

## Numerisches Praktikum

### Thema 1: Autorennen

bearbeitet von Carsten Erdmann

Auf welcher Linie kann ein Rennauto möglichst schnell durch eine Kurve fahren? Betrachten Sie folgendes Beispiel:

Das Rennauto hat eine Masse von  $m = 600$  kg und kann eine zur Fahrrichtung normal wirkende Kraft von  $F_{\max} = 18\,000$  N aufnehmen, ohne aus der Spur zu rutschen. Die maximal mögliche Beschleunigung  $a_{\max}$  ist monoton fallend von der Fahrgeschwindigkeit  $v$  abhängig, z. Bsp. über

$$a_{\max}(v) = b \left( 1 - \frac{v}{v_{\max}} \right)$$

mit gegebenem  $b$  und der Höchstgeschwindigkeit  $v_{\max} = 100 \text{ ms}^{-1}$ . Der Betrag der stärkstmöglichen Bremsbeschleunigung sei geschwindigkeitsunabhängig  $2b$ . Das Auto ist 2 m breit.

Die 10 m breite Straße beschreibt eine kreisbogenförmige  $90^\circ$ -Kurve mit dem Innenradius 20 m. Vor und nach der Kurve verläuft die Straße geradlinig. Wir nehmen an, daß das Rennauto 25 m vor Eintritt in die Kurve und spätestens 25 m nach Austritt aus der Kurve parallel zur Fahrbahn fährt. Darüberhinaus ist die Position des Autos auf der Straße nicht eingeschränkt.

Auf welcher Linie kann das Rennauto die zwei geraden Stücken à 25 m und die Kurve schnellstmöglich, also in der kürzesten Zeit durchfahren? Ist diese Kurve von  $b$  abhängig? Wie?

Das Modell wird etwas realitätsnäher, wenn man für die tatsächliche Beschleunigung  $a$  und die tatsächliche Fliehkraft  $F$  die Beschränkung

$$\left( \frac{a}{a_{\max}} \right)^2 + \left( \frac{F}{F_{\max}} \right)^2 \leq 1$$

einführt.

**Das Kleingedruckte:** Die Berichte sollen ein umfassendes Bild von der Bearbeitung des Themas vermitteln, besonders auf numerische Probleme eingehen und die Ergebnisse darstellen. Sehr geschätzt sind auch Ergänzungen und weiterführende Überlegungen selbst dann, wenn dazu keine Rechnungen durchgeführt wurden.

Die Berichte müssen in  $\text{\LaTeX}$  geschrieben werden. Es dürfen im Quelltext ausschließlich `ascii`-Zeichen verwendet werden, also keine Umlaute und länderspezifischen Sonderzeichen. Der Quelltext muß mit einem gewöhnlichen  $\text{\LaTeX}$  übersetzbar sein. Dies gilt insbesondere für Benutzer von unterstützenden Editoren wie z. Bsp. `Lyx`. Sie können die Übersetzbarkeit des Quelltextes auf `hades` mit den Befehlen `latex`, `dvips` und `ghostview` selbst testen. Der Quelltext ist abzugeben.

Der Bericht soll so strukturiert sein, daß in einem übergeordneten File der Programmkopf und der Rahmen steht und dort eine Datei mit dem Namen `them07erg.tex` für Thema 7 usw. mittels des `\input`-Befehles eingebunden ist, die nur den eigentlichen Bericht enthält. Auf Gleichungen, Bilder und Abschnitte wird mit `\ref` verwiesen. Das dazugehörige Label endet mit der zweistelligen Nummer ihres Themas, dasselbe gilt für die Namen eventuell eingebundener Bilder. Diese müssen lesbar beschriftet sein und ausreichend dicke Linien enthalten. In Matlab erstellte Plots eignen sich nach `set(txt,'LineWidth',2)`. Harte Maßangaben und Zeilen- oder Seitenumbrüche sind nicht erlaubt, ebensowenig wie das Setzen von Zählern oder Styleoptionen. Für alle Formatierungen sind die vorhandenen  $\text{\LaTeX}$ -Befehle zu verwenden. Zusätzliche Style-files sind zu vermeiden.

Zugelassen sind Berichte in deutscher Sprache in einer konsistenten Orthographie. Die Programmtexte sind im allgemeinen nicht Bestandteil der Berichte. Die Berichte werden nach der Abgabe durchgesehen, und es werden eventuell vor ihrer Annahme Nachbesserungen verlangt. Für die Richtigkeit von Sachverhalten außerhalb der Mathematik und Physik wird keine Gewähr übernommen.

# 1 Vorbemerkungen

Offensichtlich ist hier das Problem der schnellstmöglichen Kurvendurchfahrt zu lösen. Die erste Anwendung, die jemandem dazu einfällt, ist wohl die bei Autorennen, dennoch spielen dieses oder zumindestens ähnliche Probleme auch bei anderen Anwendungen eine Rolle. So sind ähnliche Aufgabenstellungen auch im Straßenverkehr oder beim Bau von Fortbewegungsmitteln, z.B. Achterbahnen, zu lösen, auch wenn es hier primär um die Sicherheit geht. Im Straßenbau ist es notwendig Kurven möglichst so auszuformen, dass die Autos in der Spur bleiben können und die zulässige Höchstgeschwindigkeit so festzulegen, dass keine Gefährdung für die Verkehrsteilnehmer besteht. Jedoch werden diese Probleme eher intuitiv gelöst, da eine mathematische Analyse für jede Kurve suboptimal wäre. Beim Bau von Achterbahnen muss beachtet werden, dass die auftretenden G-Kräfte minimiert werden, hier müssen bereits vor dem Bau Berechnungen angestellt werden. Im Folgenden wird lediglich das Kurvenproblem beim Autofahren analysiert.

Das Problem der schnellstmöglichen Kurvendurchfahrt trat wohl schon bei den ersten Autorennen auf und wurde natürlich, den jeweiligen Bedingungen entsprechend, jedesmal mehr oder weniger optimal, intuitiv durch die Rennfahrer gelöst. Wie eine Kurve optimal zu durchqueren ist, weiß jeder der schon mal Auto, Motorrad oder Fahrrad gefahren ist. Sie wird möglichst vom äußeren Straßenrand angefahren, im Scheitelpunkt der Kurve das Streckeninnere berührt, um dann wieder an den äußeren Fahrbahnrand zu gelangen. Dabei wird bis zu einem gewissen Punkt abgebremst und dann ab diesen wieder beschleunigt. Das optimale Ergebnis steht insofern bereits fest und die folgenden Untersuchungen sollten auch zu keinem Anderen kommen. Interessant wird es jedoch sein zu beobachten, wie sich die optimale Kurvendurchfahrt verändert, wenn sich andere Einflüsse, wie z.B. Fahrzeugmasse oder die maximal wirkende Bremskraft verändern. Dies ist z.B. bei Regen oder bei einer anderen Fahrbahnoberfläche der Fall.

## 2 Definition der Zielstellung

Die Aufgabenstellung sieht vor, dass ein Fahrzeug mit einer Masse von 600 kg und einer Breite von 2 m gegeben ist. Desweiteren existieren Beschränkungen der wirkenden Kräfte, wie Zentrifugalkraft oder Luftwiderstand (die hier nicht weiter berücksichtigt werden). Außerdem sind Restriktionen bezüglich der Beschleunigung und der Bremsverzögerung gegeben. Ziel ist es eine 90°-Kurve mit Kurveninnenradius von 20 m schnellstmöglich zu durchfahren. Zusätzlich besteht die Forderung; dass jeweils 25 m vor und nach der Kurve wieder parallel zur Fahrbahn gefahren werden muss. Dabei ist die Straße an jeder Stelle genau 10 m breit. Es liegt also ein klassisches Optimierungsproblem vor, wobei der zu optimierende Parameter die Zeit ist.

Ziel wird es also sein, dieses Optimierungsproblem mathematisch zu verfassen und eine numerische Lösung zu erarbeiten. Dabei wird das Problem unter den gegebenen Bedingungen gelöst und analysiert, was geschieht, wenn die gegebenen Parameter modifiziert werden.

Weiterhin wird der geometrische Ort der optimalen Streckenführung dargestellt. Der Weg, auf dem das Auto die Strecke am schnellsten durchfahren kann, wird im Folgenden mit *Ideallinie* (wie auch aus dem Rennsport bekannt) bezeichnet.

Weiterhin werden im Folgenden die unten aufgeführten Größen durch entsprechende Abkürzungen

dargestellt.

$m$	:=	Fahrzeugmasse in Kilogramm
$s$	:=	Weg in Meter
$t$	:=	Zeit in Sekunden
$r$	:=	Kreisradius in Meter
$v$	:=	aktuelle Geschwindigkeit in $\frac{m}{s}$
$v_{max}$	:=	Höchstgeschwindigkeit in $\frac{m}{s}$
$b$	:=	const. $\in \mathbb{R}_+$
$F$	:=	Kraft in Newton
$F_{max}$	:=	maximal zulässige Kraft in Newton
$a$	:=	Beschleunigung in $\frac{m}{s^2}$
$a_{max}$	:=	maximale Beschleunigung in $\frac{m}{s^2}$

### 3 Physikalische Grundlagen

Bei den nachfolgenden Überlegungen wird auf verschiedene grundlegende physikalische Formeln zurückgegriffen, die im Folgenden aufgeführt sind.

Zur Berechnung der Zentrifugalkraft, bei gegebener Masse, Geschwindigkeit und Kreisradius wird die gängige Formel

$$F = \frac{mv^2}{r} \quad (1)$$

verwendet. Zur Bestimmung der Beschleunigung zwischen 2 Punkten mit gegebener Anfangs- ( $v_0$ ) und Endgeschwindigkeit ( $v_E$ ), bei gegebener verstrichener Zeit zwischen beiden Punkten, dient die Formel

$$a = \frac{v_0 - v_E}{t} \Leftrightarrow t = \frac{v_E - v_0}{a} \Leftrightarrow v_E = at + v_0. \quad (2)$$

Zur Bestimmung des zurückgelegten Weges bei gegebener Beschleunigung, vergangener Zeit und Geschwindigkeit wird

$$s = \frac{at^2}{2} + vt \quad (3)$$

verwendet.

## 4 Numerische Lösung mittels MatLab

### 4.1 Herangehensweise

Im Folgenden wird beschrieben, wie die Fahrbahn und die Ideallinie simuliert werden. Weiterhin werden Algorithmen hergeleitet, die es ermöglichen die gegebenen restriktiven Bedingungen, insbesondere die physikalischen, einzuhalten. Danach werden Optimierungsverfahren beschrieben, die es möglich machen, die Ideallinie anzunähern.

## 4.2 Fahrbahn und Ideallinien-Modellierung

Zur 2-dimensionalen Darstellung der Strecke wird ein kartesisches Koordinatensystem, in dem die Fahrbahn eingezeichnet ist, benutzt. Dabei entspricht 1 LE in der Zeichnung  $\cong$  1 Meter in der Realität. (siehe Abbildung 1)

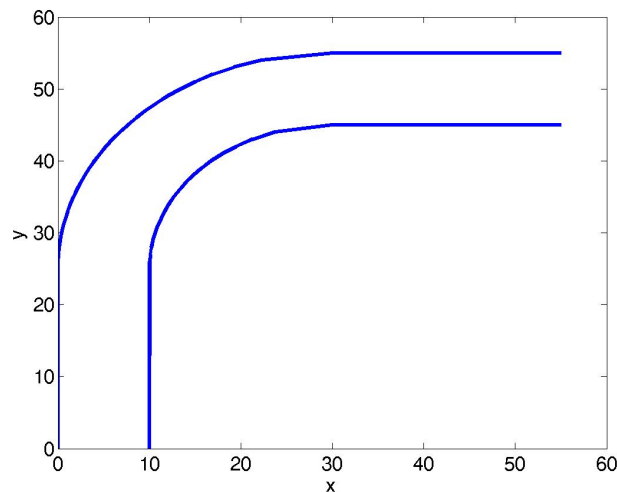


Abbildung 1: Modellierung der Fahrbahn

Dabei setzt sich die äußere bzw. innere Fahrbahnbegrenzung aus jeweils einer Geraden am Anfang und Ende, sowie einem Viertelkreis zur Darstellung des Kurvengebietes, zusammen. Die Geraden haben eine Länge von 25 LE und der Viertelkreis einen Innenradius von 20 LE.

Die Ideallinie wird mit Hilfe von Stützstellen  $x_i$ ,  $i = \{1, \dots, n\}$ , die nicht notwendigerweise äquidistant auf der Strecke verteilt sind, simuliert. Die einzelnen Stützstellen werden durch eine Gerade verbunden und nähern dadurch die Ideallinie linear an. Eine Verschiebung der Stützstellen erfolgt jeweils nur senkrecht zur Fahrbahn, womit ausgeschlossen werden kann, dass sich Stützstellen überkreuzen, bzw. dass nicht in umgedrehter Richtung gefahren wird.

Die ersten beiden, sowie die letzten beiden Stützstellen werden parallel zur Fahrbahn ausgerichtet sein, wodurch realisiert wird, dass das Auto spätestens 25 m vor bzw. nach der Kurve wieder parallel zur Fahrbahn fährt.

## 4.3 Berechnung der Fliehkraft

Da die Strecke nur linear angenähert wird, besteht das Problem, dass bei den einzelnen Stützstellen eine unendlich große Krümmung bestehen kann und somit unendlich große Fliehkräfte wirken, während zwischen den einzelnen Stützstellen keinerlei Fliehkräfte wirken, da es sich um Geraden handelt.

Zur Lösung dieses Problems wird nicht jede Stützstelle einzeln betrachtet, sondern es werden zur Bestimmung der Krümmung 3 Stützstellen herangezogen. Diese 3 Stützstellen dienen anschaulich gesehen als Eckpunkte eines Dreiecks. Um dieses Dreieck herum wird der Umkreis konstruiert, wodurch

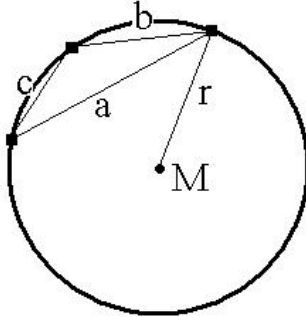


Abbildung 2: Radiusberechnung mittels Umkreis

es möglich ist den Radius zu berechnen um später mit Hilfe von (1) die Fliehkraft zu ermitteln. (Abbildung 2)

Dabei werden folgende Formeln als Grundlage verwendet

$$r = \frac{abc}{4A}, \quad A = \sqrt{s(s-a)(s-b)(s-c)}, \quad s = \frac{a+b+c}{2}.$$

Diese Rechnung wird für alle Stützstellen  $x_i, i \in \{2, \dots, n-1\}$  durchgeführt. Das Ergebnis wird für jedes  $x_i$  in einem Hilfsvektor  $r$  abgespeichert. Dies ist lediglich für  $n-2$  Stützstellen notwendig, da der vorhergehende Verlauf, bzw. der folgende Verlauf der Ideallinie außerhalb des untersuchten Bereiches nicht bekannt ist.

Weiterhin besteht das Problem, dass einige Punkte auch auf einer Geraden liegen können, dieser Fall, also wenn  $A = 0$ , muss abgefangen werden.

In einem Hilfsvektor  $v$  wird für jedes  $x_i$  die aktuelle Geschwindigkeit gespeichert. Am Anfang werden alle Einträge des Hilfsvektors auf  $v_{max}$  gesetzt, also  $v(x_i) = v_{max} \forall i \in \{1, \dots, n\}$ , wodurch es möglich ist, mit Hilfe von (1), die wirkende Kraft in  $x_i$  zu ermitteln.

Da nun für alle  $x_i, i \in \{2, \dots, n-1\}$  die wirkende Kraft bekannt ist, kann festgestellt werden, ob an einer Stelle zu große Kräfte wirken. Falls dem so ist, wird die aktuelle Geschwindigkeit in  $x_i$  so gesetzt, dass  $F(x_i) \leq F_{max} \forall i \in \{1, \dots, n\}$ , also

$$v(x_i) = \sqrt{\frac{F_{max} \cdot r}{m}} \quad i = \{1, \dots, n\}.$$

Weiterhin wird die maximal zulässige Geschwindigkeit in einem weiteren Hilfsarray  $v_{max}$  vermerkt.

#### 4.4 Simulieren des Bremsverhaltens

Da der Geschwindigkeitsunterschied zwischen benachbarten Stützstellen zuweilen sehr groß sein kann und nur begrenzte Verzögerungskapazität zur Verfügung steht, ergibt sich die Notwendigkeit, die Geschwindigkeit der vorhergehenden Stützstelle herunter zu setzen.

Zur Umsetzung dieser Lösung bietet sich ein rekursiver Algorithmus an, der alle Stützstellen, von hinten beginnend, durchgeht. Dabei wird für alle  $x_i$ ,  $i \in \{2, \dots, n\}$  die Geschwindigkeit mit der vorhergehenden Stützstelle verglichen. Findet zwischen den beiden Stützstellen eine größere Verzögerung als  $2b$  statt, so muss die Geschwindigkeit der vorhergehenden Stützstelle heruntergesetzt werden.

Zur effektiven Berechnung stehen (2) und (3) zur Verfügung. Die Formel (2) kann direkt in (3) eingesetzt werden und man gelangt zu

$$s = \frac{a}{2} \cdot \left( \frac{v_E - v_0}{a} \right)^2 + v_0 \cdot \left( \frac{v_E - v_0}{a} \right) = \frac{v_E^2 - v_0^2}{2a}.$$

Umgestellt nach  $v_0$  liefert dies unter Zuhilfenahme von  $a = -2b$

$$v_0 = \sqrt{v_E^2 - 2as} = \sqrt{v_E^2 + 4bs}.$$

Womit es möglich ist die Geschwindigkeiten des Hilfsvektors  $v$  zu aktualisieren, also

$$v(x_{i-1}) = v_{max}(x_{i-1}) = \min \left( \sqrt{4bs + v(x_i)^2}, v_{max}(x_{i-1}) \right) \quad i = n, \dots, 2.$$

Gleichzeitig wird der Hilfsvektor  $v_{max}$  aktualisiert.

#### 4.5 Simulieren der Beschleunigung

Wiederum besteht das Problem, dass der Geschwindigkeitsunterschied zwischen benachbarten Stützstellen sehr groß sein kann. Da jedoch nur begrenzt beschleunigt werden kann, muss die Geschwindigkeit der nachfolgenden Stützstelle eventuell herunterreguliert werden.

Bei der Umsetzung wird ein rekursiver Algorithmus verwendet. Dieser geht alle Stützstellen, beginnend bei der Ersten durch und misst die Beschleunigung, die zwischen den Stützstellen wirkt. Ist diese größer als  $a_{max}$ , so wird die Geschwindigkeit der folgenden Stützstelle herunterreguliert.

Dabei ist  $a$  gegeben durch

$$a_{max} = b \left( 1 - \frac{v_0}{v_{max}} \right).$$

Die Gleichung (2) in (3) eingesetzt und aufgelöst nach  $v_E$  ergibt

$$v_E = \sqrt{v_0^2 + 2as}.$$

Nach Einsetzen von  $a$  gelangt man zu

$$v_E = \sqrt{\frac{v_0^2 v_{max} + 2bsv_{max} - 2bsv_0}{v_{max}}}.$$

Womit man in der Lage ist den Geschwindigkeitsvektor zu aktualisieren, also

$$v(x_{i+1}) = \min \left( \sqrt{\frac{v(x_i)^2 v_{max} + 2bsv_{max} - 2bsv(x_i)}{v_{max}}}, v_{max}(x_{i+1}) \right) \quad i = 1, \dots, n-1.$$

## 4.6 Simulation der Realitätsnahen Bedingung

Zur besseren Annäherung an die Realität, muss die Ungleichung

$$\left( \frac{a}{a_{max}} \right)^2 + \left( \frac{F}{F_{max}} \right)^2 \leq 1 \quad (4)$$

in jedem Abschnitt der Strecke erfüllt sein. Da die Strecke abschnittsweise bestimmt wurde, muss in jedem Abschnitt geprüft werden, ob diese Bedingung erfüllt ist. Weil die Beschleunigung jeweils der vorhergehenden Stützstelle zugeordnet wird, sind 2 Fälle zu unterscheiden, zum einen die Einhaltung der Bedingung beim Bremsen und zum anderen die Einhaltung beim Beschleunigen.

### 4.6.1 Realitätsbedingung beim Bremsen

Die Formel (4) kann umgeschrieben werden zu

$$\left( \frac{a}{a_{max}} \right)^2 + \left( \frac{F}{F_{max}} \right)^2 = \left( \frac{a}{2b} \right)^2 + \left( \frac{\frac{mv_0^2}{r}}{F_{max}} \right)^2 \leq 1.$$

Umstellen nach  $a$  liefert

$$a = 2b \cdot \sqrt{1 - \frac{m^2 v_0^4}{r^2 F_{max}^2}}.$$

Durch Einsetzen in (3) zusammen mit (2) und anschließendem Umstellen nach  $v_0$  gelangt man zu

$$v_0 = \sqrt{\frac{r^2 F_{max}^2 v_E^2 + 4\sqrt{r^2 F_{max}^2 s^2 b^2 (16m^2 s^2 b^2 - m^2 v_E^4 + r^2 F_{max}^2)}}{16m^2 s^2 b^2 + r^2 F_{max}^2}}.$$

Dadurch ist es möglich die realistische Geschwindigkeit der vorhergehenden Stützstelle aufgrund der folgenden Stützstelle zu berechnen. Es folgt für  $i = n, \dots, 2$

$$v(x_{i-1}) = \min \left( \sqrt{\frac{r(x_i)^2 F_{max}^2 v(x_i)^2 + 4\sqrt{r(x_i)^2 F_{max}^2 s^2 b^2 (16m^2 s^2 b^2 - m^2 v(x_i)^4 + r(x_i)^2 F_{max}^2)}}{16m^2 s^2 b^2 + r(x_i)^2 F_{max}^2}}, v_{max}(x_{i-1}) \right).$$

Da es sich offensichtlich um einen rekursiven Algorithmus handelt, kann dieser in den vorherigen Algorithmus des 'normalen' Verzögerns eingebunden werden. Die Geschwindigkeit der vorhergehenden Stützstelle ist somit das Minimum der Werte des 'normalen' und 'realen' Bremsalgorithmusses.

#### 4.6.2 Realitätsbedingung beim Beschleunigen

Wiederum folgt aus (4)

$$\left(\frac{a}{a_{max}}\right)^2 + \left(\frac{F}{F_{max}}\right)^2 = \left(\frac{a}{b\left(1 - \frac{v}{v_{max}}\right)}\right)^2 + \left(\frac{\frac{mv_0^2}{r}}{F_{max}}\right)^2 \leq 1,$$

dass

$$a = \sqrt{\left(1 - \frac{m^2 v_0^4}{r^2 F_{max}^2}\right) \left(1 - \frac{v_0}{v_{max}}\right)} b.$$

Daraus folgt wiederum

$$v_E = \sqrt{v_0^2 + 2bs\sqrt{1 - \frac{m^2 v_0^4}{r^2 F_{max}^2}} \left(1 - \frac{v_0}{v_{max}}\right)}.$$

Für den Geschwindigkeitsvektor ergibt sich demnach

$$v(x_{i+1}) = \min \left( \sqrt{v(x_i)^2 + 2bs\sqrt{1 - \frac{m^2 v(x_i)^4}{r(x_i)^2 F_{max}^2}} \left(1 - \frac{v(x_i)}{v_{max}}\right)}, v_{max}(x_{i+1}) \right) \quad i = 1, \dots, n-1.$$

Für die Geschwindigkeit der nachfolgenden Stützstelle ist ebenfalls wieder das Minimum aus 'normalen' und 'realen' Bremsalgorithmus zu nehmen.

### 4.7 Bestimmung der Fahrtzeit

Zur Bestimmung der Fahrtzeit  $t_{gesamt}$  werden die Formeln (2) und (3) benutzt. Es folgt

$$t_{gesamt} = \sum_{i=1}^{n-1} t_i = \sum_{i=1}^{n-1} \frac{2s_i}{v(x_i) + v(x_{i+1})}.$$

## 5 Optimierungsalgorithmen

### 5.1 Lösung mittels MatLab-Optimization Toolbox

In der MatLab-Optimization Toolbox finden sich eine Vielzahl unterschiedlicher Funktionen. Da die Stützstellen als Vektor darstellbar sind und feste Grenzen vorgegeben sind, in denen sich die  $x_i$  bewegen können, ist die Funktion `fmincon()` zu verwenden.



Dabei benötigt `fmincon()` eine zu minimierende Funktion  $\phi$  mit

$$\phi : \mathbb{R}^{2n} \rightarrow \mathbb{R} \quad (x_1, \dots, x_{2n})^T \mapsto t_{gesamt}.$$

Die Zeit zu den vorgegebenen Stützstellen wird durch  $\phi$  berechnet. Danach werden durch `fmincon()` neue Stützstellen vorgegeben und erneut berechnet. Dadurch kommt `fmincon()` nach (hoffentlich) endlich vielen Schritten zu einer optimalen Lösung.

Da bei diesem Algorithmus nur ein Vektor vorgegeben werden kann, bietet es sich an die Stützstellen eindimensional darzustellen, z.B. durch den Abstand zur Fahrbahnninnenseite. Die Berechnungen zur 2-dimensionalen Darstellung müssen dann in  $\phi$  vollzogen werden. Gleichzeitig wird dadurch sichergestellt, dass sich Stützstellen nicht schneiden können und diese senkrecht zur Fahrbahn verschoben werden.

Problem bei diesem Algorithmus ist, dass die Güte der Approximation von der vorgegebenen Zahl der Stützstellen abhängig ist. Bei vielen Stützstellen tut sich `fmincon()` schwer eine optimale Lösung zu finden. Zur Behebung dieses Problems werden erst weniger Stützstellen vorgegeben, anschließend neue Stützstellen zwischen den Alten eingefügt und der Algorithmus erneut gestartet. So wird die Güte der Approximation verbessert. Dies sollte nicht all zu oft wiederholt werden, da der Zeitaufwand durch diese 'Verdopplung' ebenfalls mindestens verdoppelt wird. Genauer gesagt, kommt es bei wenigen Stützstellen zu einer Verdopplung der Zeit, während es bei einer bereits hohen Anzahl von Stützstellen und anschließender Verdopplung zu einer Verdreifachung des Zeitaufwandes kommt.

Weiterhin besteht das Problem, dass die Lösung stark vom Anfangsvektor abhängig ist, so wird die Ideallinie besser approximiert, wenn der Startvektor nicht so stark von der optimalen Lösung abweicht. Dies wurde ebenfalls durch die dynamische Anzahl der Stützstellen behoben.

## 5.2 Lösung mittels Konvergenzalgorithmus

Wie der Name schon sagt, wird die Ideallinie mit Hilfe eines Konvergenzalgorithmusses beliebig gut approximiert. Die Grundidee, die dahinter steckt, ist, dass die Ideallinie von einer beliebig vorgegeben Linie immer weiter angenähert wird. Dabei wird eine Ausgangslinie vorgegeben und für diese die Zeit ermittelt, das wird für eine zweite Linie wiederholt, unter der Variation eines beliebigen Stützstellenpunktes. Eine solche Veränderung kann in 2 Richtungen erfolgen, zum Inneren der Fahrbahn oder zum Äußeren der Fahrbahn. Führt dies zu einem schlechteren Ergebnis, so wird der Punkt wieder auf seine Ausgangsposition gesetzt und die Veränderung, sowie die Richtung der Veränderung des Punktes in einem Hilfsvektor gespeichert. Damit kann ausgeschlossen werden, dass diese Stützstelle im nächsten Durchlauf noch einmal in die selbe Richtung verändert wird. Dies wird solange wiederholt, bis alle Stützstellenpunkte an der Reihe waren. Führt eine Veränderung eines Punktes zu einer Verbesserung der Zeit, so startet der Algorithmus erneut und alle Einträge des Hilfsvektors werden gelöscht. Sind alle Punkte in alle möglichen Richtungen verändert worden und trat bei keinem eine Verbesserung der Zeit auf, so wird die Schrittweite der Veränderung der Punkte halbiert und wieder für jeden Punkt neu getestet.

Dies wird solange fortgesetzt, bis die Schrittweite kleiner als ein gewisses  $\epsilon$  ist. Dann terminiert der Algorithmus, da eine Verbesserung eines beliebigen Punktes um dieses  $\epsilon$  nichts mehr bringt.

Probleme bei dem bisher beschriebenen Algorithmus sind jedoch die Folgenden:

- bei einer hohen Anzahl von Stützstellen ist die Rechenzeit extrem hoch
- bei einer hohen Anzahl von Stützstellen, muss das  $\epsilon$  um so kleiner gewählt werden, damit eine Konvergenz auftritt
- es wird jeweils nur eine Stützstelle verändert, dabei treten zwangsläufig 'Ecken' auf, die zu Geschwindigkeitsverlust führen und damit zu einer schlechteren Zeit

Die Folge dieser Probleme ist eine Nicht-Konvergenz des Algorithmuses. Um dieses Problem zu beheben wurde nicht von einer fixen Anzahl von Stützstellen ausgegangen, sondern von einer dynamischen Anzahl. So wurde der oben beschriebene Algorithmus erst verwendet, um für 6 Stützstellen den optimalen Weg zu ermitteln. Danach wurden jeweils zwischen den vorhandenen Stützstellen neue eingeführt, natürlich so, dass sich diese auf dem zulässigen Teil der Fahrbahn befinden. Die Anzahl der Stützstellen wurde sukzessive erhöht und der obige Algorithmus erneut angewendet. Dies wurde solange wiederholt bis eine einigermaßen zufriedenstellende Anzahl von Stützstellen erreicht wurde, in dem Sinne, dass die Kurve nicht mehr so eckig aussieht.

Problem dieses Algorithmuses ist, dass dieser bei hohen Werten von  $b$  oder anderen veränderten Werten, Rundungen entwickelt, die intuitiv von der Ideallinie abweichen. Da der Algorithmus nur für 'reale' Werte von  $b$  arbeitet und somit eine tiefergehende Analyse der Auswirkungen des Parameters  $b$  auf die Ideallinie nicht möglich ist, wird für die folgende Auswertung, der Optimierungsalgorithmus `fmincon()` verwendet.

## 6 Auswertung

Wie vermutet, kommt man zu dem Ergebnis, dass die intuitiv gedachte Lösung auch die Optimale ist. Die Kurve wird vom äußeren Fahrbahnrand angefahren, dann wird im Scheitelpunkt der Kurve das Fahrbahnnere berührt, um anschließend wieder am Kurvenausgang zur äußeren Fahrbahnbegrenzung zu gelangen. Dabei setzt sich der Weg aus einer Verzögerungsphase und einer anschließenden Beschleunigungsphase zusammen.

Im Folgenden sind die Ergebnisse für verschiedene Parameter bezüglich modifizierter Bremsbeschleunigung, Einhaltung der Realitätsbedingung, Masse und maximal zulässiger Kraft dargestellt.

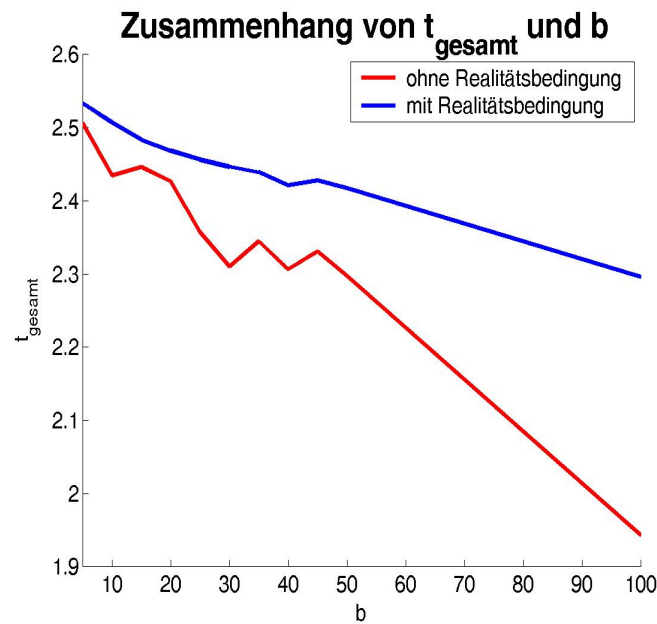


Abbildung 3: Zusammenhang  $t_{gesamt}$  und  $b$

b	m	$F_{max}$	$v_{max}$	real	$t_{gesamt}$	Bemerkungen
5	600	18000	100	ohne	2,5059	
5	600	18000	100	mit	2,5336	
10	600	18000	100	ohne	2,4344	
10	600	18000	100	mit	2,5071	
15	600	18000	100	ohne	2,4460	
15	600	18000	100	mit	2,4835	
20	600	18000	100	ohne	2,4264	
20	600	18000	100	mit	2,4684	
25	600	18000	100	ohne	2,3567	
25	600	18000	100	mit	2,4562	
30	600	18000	100	ohne	2,3099	
30	600	18000	100	mit	2,4466	
35	600	18000	100	ohne	2,3450	
35	600	18000	100	mit	2,4388	
40	600	18000	100	ohne	2,3063	
40	600	18000	100	mit	2,4213	
45	600	18000	100	ohne	2,3310	
45	600	18000	100	mit	2,4276	
50	600	18000	100	ohne	2,2977	
50	600	18000	100	mit	2,4177	
100	600	18000	100	ohne	1,9433	
100	600	18000	100	mit	2,2959	
22	600	27000	95	mit	2,0311	Formel 1, trockener Asphalt*
10	600	12000	95	mit	3,0373	Formel 1, nasser Asphalt*
4	1300	11000	55	mit	4,722	Golf, trockener Asphalt*
2	1300	8000	55	mit	5,6387	Golf, nasser Asphalt*
4	200	4000	55	mit	3,0995	Motorrad, trockener Asphalt*
3	40000	200000	30	mit	5,0531	LKW, trockener Asphalt*

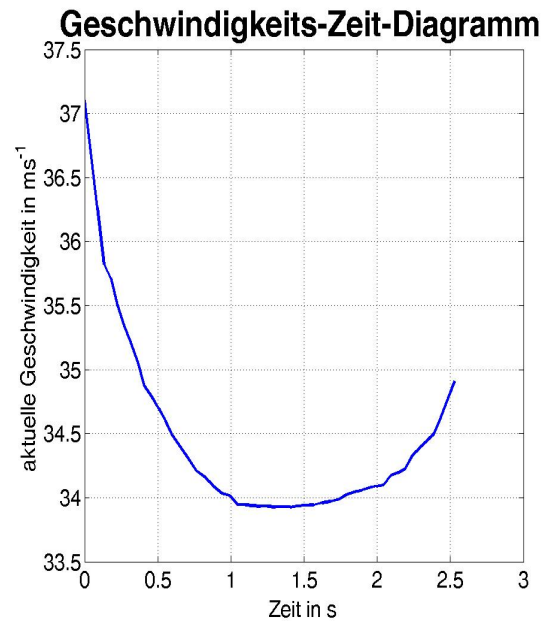
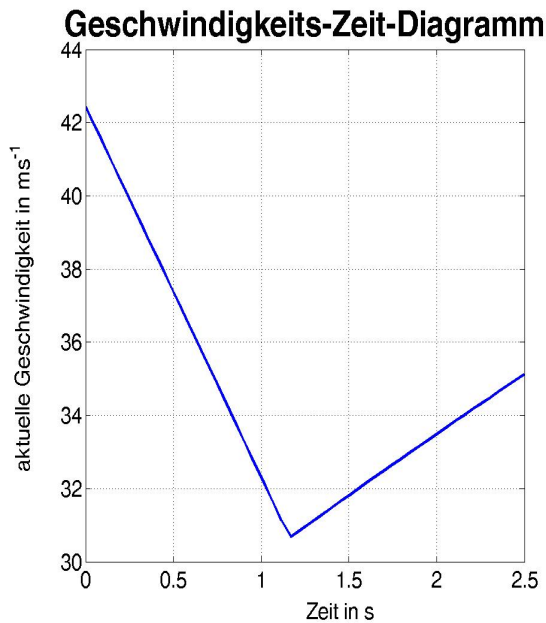


Abbildung 4: v-t-Diagramme für  $b = 5$ , ohne und mit Realitätsbedingung

\* Werte aus [2] und von [www.racingtrack.de](http://www.racingtrack.de)

Es ist festzustellen, dass die benötigte Zeit sowohl von  $b$  als auch von der Realitätsbedingung abhängig ist. Ohne Realitätsbedingung sind die Zeiten bei gleichem  $b$  durchweg besser. Betrachtet man das Geschwindigkeits-Zeit-Diagramm so wird der Grund für die Differenz klar. Unter Einhaltung der Realitätsbedingung kommt es zu einer nicht so starken Beschleunigung, bzw. Verzögerung, wodurch natürlich dann auch  $t_{gesamt}$  kleiner wird. Die Veränderungen geschehen 'sanfter' und nicht so 'ruckartig'. (siehe Abbildung 4)

Weiterhin ist festzuhalten, dass die Zeiten bei steigendem  $b$  besser werden, wie in Abbildung 3 zu erkennen ist. Man beachte, dass in Abbildung 3 keine Steigungen vorhanden sein sollten. Zurückzuführen sind diese Abweichungen auf einen Abbruch der Optimierung durch `fmincon()`. Die Optimierung wird beendet, sobald eine vorgegebene  $\epsilon$ -Schranke, welche angibt, wie weit die letzten Ergebnisse voneinander abweichen, unterschritten wird. Durch die unterschiedlichen Werte, die  $b$  annimmt, findet somit mal eine etwas exaktere Approximation und mal eine etwas schlechtere Approximation der Ideallinie statt, daher die 'Ecken' in Abbildung 3. Eine Verkleinerung von  $\epsilon$  brachte ebenfalls Unregelmäßigkeiten hervor. Jedoch bleibt insgesamt erkennbar, dass mit steigendem  $b$  auch  $t_{gesamt}$  fällt. Betrachtet man wiederum das Geschwindigkeits-Zeit-Diagramm so erkennt man auch die Ursache für dieses Verhalten. Bei einem höheren  $b$  kann stärker beschleunigt, sowie auch stärker verzögert werden. Dementsprechend kann die Kurve mit einer höheren Geschwindigkeit angefahren und wieder verlassen werden. (siehe Abbildung 5)

Außerdem ist festzustellen, dass die Ideallinie ebenfalls von den Parametern  $b$  und der Realitätsbedingung abhängig ist, jedoch in unterschiedlicher Art und Weise. Bei einer Veränderung von  $b$  mit Realitätsbedingung verändert sich die Ideallinie nur minimal. Sie verändert sich leicht zum Kurveninneren hin. (siehe Abbildung 6)

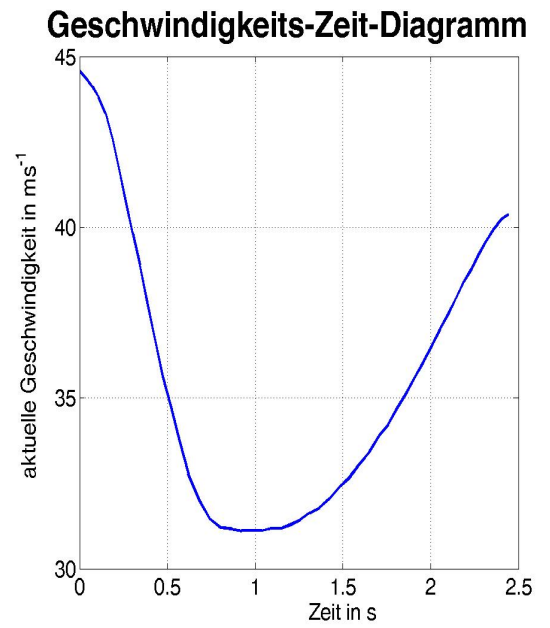
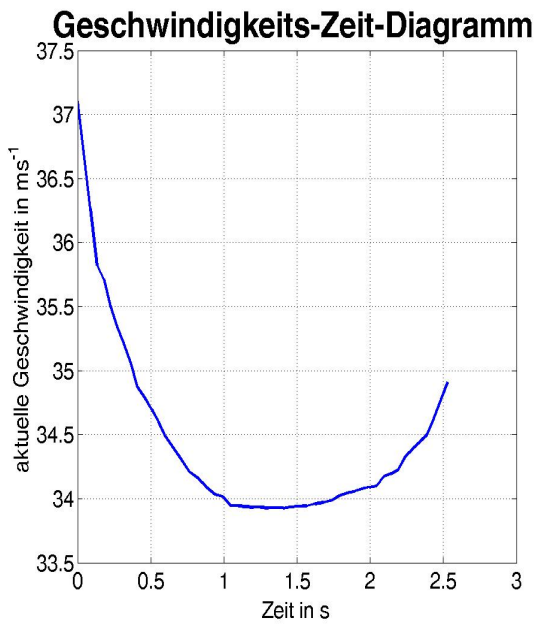


Abbildung 5: v-t-Diagramme für  $b = 5$  und  $b = 30$ , mit Realitätsbedingung

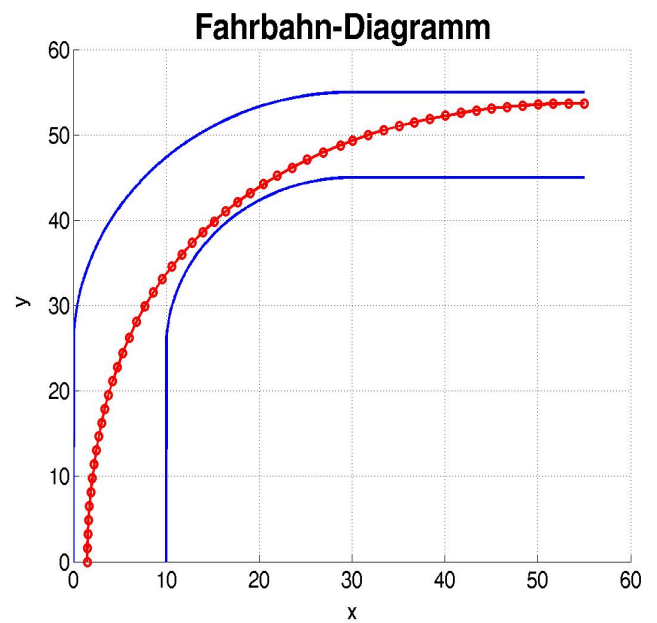
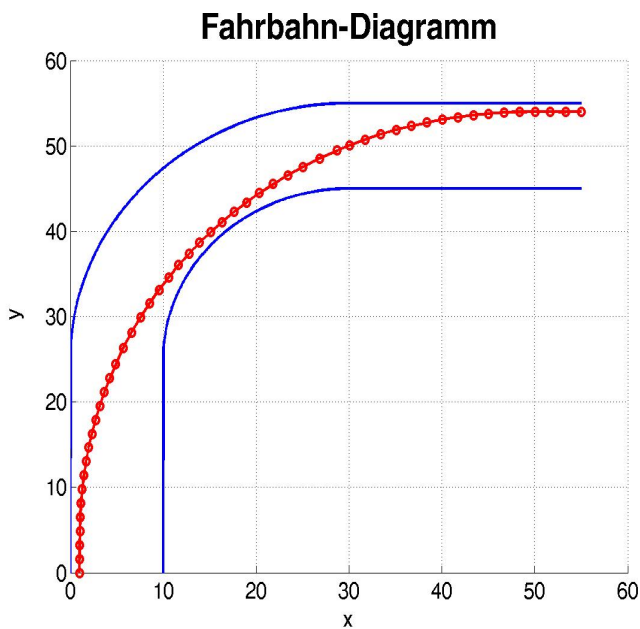


Abbildung 6: Verlauf der Ideallinie für  $b = 5$  und  $b = 30$ , mit Realitätsbedingung

Deutlich erkennbar wird eine Veränderung erst bei großen, unrealistischen Werten von  $b$ , z.B. für  $b = 50$ . Dies entspricht im Modell einer Verzögerung von  $100 \text{ m s}^{-2}$ , also ungefähr 10-facher Erdbeschleunigung. Dies ist selbst mit einem Formel 1 - Rennwagen nicht zu schaffen, dieser schafft im besten Fall  $4,5 G$ . Zu erklären ist dieses Verhalten damit, dass durch die verbesserten Verzögerungs- und Beschleunigungswerte weniger die Minimierung der Krümmung im Vordergrund steht, als viel mehr die Minimierung des Weges. (siehe Abbildung 7)

Betrachtet man das Ganze ohne Realitätsbedingung, so sieht die Sache schon etwas anders aus. Vergrößert man  $b$  so lässt sich bereits eine klare Veränderung der Ideallinie beobachten, für hohe  $b$  neigt sie sehr viel mehr zum Kurveninneren als mit Realitätsbedingung. (siehe Abbildung 8)

Erhöht man  $b$  auf unrealistisch hohe Werte, so wird die Ideallinie eckig. Dies ist damit zu erklären, dass in diesem Fall für einen Bruchteil einer Sekunde die Geschwindigkeit von  $v_{max}$  auf fast 0 reduziert werden kann, wobei in diesem Moment die Richtungsänderung erfolgt, und dann sofort wieder auf  $v_{max}$  beschleunigt wird.

Interessant ist nun ein Vergleich zwischen einem Formel 1 Rennauto ( $m = 600 \text{ kg}$ ,  $v_{max} = 95 \text{ m s}^{-1}$ ,  $F_{max} = 27000 \text{ N}$ ,  $b = 22$ ) und einem normalen PKW (z.B. der Golf 4 ( $m = 1300 \text{ kg}$ ,  $v_{max} = 55 \text{ m s}^{-1}$ ,  $F_{max} = 1100 \text{ N}$ ,  $b = 4$ )), wie in Abbildung 9 geschehen.

Auffällig ist, dass die Ideallinie nahezu die gleiche ist, unterschiedlich ist nur die Zeit.

Daher kann man nur zu dem Resultat kommen, dass in der Realität, sofern die Straßenverhältnisse auf der gesamten Fahrbahn gleich sind, die Ideallinie eindeutig festgelegt ist, egal welches Fahrzeug betrachtet wird.

## 7 Kritik am Modell

Es ist erkennbar, dass das vorgegebene Modell der Aufgabenstellung nur für bestimmte Rahmenbedingungen zutrifft. Es sind starke Interdependenzen erkennbar, wenn man nicht eben die Werte eines Rennautos benutzt, z.B. die eines Golfs, so ist eine Zeit von knapp 5,6 s bei Regen nicht sehr realistisch. Deutlichere Unterschiede sind beim Vergleich zu einem LKW erkennbar.

Um ein allgemeineres und besseres Programm zur Kurvenberechnung zu verfassen ist daher ein detaillierteres Modell nötig, das zwischen den verschiedenen Rahmenbedingungen einerseits und zwischen unterschiedliche Fahrzeugtypen andererseits differenziert.

Die wirkenden Kräfte, ausgenommen Brems- und Beschleunigungskräfte wurden vollkommen außer acht gelassen.

Desweiteren werden im gesamten Streckenbereich identische Fahrbahnbeschaffenheiten vorausgesetzt, was nicht der Realität entspricht. So kann die Fahrbahn Löcher oder Verunreinigungen auf der berechneten Ideallinie aufweisen.

Das Fahrzeug wird als Punktmasse und nicht als Auto auf 4 Rädern betrachtet. In der Realität wirken auf die verschiedenen Räder unterschiedlich große Kräfte, was das Bremsverhalten und die Beschleunigung beeinflusst.

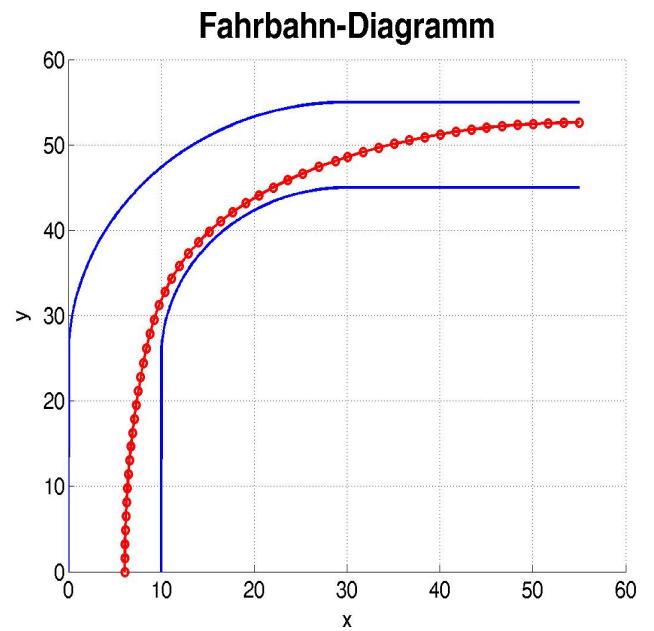
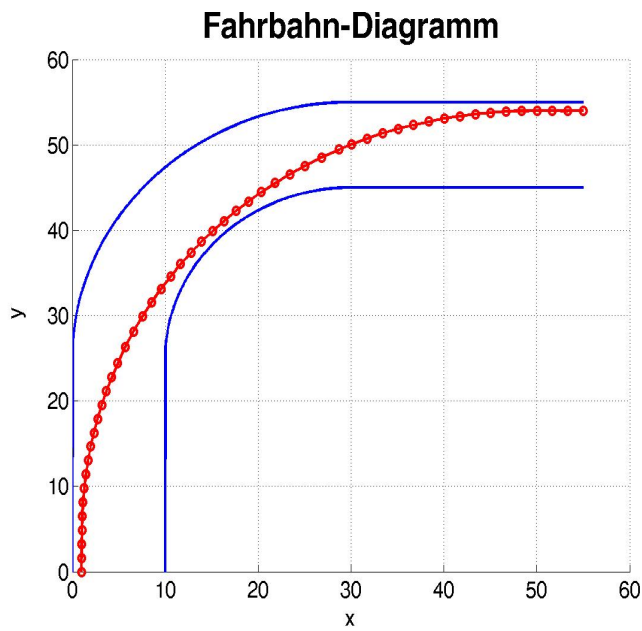


Abbildung 7: Verlauf der Ideallinie für  $b = 5$  und  $b = 50$ , mit Realitätsbedingung

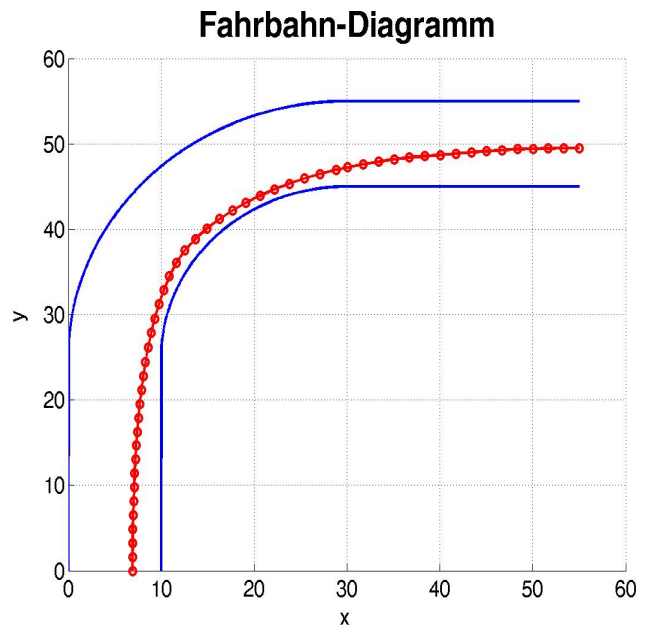
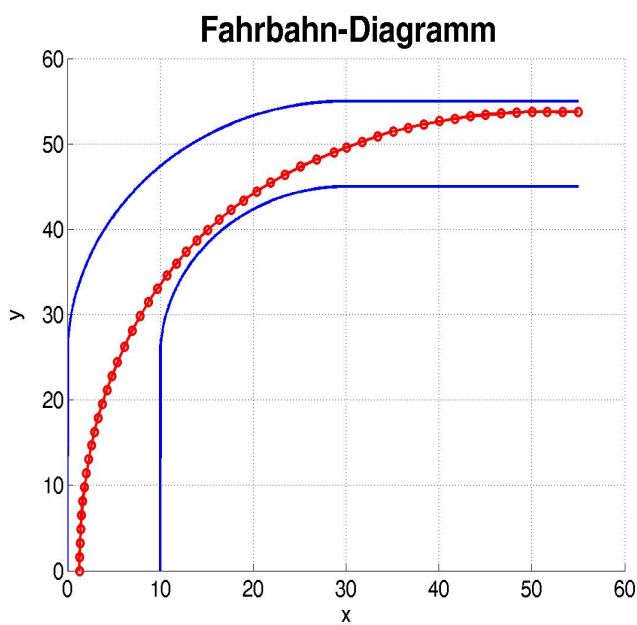


Abbildung 8: Verlauf der Ideallinie für  $b = 5$  und  $b = 30$ , ohne Realitätsbedingung

## 8 Verbesserungsvorschläge

Die größte Schwäche der Lösung ist sicherlich die lineare Annäherung der Ideallinie. Dadurch wurden Geschwindigkeiten und Beschleunigungen nur jeweils einem einzigen Punkt zugeordnet. Die Punkte zwischen den Stützstellen wurden somit außer Acht gelassen. Besser wäre eine Approximation der Ideallinie mittels glatter Funktionen oder Splines, auch wenn sich die Rechenzeit dadurch extrem vergrößern würde. Zu nennen wären hier kubische Splines, die ohnehin bereits die Eigenschaft besitzen die auftretenden Fliehkräfte zu minimieren.

Das Problem der Krümmung wurde nur vereinfacht durch eine Kreisannäherung gelöst. Die Zeitberechnung erfolgte mit Hilfe der Durchschnittsgeschwindigkeit zwischen 2 Stützstellen, wodurch die ermittelte Fahrzeit zu niedrig ist, da die Beschleunigung quadratisch und nicht linearer Natur ist. Dies ist daran erkennbar, dass bei einer Hinzunahme von weiteren Stützstellen die benötigte Zeit anwächst.

## Literatur

- [1] *“Das große Tafelwerk”*, Volk und Wissen, Berlin, 1999
- [2] Bert Breuer, Karlheinz H. Bill, *“Bremshandbuch: Grundlagen, Komponenten, Systeme, Fahr-  
dynamik”*, Vieweg+Teubner Verlag, 2006



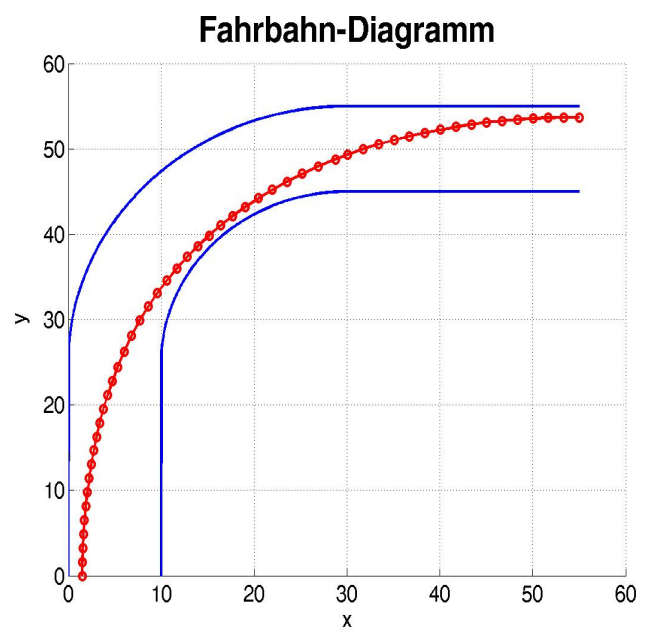
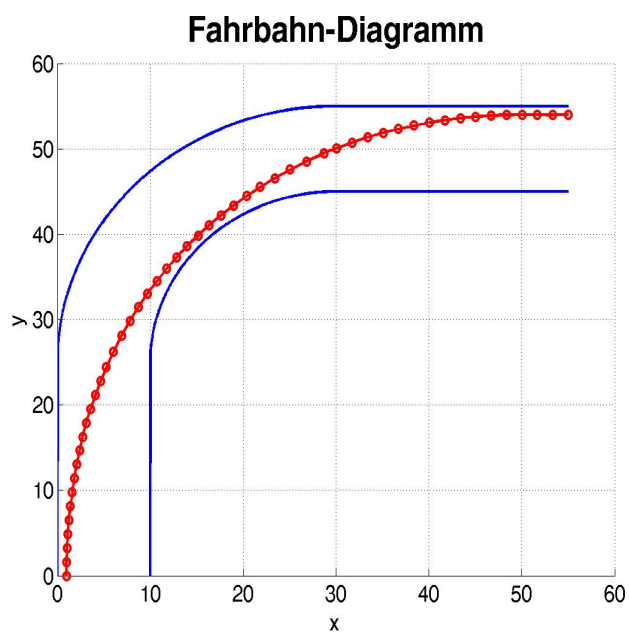


Abbildung 9: Kurvenverlauf eines VW Golf 4 und eines Formel 1 Boliden, trockener Asphalt