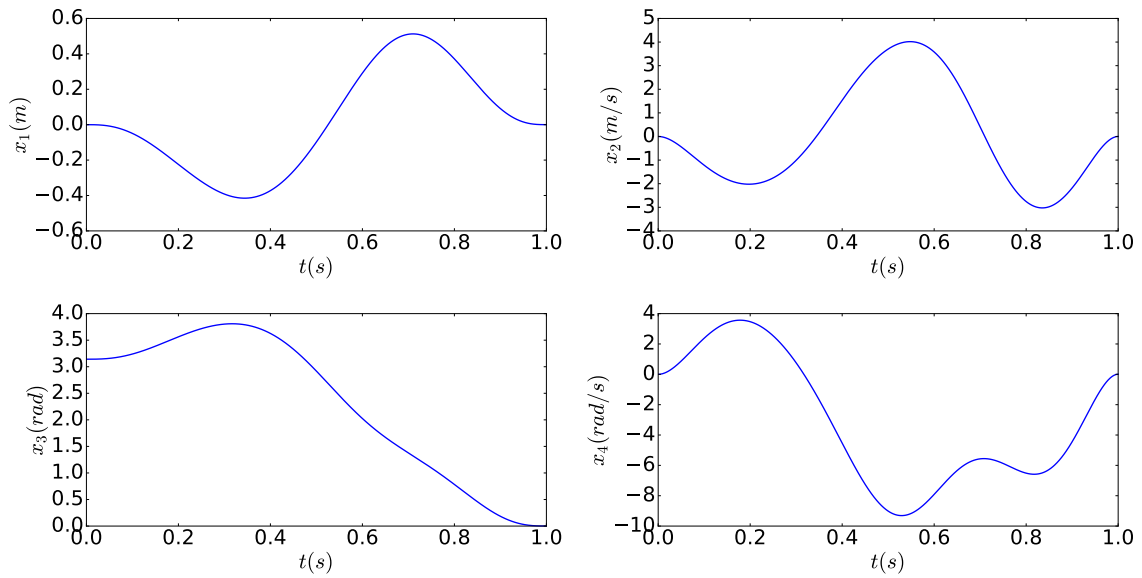


Abbildung 8 – Systemzuständen des inversen-Pendels (ohne k) mit dem virtuellen Eingang.

Wenn das System partiell linearisiert und einen virtuellen Eingang, die Beschleunigung des Wagens \dot{x}_2 anstatt der Kraft u gegeben wird, lässt sich die Systemdarstellung vielmehr einfacher darstellen:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= u \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{g}{l_p} \sin(x_3) + \frac{1}{l_p} \cos(x_3) u
 \end{aligned} \tag{1.13}$$

Weil der Pendel sich von unten nach oben drehen und der Wagen endlich zum Startpunkt zurückgehen soll, ist die Trajektorie wie $(0, 0, \pi, 0) \rightarrow (0, 0, 0, 0)$ erfordert. Der Anfangszahl der Spline-Abschnitt und das Iterationsvielfache ist jeweils 2. All die Anfangsschätzwerte der Polynomkoeffizienten sind 0.1. Die Ergebnisse sind in Abb. 8, Abb. 9, Abb. 10 und Abb. 11 dargestellt.

Nach 5 Mal Iteration (also 32 Spline-Abschnitt) bekommen beide Systeme Lösung. Der Wert von k ist 1.48s. Egal das System ohne k oder mit k sind die Kurvenform vom Zustand und Eingang für dieses Beispiel fast gleich. Aber Offensichtlich liefert die Trajektorie des Systems mit k eine kleinere Bewegungsbereich für \mathbf{x} und u (z.B. der Bereich von x_1 ist jeweils $(0.4 - +0.5)m$ und $(-0.57 - +0.31)m$). In Betracht auf winzigen Unterschied zwischen der Überführungszeit ist die Planung für das System mit k ausgezeichnet.

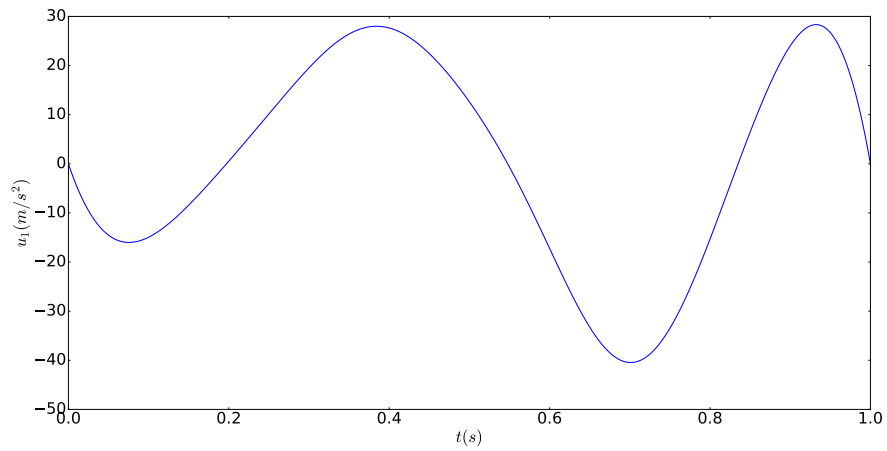


Abbildung 9 – Systemeingang des inversen-Pendels (ohne k) mit dem virtuellen Eingang.

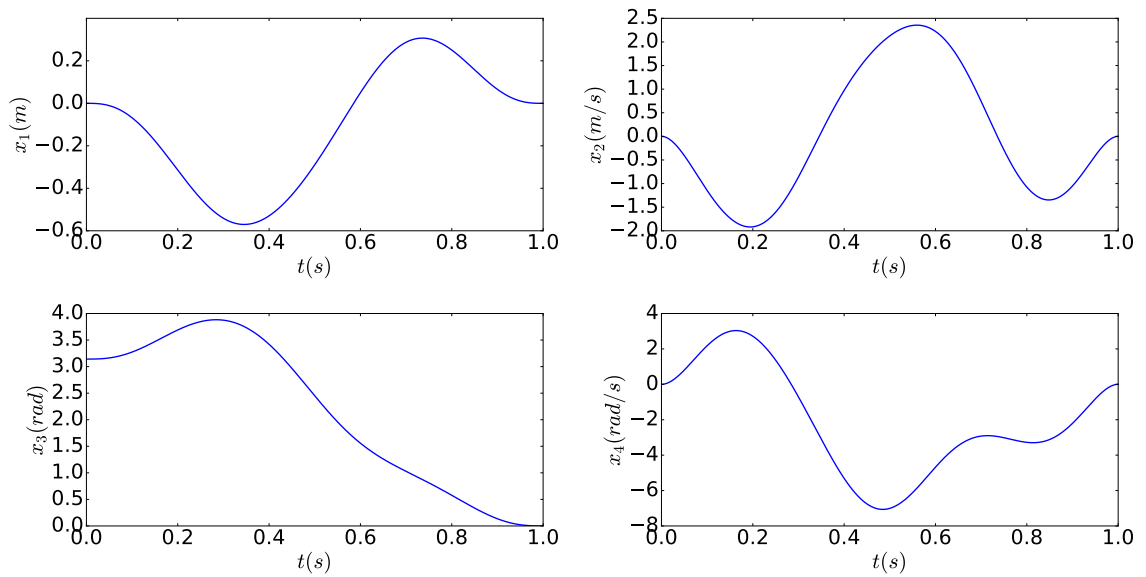


Abbildung 10 – Systemzustandskurven des inversen-Pendels (mit k) mit dem virtuellen Eingang.

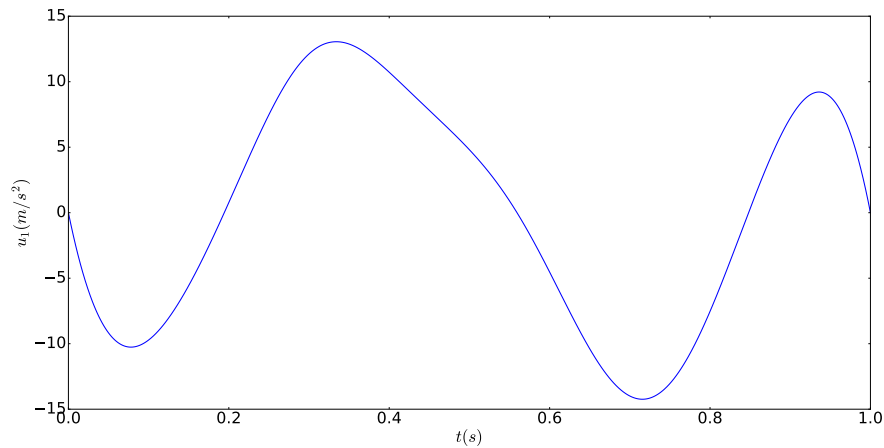


Abbildung 11 – Systemeingang des inversen-Pendels (mit k) mit dem virtuellen Eingang. Die maximale Beschleunigung trägt circa $12m/s^2$ und nur Hälfte des Wertes in Abb. 9.

Jetzt wird ein anderer Vorteil von System mit k diskutiert. Wie würde die Lösung des LM-Algorithmus verändern, wenn der Systemzustand eine Beschränkung ausgeübt würde? Der Testzustand hier ist die Wagenverschiebung x_1 . Der Wagen kann nur im Bereich von $(-0.2 - +0.4)m$ bewegen (Die Zahl ist eine Beschränkung für beide System). Nach 8 Mal Iteration rechnet das System mit k eine Lösung mit $k = 1.0535$ aus, dennoch das andere System kein Ergebnis erhalten kann (mindestens bis 8-te Iteration).

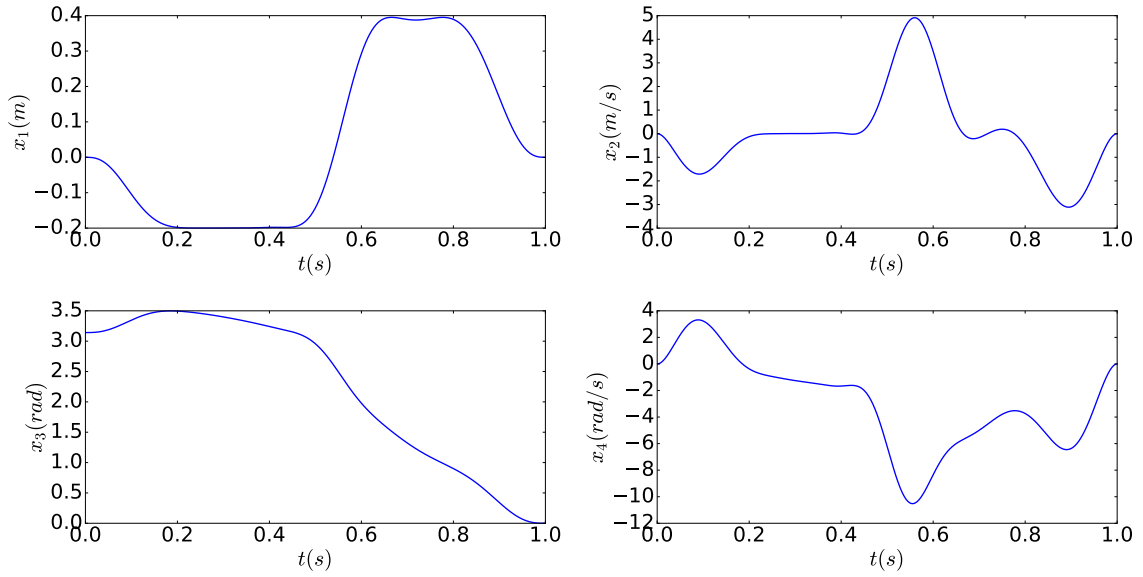


Abbildung 12 – Systemzustände des inversen-Pendels (mit k) mit dem virtuellen Eingang. x_1 ist zwischen $(-0.2, 0.4)$ eingeschränkt.

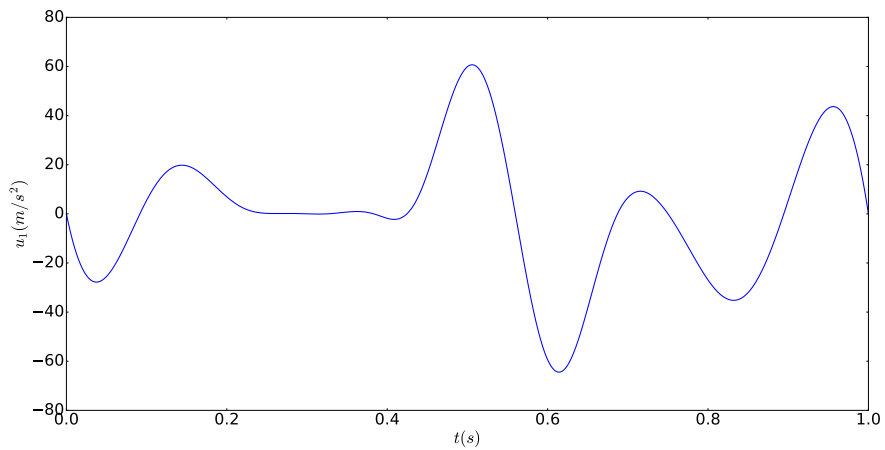


Abbildung 13 – Systemeingang des inversen-Pendels (mit k) mit dem virtuellen Eingang.

Zusammengefasst aus obere Beispiele kann das System unter Berücksichtigung der Überführungszeit eine sparsamere Trajektorie eingeplant werden, sofern gute Anfangswerte des Polynomparameter vorgegeben werden. Aber praktisch ist es nicht so einfach, einige geeignete Anfangswerte zu finden. Deswegen werden einige Methode zum Beschränken der Wertbereich von k verwendet. dass im nächsten Subschnitt beschrieben wird.

1.2.2 Bereichgrenzung von k

Die prinzipielle Idee kommt aus der Beschränkung von Systemzustände, die im Paket “Pytrajectory” schon realisiert wird. Der Anfangspunkt liegt in der Ersetzung der originalen Systemfunktion mit einer monotonen steigenden Sättigungsfunktion, die keine Begrenzung besitzt. Das folgt stellt die Sättigungsfunktion von “ sk ” dar.

$$k = \psi(sk, sk^\pm) = sk^+ - \frac{sk^+ - sk^-}{1 + e^{m \cdot sk}}$$

Weiterhin gilt $m = \frac{4}{sk^+ - sk^-}$. k wird in sk^+ und sk^- beschränkt und mit der Form von ψ in Systemfunktion umgeschrieben. Nachdem der Wert von sk ausgerechnet wurde, wird die Lösung von k auch durch Umkehrfunktion von sk bekommen. Siehe [?] und [?] für weitere Information.

Das Ergebnis vom “begrenzten k ” zeigt daran, den Wert von k immer größer zu sein(egal ob eine approximierte Trajektorie finden kann). Aderersatz vergrößert sich k zu 29.9999 falls $sk^+ = 30$ und $sk^- = 0.1$. Grund dafür ist die Vergrößerung der Iterationsschritt h in jeder Iteration. Wegen der unbeschränkte vergrößerte sk versucht k zur oberen Grenze zu erreichen.

1.2.3 Gütefunktion von k

Eine weitere Methode zur Begrenzung von k liegt darin, eine Gütefunktion am Ende der Zustandsfunktion hinzufügen. Die Gütefunktion ist eine Kombination von Parabel-und Nullfunktion. In dem von den zwei Eingänge(x_{min} und x_{max}) definierten Definitionsmenge ist die Zielmenge in der Nähe von 0. Dagegen gilt der Bildwert außerhalb diesem Bereich ähnlich wie $(x - x_{mid})^2$ mit x_{mid} dem Mittelwert von x_{max} und x_{min} . Die konkrete Form von *pe-Funktion* ist:

$$pe = \frac{(x - x_{mid})^2}{1 + e^{5 \cdot (x - x_{min})}} + \frac{(x - x_{mid})^2}{1 + e^{5 \cdot (x_{max} - x)}} \quad (1.14)$$

Ein Beispiel der Kurven von pe mit $x_{min} = 0$ und $x_{max} = 10$ ist wie Abb.15 gezeigt. Zwischen $(0, 10)$ ist pe ungefähr 0. Der andere Teil läuft wie eine Parabel. Mit dieser Gütefunktion von k probiert das System eine Lösung in der Nähe von x_{mid} zu finden.

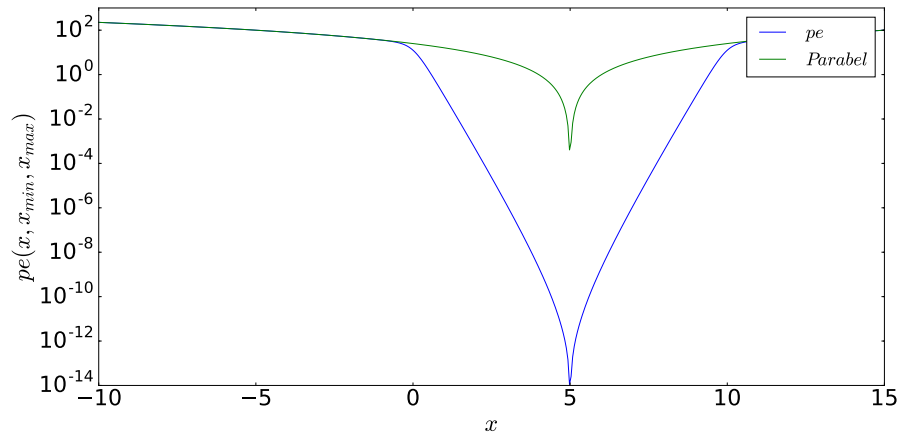


Abbildung 14 – Gütefunktion von k : $x_{min} = 0$, $x_{max} = 10$

Beispiel 1.3. Geht man zurück auf das Beispiel Doppelintegrator. Angenommen, dass x_{min} und x_{max} jeweils 0.1, 3 und der Anfangsschätzwert von k 1.5 ist. Die andere Bedingungen bleiben wie zuvor. Die Zeitverlauf vom Eingang u ist noch sinusförmig aber mit dem maximalen Wert 4m/s^2 . Der maximale Wert von der Wagengeschwindigkeit x_2 ist auch ähnlich wie das System ohne k : circa 1.6m/s . In Anbetracht auf das Ergebnis und die Lösung von k (1.87) zeigt dieses Beispiel eine annähernde Trajektorie wie das originale System.

Kapitel 2

Brockett-Bedingung für unteraktuiertes mechanisches System

2.1 Erläuterung zur Brockett-Bedingung

In Anbetracht von der mathematischen Beschreibung und dem Beweis für Brockett-Bedingung ist zuerst die Erklärung einiger mathematischen Terme notwendig.

Stetigkeit (engl.: continuous) [?, S.250]: Es sei $a \in M$. Die Funktion $f : M \subseteq \mathbb{R} \rightarrow \mathbb{R}$ ist genau dann im Punkt a stetig, wenn es zu jeder reellen Zahl $\varepsilon > 0$ eine reelle Zahl $\delta > 0$ gibt, sodass $|f(x) - f(a)| < \varepsilon$ für alle $x \in M$ mit $|x - a| < \delta$ gilt.

D.h. eine stetige Funktion erfüllt die Bedingung: wenn der Abstand zweier Elemente der Definitionsmenge infinitesimal ist, muss der Abstand ihrer entsprechenden Wertemenge auch infinitesimal sein.

Stetige Differenzierbarkeit (engl.: continuously differentiable) [?, S.256]: Eine differenzierbare Abbildung $f : E \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ sei stetig differenzierbar in E , wenn f' eine stetige Abbildung von E in $L(\mathbb{R}^n, \mathbb{R}^m)$ ist, wobei E eine offene Menge und $L(X, Y)$ Raum linearer Abbildungen ist.

Klasse C^k (engl.: class C^k) [?, S.265]: Eine Funktion auf einer offenen Umgebung des Punktes p stetige Ableitungen bis zur Ordnung k besitzt.

Basierend auf die obige Definition hat eine Funktion von Klasse C^1 die Ableitung 1. Ordnung, die auch stetig ist.

Glatte Funktion (engl.: smooth function) [?, S.5]: Ein Synonym für C^∞ ist "Glatt". Eine glatte Funktion ist nämlich eine Funktion mit stetigen Ableitungen zur unendlichen Ordnung.

Surjektiv (engl.: onto) [?, S.931]: Gegeben sei die Abbildung $f : X \rightarrow Y$. Betrachtet

die Gleichung $f(x) = y$. Wenn die Gleichung für jedes $y \in Y$ eine Lösung $x \in X$ besitzt, d.h. $f(X) = Y$, dann heißt f genau dann *surjektiv*.

Falls jedes Element y der Wertemenge Y kann erreicht werden, dann ist diese Abbildung surjektiv.

Homotopie (engl.: homotopy) [,] (noch nicht geschrieben wird.)

Häufungspunkt (engl.: limit point) [?, S.35]: Ein Punkt p ist ein *Häufungspunkt* der Menge E , wenn in jeder Umgebung von p ein Punkt $q \in E$ mit $q \neq p$ liegt.

Abgeschlossene Menge (engl.: closed set) [?, S.36]: E heißt abgeschlossen, wenn jeder Häufungspunkt von E in E liegt.

Beschränkte Menge (engl.: bounded set) [?, S.36]: E ist beschränkt, wenn eine reelle Zahl M und ein Punkt $q \in X$ existieren, sodass der Abstand von (p, q) kleiner als M für alle $p \in E$ gilt. X ist hier ein metrischer Raum, dessen Teilmenge E ist.

Kompakte Menge (engl.: compact set) [?, S.45]: (Satz, nicht Definition) Falls eine Menge E in \mathbb{R}^k abgeschlossen und beschränkt, dann ist sie kompakt.

Ein sehr einfaches Beispiel der kompakten Menge E ist z.B. $[1, 2]$ mit 1 und 2 jeweils dem linken und rechten Häufungspunkt.

Niveaumenge (engl.: level set) [?, S.94]: Eine Niveaumenge einer Abbildung $f : N \rightarrow M$ ist die Submenge $f^{-1}(c) = \{p \in N \mid f(p) = c\}$ für einige $c \in M$.

Also die Niveaumenge $f^{-1}(c)$ besteht aus die Elemente der Definitionsmenge, deren Bildmenge eine Konstante c ist.

Distribution

Lefschetz-fixed-point-formula

Lokale Lipschitzstetigkeit (engl.: locally Lipschitz continuity) [?, S.553]: Lipschitz-Bedingung bezüglich y ist die Forderung $|f(x, t) - f(y, t)| \leq L|x - y|$ für alle (x, t) und (y, t) . Inzwischen ist L eine beliebige Konstante.

Das heißt, wenn die Ableitung der Funktion von f beschränkt ist, erfordert sie Lipschitz-Bedingung. Die Lipschitzstetigkeit ist stärker als Stetigkeit.

Ein nichtlineares Zustandsraummodell lässt sich durch Gl. 2.1 darstellen:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)), \quad t \geq 0, \quad \mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n, \quad \mathbf{f}(\mathbf{x}_0, 0) = \mathbf{0} \quad (2.1)$$

mit \mathbf{x} dem Systemzustand, \mathbf{f} der nichtlinearen Zustandsfunktion, u der Eingangsgröße und \mathbf{x}_0 dem initialen Zustand.

Jetzt stellt sich die Frage: gibt es die Möglichkeit, dass das obere nichtlineare System um die Ruhelage $\mathbf{x} = \mathbf{x}_0$ mit einer nichtlinearen Zustandsrückführung (nämlich hier u) asymptotisch stabilisierbar sein kann? Zum Antworten der Frage etabliert der amerikanische Mathematiker Roger W. Brockett das folgende berühmte Kriterium[?]:

Theorem 2.1 (Brockett-Bedingung[?]). *Betrachtet man das System (2.1) mit \mathbf{f} stetig differenzierbar in der Umgebung von $(\mathbf{x}_0, 0)$ ist. Angenommen, dass $(\mathbf{x}_0, 0)$ in $\mathbb{R}^n \times \mathbb{R}^m$ asymptotisch stabil unter einer stetigen differenzierbaren Rückführung u ist, dann ist das Bild der Abbildung*

$$(\mathbf{x}, u) \mapsto \mathbf{f}(\mathbf{x}, u)$$

surjektiv zur offenen Menge, die 0 enthält.

Die obere notwendige Bedingung basiert auf eine *zusätzliche* notwendige Bedingung: das linearisierte System enthält keines nicht steuerbares Moduls, dessen Eigenwerte in der rechte Halbebene liegen. Wenn $(\mathbf{x}_0, 0)$ asymptotisch stabil ist, hat das linearisierte System nur steuerbare Module (mit oder ohne Eigenwerte in der rechte Halbebene) oder nicht steuerbare Module mit stabilen Eigenwerte. Mit anderen Worten muss instabile Module steuerbar oder nicht steuerbare Module stabil sein.

Ein leicht verwirrender Punkt dieses Kriterium liegt in die Bedingungen von \mathbf{f} und u in Gl. 2.1. Nach der Beschreibung des Theorems sind beide \mathbf{f} und u *stetig differenzierbar*, und in dem Beweis zitiert Brockett eine stetig differenzierbare Lyapunov Funktion, die aber im Original *glatt* ist. (siehe [?, S.186] und [?, S.324].) In anderen Literaturen sind auch unterschiedliche Annahmen ermöglicht: [?] und [?] setzen \mathbf{f} und u *stetig und zeitinvariant* als bekannt voraus, während in [?] und [?] bringen die Autoren strengeren Bedingung vor: *lokal lipschitz*. Im Buch vom argentinischen Mathematiker Eduardo D. Sontag [?] werden die Bedingung von \mathbf{f} und u gleich wie Brockett (C^1). Eine noch strengere Voraussetzung werde von G. Oriolo und Y. Nakamura in [?] aufgestellt, dass \mathbf{f} *stetig differenzierbar* und u *glatt* sein muss.

Zurück auf Brocketts Beweis????????????????????????????

Skizze des Beweises ([?],[?]). Falls die Ruhelage $(\mathbf{x}_0, 0)$ asymptotisch stabil ist, existiert es nach [?] eine glatte Lyapunov Funktion V , die eine sphärischer homotopy Niveaumenge $V^{-1}(c)$ (c eine kleine Konstante) hat. Wegen der Kompaktheit von $V^{-1}(c)$ ist die Richtung von $\mathbf{f}(\mathbf{x}, u)$ in der Menge $R := \{\mathbf{x} : V(\mathbf{x}) \leq c\}$. Es existiert auch $\xi \in \mathbb{R}^n$ mit $\|\xi\|$ genügend klein, dass $\mathbf{f}(\mathbf{x}, u) - \xi$ auch in R zeigt. Durch die Anwendung des Fixpunktsatz gilt es $\mathbf{f}(\mathbf{x}, u) - \xi = 0$ (oder $\mathbf{f}(\mathbf{x}, u) = \xi$) für einige \mathbf{x} in R . Weil $\|\xi\|$ beliebig und sehr klein ist, bedeutet die obige Funktion, dass \mathbf{f} lösbar in beliebiger Umgebung von 0 ist. \square

2.2 Ausgang: Doppelintegrator

Doppelintegrator ist ein erfüllendes Brockett-Bedingung Beispiel. Die asymptotische Stabilität kann man mithilfe des Paket *Pytrajectory* überprüfen. Das linearisierte affine

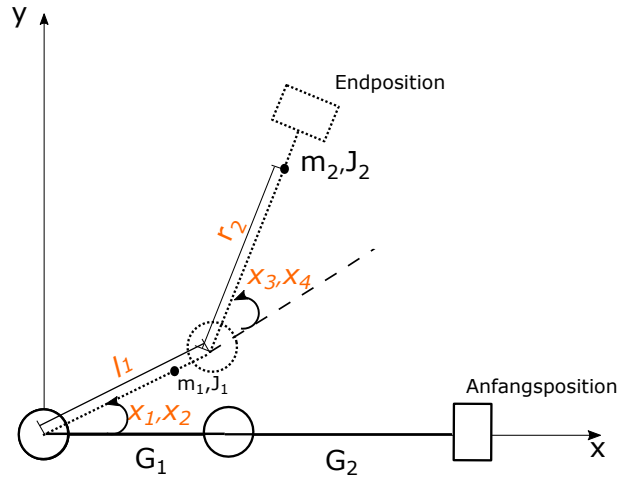


Abbildung 15 – Gütefunktion von k : $x_{min} = 0$, $x_{max} = 10$

System davon lautet:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (2.2)$$

mit der Steuerbarkeitsmatrix $Q_s = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ und $\text{rang}(Q_s) = 2$. Das System ist vollständig steuerbar. Die zusätzliche Bedingung erfüllt.

Jetzt wird die Brockett-Bedingung überprüft. Setzt man voraus, dass ϵ in der offenen Menge von 0 liegt, dann ist

$$\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} = \begin{bmatrix} x_{2\epsilon} \\ u_\epsilon \end{bmatrix} \quad (2.3)$$

Gl. 2.3 kann jeden Wert in der Umgebung von $\mathbf{0}$ ohne Ausnahme erreichen. Dann ist die Brockett-Bedingung automatisch erfüllt. $\mathbf{x}_0 = (x_{beliebig}, 0)$ ist eine asymptotisch stabile Ruhelage.

2.3 Unteraktuiertes mechanisches System

2.4 Ausgang: Zwei-Gelenke-Manipulator