

Virtuelle Rundenzeitoptimierung mittels realistischer Rennfahrzeugsimulation

Torsten Butz, Martin Ehmann

TESIS *DYNAware* GmbH, Baierbrunnerstr. 15, D-81379 München,

URL: <http://www.thesis.de>, eMail: {t.butz, m.ehmann}@thesis.de

1 Einführung

Der Einsatz von maßgeschneiderten Fahrdynamik-Simulationspaketen ist Stand der Technik bei der Entwicklung von Rennfahrzeugen. Neben dem virtuellen Prototypenbau können schnelle Simulationsmodelle zu Konzeptuntersuchungen und zur Optimierung einzelner Fahrzeugkomponenten dienen. Die Entwicklung von Fahrdynamikreglern wird während des gesamten Entwicklungszyklus unterstützt, vom ersten Entwurf des Regelalgorithmus bis hin zu Funktionstests und Abstimmung des Steuergeräts in Echtzeitumgebungen, wo eine Implementierung des Reglers in Software oder Hardware mittels Sensoren und Aktuatoren mit der Vollfahrzeugsimulation verbunden ist.

Zur Simulation der Rennfahrzeugdynamik wird neben einem präzisen physikalischen Fahrzeugmodell ein Fahrermodell benötigt, das das virtuelle Fahrzeug im fahrdynamischen Grenzbereich präzise längs zeitoptimaler Führungsgrößen für die Ideallinie und das Geschwindigkeitsprofil führen kann. In diesem Beitrag wird eine Strategie zur Rundenzeitoptimierung vorgestellt, in der eine näherungsweise optimale Startlösung für die Sollgeschwindigkeit durch den Einsatz von nichtlinearen Optimierungsverfahren iterativ verbessert wird. Ergebnisse der Optimierung und die Grenzen der Vorgehensweise werden diskutiert.

2 Simulation der Rennfahrzeugdynamik

Zur realitätsgetreuen Simulation von Rennwagen wird ein umfassendes Simulationsmodell benötigt, in dem neben der Vollfahrzeugsimulation auch die Charakteristika des Rennkurses sowie Steuer- und Regeleingriffe des Fahrers mit hinreichender Genauigkeit abgebildet werden können. Dabei ist nicht nur im Hinblick auf den Echtzeitbetrieb in Hardware- und Software-in-the-Loop Simulatoren die Realisierung niedriger Rechenzeiten von erheblicher Bedeutung. Auch die Bewältigung von extrem rechenzeitintensiven Anwendungen mit einer Vielzahl von Simulationsläufen erfordert eine effiziente Implementierung der Modellgleichungen und des numerischen Löser.

Ein Quasi-Standard für die Simulation des Vollfahrzeugs in beliebigen Fahrmanövern ist das Softwarepaket *veDYNA* [15] der TESIS *DYNAware*. Neben hoher Performance erreicht die Einbettung des Programmkernelns in eine Matlab/Simulink Umgebung große Flexibilität bei der Handhabung des Fahrzeugmodells, das leicht durch zusätzliche Komponenten aus Modellbibliotheken und benutzerdefinierte Modelle erweitert werden kann. *veDYNA* ist bei einer Vielzahl von Fahrzeugherstellern und Zulieferern im Einsatz, wo es für Parameterstudien, Komfort- und Sicherheitsuntersuchungen auf dem PC ebenso verwendet wird wie für Steuergerätestests und -abstimmung sowie Dauerlaufsimulationen in Echtzeit-Umgebungen.

Fahrzeugmodell. Das Basismodell für das Fahrzeug in *veDYNA* ist ein generisches Mehrkörpersystem, das durch geeignete Parametrierung auf nahezu beliebige Fahrzeugtypen, wie Rennfahrzeuge, angepasst werden kann. Durch einen Torsionsfreiheitsgrad zwischen vorderem und hinterem Fahrzeugteil lassen sich Verwindungen des Cockpits aufgrund der hohen Beanspruchung durch äußere Kräfte abbilden. Der Einfluss der Motorlagerung kann durch weitere, frei positionierbare Körper berücksichtigt werden. Zur Modellierung der Achse stehen wahlweise nichtlineare Kinematiktabelle oder detaillierte MKS-Modelle zur Verfügung, mit denen die

Geometrie und die Elastizitäten der Achse inklusive aller enthaltenen Lenker, Lager und Hilfsrahmen behandelt werden können (Abb. 1).

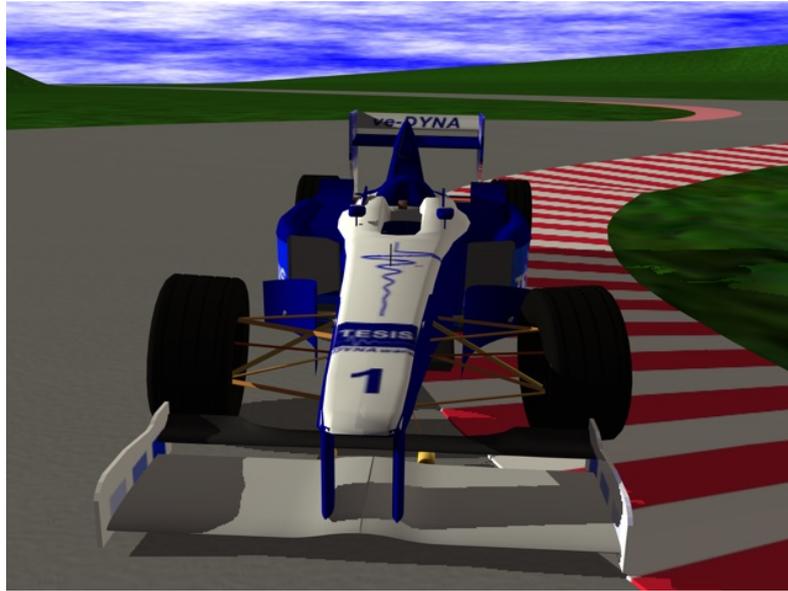


Figure 1: MKS-Modell für das Rennfahrzeug mit Lenksystem und detailliertem Achsmodell.

Spezielle fahrdynamische Eigenschaften werden mittels geeigneter Komponentenmodelle abgebildet (Abb. 2): Der Antriebsstrang setzt sich aus Modellen für den Motor, die Kupplung, das Getriebe und die vorhandenen Differentiale zusammen; darüber hinaus werden die Elastizitäten, Dämpfungskoeffizienten und Trägheiten der Antriebswellen berücksichtigt. Als Reifenmodell dient das semi-empirische Kontaktpunktmodell TM-Easy [12], in dem die Haft- und Gleitcharakteristiken der statischen Reifenkräfte ebenso enthalten sind wie dynamische Reifeneffekte. Der Austausch gegen andere Reifenmodelle, wie MF Tire, ist durch konfigurierbare Subsysteme in Simulink möglich. Ebenso können weitere benutzerdefinierte Modelle, wie z. B. für aerodynamische Effekte, komplexe Federungssysteme in der Achsaufhängung und Hydrolager zur Motorlagerung, in das Simulink-Fahrzeugmodell eingefügt werden. Sensible Modellkoeffizienten, wie Masse- und Trägheitsparameter, können zur Laufzeit verändert werden, um beispielsweise den Einfluss des Kraftstoffverbrauchs zu simulieren.

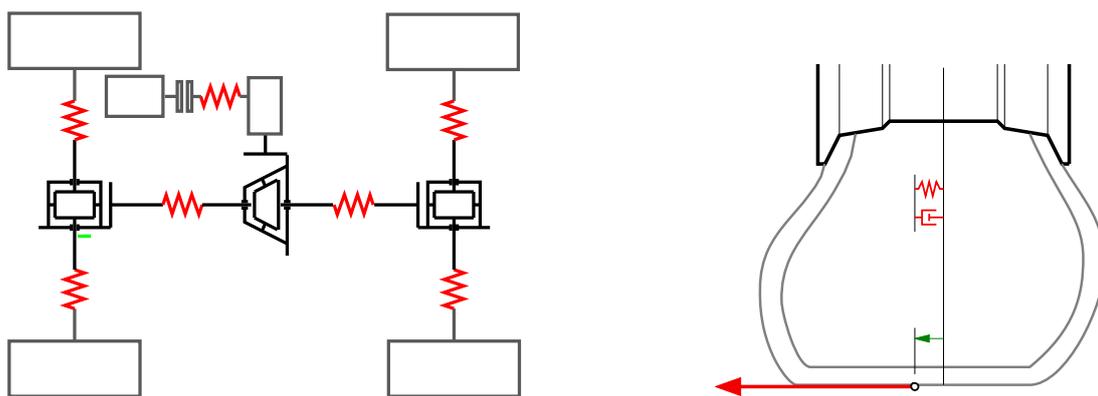


Figure 2: Komponentenmodelle für Antriebsstrang und Reifendynamik [12].

Standardverfahren zur Behandlung von Mehrkörpersystemen entwickeln die Bewegungsgleichungen in Deskriptorform und liefern ein differential-algebraisches System vom Index 3. Durch Ausnutzen der Fahrzeugtopologie und die Wahl von geeigneten generalisierten Koordinaten lassen sich die algebraischen Nebenbedingungen in

veDYNA jedoch eliminieren und die Bewegungsgleichungen auf ein System gewöhnlicher Differentialgleichungen reduzieren. Zur numerischen Integration wird ein teilimplizites Einschrittverfahren verwendet, das eine stabile Lösung auch für große Schrittweiten im Bereich von mehreren Millisekunden gestattet [12]. Damit weist veDYNA im Vergleich zu herkömmlichen MKS-Programmen um ein Vielfaches kürzere Simulationszeiten auf, ohne dass die Genauigkeit der Ergebnisse beeinträchtigt wird. Auch mit Integrationsschritten im Bereich von Zehntel Millisekunden wird Echtzeitfähigkeit auf moderner PC- und Echtzeithardware erreicht, selbst wenn zusätzlicher Zeitbedarf für benutzerdefinierte Komponenten, Regelsysteme und Kommunikation mit der angeschlossenen Hardware besteht.

Straßenmodellierung. Zur Abbildung des Rennkurses eignet sich das Straßenmodell veDYNA Road, mit dem beliebige Straßenzüge und -bedingungen mit hoher Genauigkeit dargestellt werden können [2, 18]. Der horizontale Straßenverlauf lässt sich nach dem Baukastenprinzip aus Geraden, Kreisstücken und Klothoiden zusammensetzen oder durch die Vorgabe von Raumkoordinaten des Straßenverlaufs konstruieren. Beim Höhenprofil können ebene Segmente und Teilstücke mit konstanter Steigung glatt mit stückweise kubischen Abschnitten kombiniert werden. Als Fahrbahnbelag ist die Definition von geometrischen Oberflächenprofilen, wie Spurrillen und Kerbs, ebenso möglich wie die Berücksichtigung stochastischer Rauigkeit.

Manöversteuerung Neben Open-Loop Manövern gestattet der im nächsten Abschnitt beschriebene Fahrerregler veDYNA Driver [14, 18] die Implementierung von anspruchsvollen Fahraufgaben zur Untersuchung des geschlossenen Regelkreises von Fahrzeug, Fahrer und Umgebung. Grundlage ist ein nichtlinearer Positionsregler, der in der Lage ist, das virtuelle Rennfahrzeug entlang anspruchsvoller Kurs- und Geschwindigkeitsvorgaben zu führen und vom normalen Fahrbetrieb bis hin zu kritischen Fahrsituationen zu beherrschen. Durch geeignete Variation der Reglerparameter können auf reproduzierbare und konfigurierbare Art und Weise menschliche Fahrertypen mit unterschiedlicher Fahrweise und -erfahrung abgebildet werden [8, 14].

3 Fahrermodell

Im Fahrermodell von veDYNA wird die Fahrtätigkeit des Menschen in zwei Ebenen eingeteilt: Auf der Bahnplanungsebene werden Sollgrößen für die Fahrzeugführung bestimmt. Für die Motorsportanwendungen sind dies eine zeitoptimale Ideallinie und ein passendes Geschwindigkeitsprofil. Zur Berechnung von geeigneten Sollgrößen untersuchen wir in diesem Beitrag zwei Strategien, die auf Verfahren der nichtlinearen Optimierung basieren. Auf der Stabilisierungsebene werden die vorgegebenen Führungsgrößen eingeregelt und eventuelle Störungen kompensiert. Dazu dient ein nichtlinearer Positionsregler, der sich auf das Verfahren der nichtlinearen Systementkopplung und Regelung stützt.

3.1 Bahnplanung

Ein Standardverfahren zur Bestimmung von Sollgrößen für die Simulation ist die Verwendung eines quasi-statischen Fahrzeugmodells, dessen Komplexität auf wenige Freiheitsgrade des Fahrzeugs beschränkt ist. Ein einfaches Beispiel ist ein g-g-Diagramm (z. B. [3]), das im wesentlichen ein geschwindigkeitsabhängiges Kennfeld für die erreichbaren Fahrzeugbeschleunigungen darstellt. Durch Manövrieren entlang der Beschleunigungsgrenzen des Fahrzeugs wird mittels Nachschlagen im Kennfeld ein zeitoptimales Geschwindigkeitsprofil ermittelt. Bei genaueren Modellen, wie im Programm LapSim [16], wird das Fahrzeugverhalten durch einen reduzierten Satz von Gleichungen beschrieben, in denen neben der Translation und Rotation des Fahrzeugkörpers auch Freiheitsgrade für die vier Reifen enthalten sind. Aufgrund der einfachen Modellgleichungen ist die Auswertung in Vielfachen der Echtzeit möglich.

Im Unterschied dazu untersuchen wir im folgenden die Durchführbarkeit von Optimierungsstrategien mit dynamischen Fahrzeugmodellen:

Naherungsweise zeitoptimale Sollgroen fur veDYNA konnen durch Losen eines Optimalsteuerungsproblems fur ein erweitertes Einspurmodell (vgl. Abschnitt 4) bestimmt werden. Dabei werden die nichtlinearen Reifencharakteristiken aus dem Reifenmodell TM-Easy berucksichtigt; das Zielkriterium ist die minimale Rundenzeit fur einen vorgegebenen Rennkurs. Die berechneten Kurven von Ideallinie und Geschwindigkeitsprofil dienen als Sollgroen fur die Simulation mit dem Vollfahrzeugmodell.

Zur weiteren Verbesserung der Rundenzeit verwenden wir die Ergebnisse der Optimalsteuerung als Startlosung fur ein nichtlineares Minimierungsproblem: Durch wiederholtes Abfahren eines Rennkurses bzw. Teilstucken einer Runde mit veDYNA werden die Vorgaben fur die Positionsregelung im Laufe der Iterationen sukzessive verbessert. Mittels Verfahren der nichtlinearen Optimierung werden geeignete Geschwindigkeitsprofile berechnet, die im Rahmen der Rechengenauigkeit optimal fur das Vollfahrzeugmodell sind (vgl. Abschnitt 5).

3.2 Positionsregelung

Die Modellierung der Regeltatigkeit des Fahrers basiert auf dem Verfahren der nichtlinearen Systementkopplung und Regelung (vgl. [10]). Anders als bei vielen in der Literatur verbreiteten Fahrreglern der linearen Regelungstechnik stellt dieses Modell keine Linearisierung des Fahrerhaltens um eine bestimmte Fahrsituation dar, sondern berucksichtigt die nichtlinearen Verkopplungen von Langs- und Querdynamik innerhalb der Regelstrecke Fahrzeug uber eine interne Reprasentation der Fahrdynamik. Dabei steht zunachst weniger die Nachbildung menschlichen Fahrerhaltens als das moglichst exakte Nachfahren einer vorgegebenen Bahnkurve im Vordergrund.

Die Eingabegroe des Positionsreglers ist ein zeitlich veranderlicher Sollpunkt fur die Fahrzeugposition, der im wesentlichen durch den Verlauf von Sollbahn und Sollgeschwindigkeit festgelegt wird. Zur internen Darstellung der Fahrzeugeigenschaften dient ein reduziertes Einspurmodell (vgl. Abschnitt 4.1), das durch Linearisierung von trigonometrischen Funktionen und Vernachlassigung einzelner Krafteinflusse vereinfacht ist. Die Stellgroen der Positionsregelung sind die Seitenkraft $F_{y,V}$ am Vorderrad sowie die Antriebs- bzw. Bremskraft F_x , die sich aus

$$F_{y,V} = -F_{y,H} + m \left[\left(\beta \cos(\psi + \beta) - \sin(\psi + \beta) \right) a_X^{des} + \left(\beta \sin(\psi + \beta) + \cos(\psi + \beta) \right) a_Y^{des} \right] \quad (1)$$

$$F_x = m \left[\cos(\psi + \beta) a_X^{des} + \sin(\psi + \beta) a_Y^{des} \right] + F_A \quad (2)$$

ergeben. Dabei bezeichnen ψ und β den Gier- bzw. Schwimmwinkel des Fahrzeugs, $F_{y,H}$ und F_A die Seitenkraft am Hinterrad bzw. die Gesamtwirkung der aerodynamischen Krafte, und m ist die Fahrzeugmasse. In den gewunschten Beschleunigungen a_X^{des} und a_Y^{des} sind die aktuellen Abweichungen von den Sollwerten des Positions- und Geschwindigkeitsvektors sowie Stabilitatsparameter der Regelung zusammengefasst. Zur Bedienung der Stellglieder des Vollfahrzeugmodells mussen (1), (2) unter Berucksichtigung von inversen Kennfeldern fur Reifenkrafte, Bremskreis und Motorleistung in geeignete Lenkradwinkel sowie Brems- und Gaspedalwege umgerechnet werden [18]. Das vorgegebene Geschwindigkeitsprofil wird vom Regelungsmodul, soweit moglich, wahrend der Fahrt an Fahrzeugbeschrankungen angepasst.

Durch Einbeziehung interner Fahrzeugzustande, wie dem Schwimmwinkel, wird eine prazise Bahnfolgeregelung des Fahrzeugs bis hin zum fahrdynamischen Grenzbereich erreicht, die fur professionelle Rennfahrer charakteristisch ist. Zur Abbildung der tatsachlichen Fahigkeiten von Durchschnittsfahrern konnen die Parameter des Positionsreglers verandert werden, um verschiedene fahrerische Qualitat und Erfahrung abzubilden. Durch Anpassung der Vorausschau des Fahrers, der maximalen Lenkausschlage und -geschwindigkeiten sowie der Reglerverstarkung konnen Unterschiede in der vorausschauenden Fahrweise, der physischen Fahrzeugbeherrschung und der Aggressivitat des Fahrers modelliert werden. Daruber hinaus ist eine Feinabstimmung des Fahrerhaltens mittels einer Vielzahl weiterer Parameter moglich [14].

4 Startlösung durch Bahnplanung mittels Optimalsteuerung

Näherungsweise zeitoptimale Lösungen für die Ideallinie und die Sollgeschwindigkeit des Vollfahrzeugs lassen sich mittels optimaler Bahnplanung für ein reduziertes Fahrdynamikmodell berechnen. Maßgebliche Beschränkung für die Komplexität des Modells ist dabei, dass es mit Methoden der optimalen Steuerung behandelt werden kann. Wir formulieren im folgenden eine geeignete Optimalsteuerungsaufgabe. Das dynamische System ist ein erweitertes Einspurmodell; das Optimalitätskriterium ist die benötigte Rundenzeit. Darüber hinaus werden nichtlineare Zustands- und Steuerbeschränkungen in Form der Fahrzeugabmessungen, der Kursberandungen und physischen Grenzen des Fahrers berücksichtigt. Zur numerischen Lösung des Problems wird ein direktes Kollokationsverfahren verwendet.

4.1 Optimalsteuerungsmodell

Als reduziertes Fahrdynamikmodell für die Optimalsteuerungsaufgabe verwenden wir ein Einspurmodell nach Rieckert und Schunck [11]. Dieses ist ein ebenes Fahrzeugmodell, bei dem die Vorder- und Hinterräder jeweils zu einem einzigen Rad zusammengefasst sind und die Vertikaldynamik, wie Nick- und Wankbewegungen, vernachlässigt werden (Abb. 3 links).

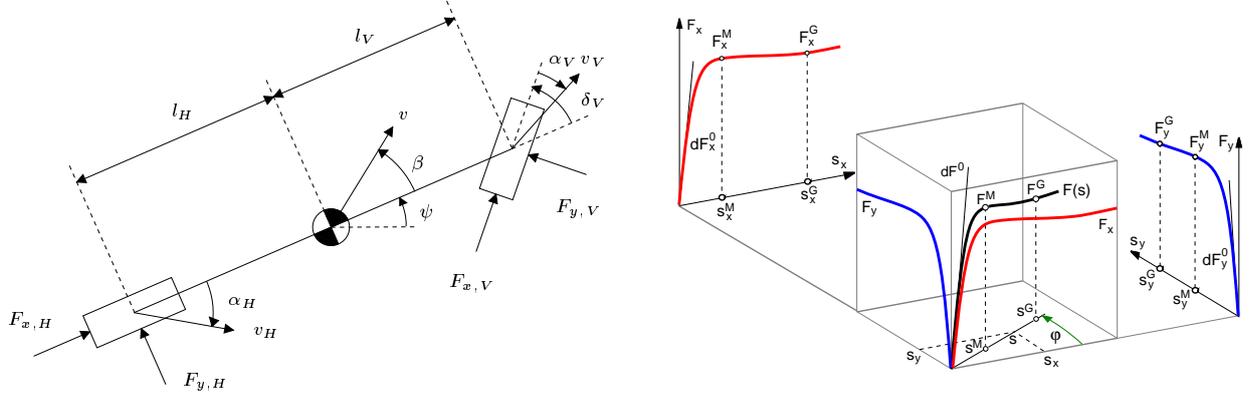


Figure 3: Einspurmodell [11] und statisches Reifenmodell [12] für die Optimalsteuerung.

Neben den Gier- und Schwimmwinkeln ψ bzw. β sind die Zustandsgrößen in diesem Modell die Gierrate $\omega_z = \dot{\psi}$, die Fahrzeuggeschwindigkeit v sowie die Fahrzeugposition X und Y in Weltkoordinaten. Im Hinblick auf physikalisch sinnvolle Lösungen der Optimalsteuerungsaufgabe wird der Lenkwinkel δ_V als eigene Zustandsvariable behandelt. Die Dynamik des Einspurmodells kann damit durch das System von gewöhnlichen Differentialgleichungen

$$\dot{\beta} = -\omega_z + \frac{1}{mv} [F_{y,H} \cos \beta - F_{x,H} \sin \beta + F_{y,V} \cos(\delta_V - \beta) + F_{x,V} \sin(\delta_V - \beta)] \quad (3)$$

$$\dot{\omega}_z = \frac{1}{I_z} [(F_{y,V} \cos \delta_V + F_{x,V} \sin \delta_V) l_V - F_{y,H} l_H] \quad (4)$$

$$\dot{v} = \frac{1}{m} [F_{y,H} \sin \beta + F_{x,H} \cos \beta - F_{y,V} \sin(\delta_V - \beta) + F_{x,V} \cos(\delta_V - \beta) - F_A] \quad (5)$$

$$\dot{\psi} = \omega_z \quad (6)$$

$$\dot{X} = v \cos(\psi + \beta) \quad (7)$$

$$\dot{Y} = v \sin(\psi + \beta) \quad (8)$$

$$\dot{\delta}_V = \omega_\delta \quad (9)$$

beschrieben werden. Als Steuervariablen verwenden wir die Lenkwinkelgeschwindigkeit ω_δ sowie die Antriebs- bzw. Bremskraft F_x , die sich je nach Antriebsart auf die Kräfte $F_{x,V}$ und $F_{x,H}$ an Vorder- und Hinterrad verteilt. Neben der Masse m treten als Fahrzeugparameter das Trägheitsmoment I_z um die Hochachse des

Fahrzeugs sowie die Abstände l_V und l_H der Räder vom Fahrzeugschwerpunkt auf; F_A bezeichnet wie zuvor den Einfluss der aerodynamischen Kräfte.

Die Seitenkräfte $F_{y,V}$ und $F_{y,H}$ vorn und hinten sind nichtlineare Funktionen der Schräglaufwinkel α_V und α_H . Zur Behandlung von im Rennbetrieb häufig auftretenden kombinierten Schlupfzuständen, wie beim Anbremsen vor und Herausbeschleunigen aus Kurven, werden die statischen Reifenkennlinien des Reifenmodells TM-Easy [12] aus veDYNA verwendet. Darin werden die effektiven Reifenkräfte durch Berechnung des Summschlupfs und elliptische Interpolation aus den Einzelcharakteristiken in Längs- und Querrichtung bestimmt (Abb. 3 rechts). Durch die kombinierte Betrachtung von Längs- und Querkräften werden die Beschränkungen der maximal erreichbaren Kräfte aufgrund der Haftgrenzen der Reifen in der Optimalsteuerung korrekt berücksichtigt.

Als Nebenbedingungen für die Rundenzeitoptimierung werden periodische Anfangs- und Endbedingungen sowie einfache Schranken an den zulässigen Wertebereich der Zustandsgrößen des dynamischen Systems (3) – (9) eingeführt. Ferner sind in der Formulierung des Optimalsteuerungsproblems zusätzliche Zustandsbeschränkungen

$$q_1(X) - Y \geq 0 \quad (10)$$

$$Y - q_2(X) \geq 0 \quad (11)$$

notwendig, um die Bewegung des Schwerpunkts innerhalb der oberen und unteren Straßenränder q_1 bzw. q_2 zu erreichen. Die Auswertung von (10), (11) erfolgt durch Abtastung des Straßenmodells veDYNA Road, das auch in der Vollfahrzeugsimulation verwendet wird. Die physischen Grenzen des Fahrers gehen in die Steuerbeschränkung

$$-\omega_{\max} \leq \omega_\delta \leq \omega_{\max} \quad (12)$$

an die Lenkwinkelgeschwindigkeit ein.

Entsprechend der primären Fahraufgabe im Rennsport, einen vorgegebenen Kurs schnellstmöglich zu bewältigen, ist das Zielkriterium für die Optimalsteuerung die Minimierung der Rundenzeit

$$\min t_f . \quad (13)$$

Der zurückzulegende Weg wird dabei durch die Randbedingungen der Optimierung vorgegeben.

4.2 Numerisches Lösungsverfahren

Zur numerischen Lösung des Optimalsteuerungsproblems verwenden wir ein Verfahren der direkte Kollokation, das im Programmpaket DIRCOL [17] implementiert ist. Dabei findet die Integration der Modellgleichungen (3) – (9) simultan mit der Minimierung des Optimalitätskriteriums (13) statt.

Bei der Kollokation wird das Zeitintervall $[0, t_f]$ gemäß

$$0 = t_0 < t_1 < \dots < t_{N-1} < t_N = t_f, \quad (14)$$

unterteilt und die dynamischen Gleichungen näherungsweise auf diesem Gitter gelöst. Die Diskretisierung wird im Laufe der Iterationen adaptiv verfeinert. Als Näherung für die Steuerungsvariablen $u = (\omega_\delta, F_x)$ definieren wir stückweise lineare Polynome

$$\hat{u}(t) = u_{j-1} + \frac{t - t_{j-1}}{t_j - t_{j-1}} (u_j - u_{j-1}), \quad t_{j-1} \leq t < t_j, \quad (15)$$

wobei die u_j die Stützwerte von \hat{u} an den Gitterpunkten bezeichnen. Die Zustandsgrößen $x = (\beta, \omega_z, v, \psi, X, Y, \delta_V)$ approximieren wir durch stückweise kubische Polynome gemäß

$$\hat{x}(t) = \sum_{k=0}^3 c_j^k \left(\frac{t - t_{j-1}}{h_j} \right)^k, \quad t_{j-1} \leq t < t_j . \quad (16)$$

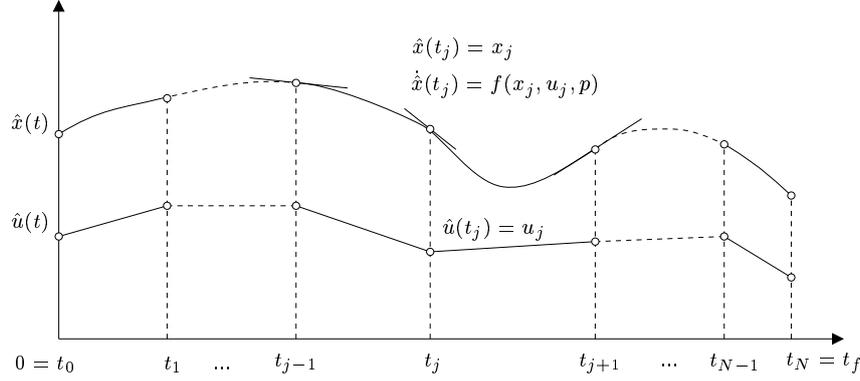


Figure 4: Diskretisierung von Steuer- und Zustandsvariablen der Optimalsteuerungsaufgabe bei der direkten Kollokation.

Dabei können die Koeffizienten c_j^k so gewählt werden, dass die Bewegungsgleichungen des Einspurmodells an den Gitterpunkten t_j automatisch erfüllt sind [17].

Die Optimalsteuerungsaufgabe reduziert sich damit auf ein nichtlineares Optimierungsproblem mit den Stützwerten u_j und c_j^0 der diskretisierten Steuerungs- und Zustandsgrößen als Unbekannte. Die Zielfunktion ist die freie Endzeit $t_f = t_N$. Zusätzliche Nebenbedingungen in Form von Gleichungs- und Ungleichungsbeschränkungen ergeben sich aus der Diskretisierung der Modellgleichungen an den Intervallmitten $t_{j-1}/2 + t_j/2$ bzw. der Fahrbahn- und Steuerbeschränkungen (10) – (12). Zur Lösung wird ein Verfahren der sequentiellen quadratischen Optimierung (SQP) benötigt (vgl. Abschnitt 5.2). Durch Verwendung der Implementierung SNOPT [5] ist es dabei möglich, die dünnbesetzte Struktur des Problems auszunutzen und sehr feine Diskretisierungen mit einer großen Zahl von Unbekannten zu behandeln.

5 Iterative Verbesserung der Führungsgrößen

Die Ergebnisse aus der Optimalsteuerung sind nur näherungsweise optimal für das Vollfahrzeug. Nach dem Vorbild von Rennfahrern versuchen wir nun, die Rundenzeiten durch wiederholtes Abfahren des Rennkurses zu verbessern. Wir formulieren dazu ein Optimierungsproblem, dessen Unbekannte durch die Diskretisierung der Sollgrößen gegeben sind. Zur Berechnung einer zeitminimalen Lösung verwenden wir Verfahren der mathematischen Optimierung.

5.1 Formulierung der Optimierungsaufgabe

Zur Bestimmung von optimalen Sollgrößen für das Vollfahrzeugmodell minimieren wir analog zur Optimalsteuerung die Zeit, die zum Absolvieren einer Runde bzw. von Teilen einer Runde benötigt wird:

$$\min t_f . \quad (17)$$

Der wesentliche Unterschied zum vorhergehenden Problem ist jedoch, dass sich die Rundenzeit aus der Simulation des Vollfahrzeugs mit veDYNA ergibt. Dabei werden dynamische Anfangs- und Endbedingungen an die Zustandsvariablen der Fahrzeugmodells vorgeschrieben, indem zu Beginn und am Ende jeweils konsistente Fahrzustände, z. B. das Überqueren der Ziellinie mit Höchstgeschwindigkeit, erreicht werden müssen.

Die Unbekannten der Optimierung sind die Stützstellen und -werte der jeweiligen Sollgrößen:

Zur Diskretisierung der Ideallinie empfiehlt sich die Verwendung von Hermiteschen Polynomen, z. B. [13]. Durch die Wahl synchroner Stützstellen für die Ortskoordinaten X und Y des Fahrzeugs wird die Ideallinie auf natürliche Weise in eine Folge von Segmenten $[s_i, s_{i+1}]$ aufgeteilt; dabei ist die unabhängige Variable s die

Bogenlänge der Sollbahn. Die Hermiteschen Basisfunktionen vom Grad 3 sind gegeben durch

$$\Phi_1(\sigma) = 2\sigma^3 - 3\sigma^2 + 1 \quad (18)$$

$$\Phi_2(\sigma) = -2\sigma^3 + 3\sigma^2 \quad (19)$$

$$\Phi_3(\sigma) = \sigma^3 - 2\sigma^2 + \sigma \quad (20)$$

$$\Phi_4(\sigma) = \sigma^3 - \sigma^2 \quad (21)$$

mit $\sigma = (s-s_i)/(s_{i+1}-s_i) \in [0,1]$. Durch Linearkombinationen mit den Ortskoordinaten (x_i, y_i) und (x_{i+1}, y_{i+1}) sowie den Tangentenvektoren $(\cos \alpha_i, \sin \alpha_i)$ und $(\cos \alpha_{i+1}, \sin \alpha_{i+1})$ am Segmentanfang bzw. -ende ergeben sich die eindeutigen Interpolationsfunktionen

$$x(s) = x_i \Phi_1(\sigma) + x_{i+1} \Phi_2(\sigma) + \cos \alpha_i \Delta s \Phi_3(\sigma) + \cos \alpha_{i+1} \Delta s \Phi_4(\sigma) \quad (22)$$

$$y(s) = y_i \Phi_1(\sigma) + y_{i+1} \Phi_2(\sigma) + \sin \alpha_i \Delta s \Phi_3(\sigma) + \sin \alpha_{i+1} \Delta s \Phi_4(\sigma) \quad (23)$$

(Abb. 5). Dieser Ansatz ist ein Kompromiss zwischen der Glattheit der Approximation, in der von Splinefunktionen bekannte Oszillationen weitgehend vermieden werden, und der verhältnismäßig geringen Anzahl von drei Unbekannten, nämlich den Ortskoordinaten x_i und y_i sowie dem Tangentenwinkel α_i , pro Stützpunkt.

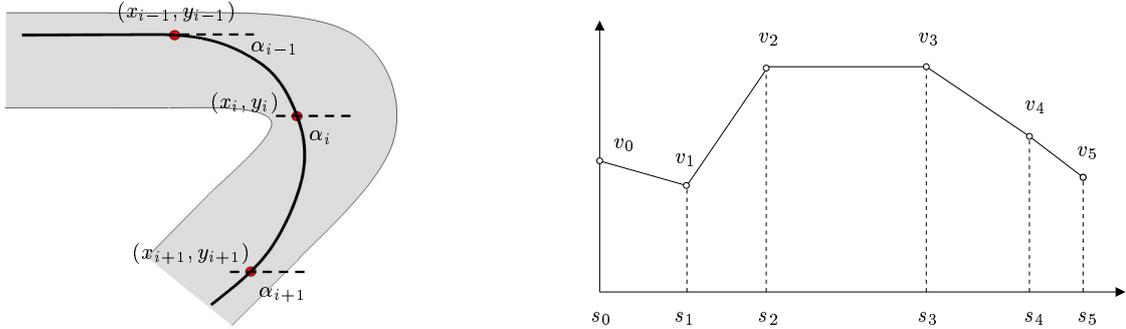


Figure 5: Diskretisierung von Ideallinie und Geschwindigkeitsprofil bei der Optimierung.

Für die Approximation des Geschwindigkeitsprofils genügt eine stückweise lineare Funktion

$$v(s) = v_i + \sigma (v_{i+1} - v_i) \quad (24)$$

mit den Geschwindigkeitswerten v_i an den Stützpunkten als freie Parameter (Abb. 5).

In beiden Fällen ergeben sich die Startwerte für die Optimierung aus der Lösung der Optimalsteuerungsaufgabe. Bei Bedarf können auch die Abszissen s_i der Sollgeschwindigkeit zur Variation freigegeben und unterschiedliche Diskretisierungen für Ideallinie und Geschwindigkeitsprofil gewählt werden.

Der Ablauf der Optimierung ist schematisch in Abb. 6 dargestellt. Zur Auswertung der Zielfunktion innerhalb der Optimierungsschleife ist jeweils die Durchführung einer Rundenzeitsimulation mit veDYNA notwendig. Sofern zur Optimierung Ableitungen der Zielfunktion (17) nach den freien Parametern benötigt werden, können diese durch numerische Differentiation mittels finiter Differenzenquotienten bestimmt werden (z. B. [13]). Dazu sind weitere Auswertungen der Zielfunktion erforderlich.

Im Hinblick auf die Lösbarkeit des Optimierungsproblems mit numerischen Verfahren ist es sinnvoll, die Komplexität und damit die Anzahl von Unbekannten möglichst gering zu halten. Dazu zerlegen wir längere Kurse in Teilstücke und behandeln die jeweiligen Kurven oder Kurvenkombinationen einzeln. Durch Wahl geeigneter Schnittpunkte und Berücksichtigung der Ergebnisse von früheren Teilstücken ist es möglich, die Teilergebnisse zu einer optimalen Lösung für den gesamten Kurs zusammenzusetzen.

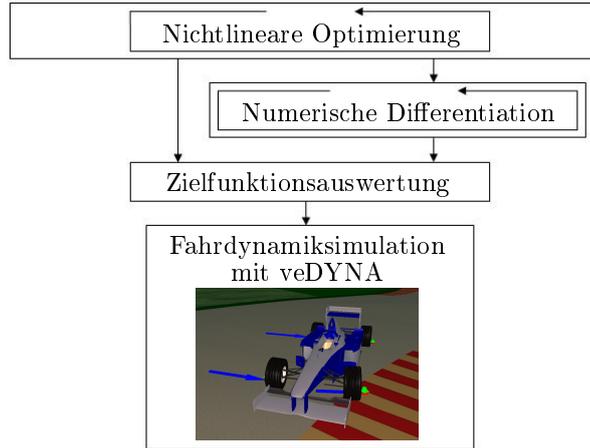


Figure 6: Schema zur Optimierung der Sollgrößen.

5.2 Optimierungsverfahren

Zur Lösung des nichtlinearen Optimierungsproblems sind eine Reihe von numerischen Verfahren denkbar. Nahelegend ist die Verwendung eines gradientenbasierten Verfahrens, beispielsweise der sequentiellen quadratischen Optimierung (SQP), da damit häufig eine effiziente Minimierung von nichtlinearen Zielfunktionen möglich ist, z. B. [5, 6]. Die wesentlichen Nachteile von gradientenbasierten Verfahren sind ihre lokalen Konvergenzeigenschaften und die hohen Anforderungen an die Differenzierbarkeit der Zielfunktion. Zum Vergleich setzen wir daher auch heuristische Suchverfahren ein, die ohne Information über den Verlauf der Ableitung auskommen. Diese sind vergleichsweise robust gegenüber Unstetigkeiten und Knicken im Modell, benötigen dafür häufig eine größere Anzahl von Iterationen bzw. Zielfunktionsberechnungen. Exemplarisch betrachten wir das Hooke-Jeeves Verfahren und das Simulated Annealing.

SQP-Verfahren. SQP-Verfahren haben zur Lösung von nichtlinearen Optimierungsproblemen der Form

$$\min_y \Psi(y) \quad (25)$$

mit Nebenbedingungen

$$h(y) = 0 \quad (26)$$

$$g(y) \geq 0 \quad (27)$$

in Form von Gleichungs- und Ungleichungsrestriktionen Verbreitung gefunden. Die grundlegende Idee besteht darin, zur Bestimmung der Abstiegsrichtung ein Hilfsproblem zu lösen, das sich durch quadratische Approximation der Zielfunktion und Linearisierung der Nebenbedingungen ergibt (z. B. [5]). Notwendige Voraussetzung ist hinreichende Glattheit der Problemfunktionen.

Die quadratische Teilaufgabe zur Bestimmung der Suchrichtung η ergibt sich durch Taylorentwicklung von (25) um die aktuelle Iterierte $y^{(l)}$

$$\min_{\eta} \frac{d\Psi}{dy}(y^{(l)}) \eta + \frac{1}{2} \eta^T H(y^{(l)}) \eta, \quad (28)$$

wobei $H(y^{(l)})$ eine geeignete Approximation der Hessematrix der Lagrange-Funktion ist [5]. Die linearisierten Nebenbedingungen haben die Gestalt

$$h(y^{(l)}) + \frac{dh}{dy}(y^{(l)}) \eta = 0 \quad (29)$$

$$g(y^{(l)}) + \frac{dg}{dy}(y^{(l)}) \eta \geq 0. \quad (30)$$

Zur Bestimmung einer geeigneten Schrittweite, die gleichermaßen die Reduzierung des Zielfunktionswerts und die Einhaltung der nichtlinearen Beschränkungen berücksichtigt, wird eine geeignete Bewertungsfunktion minimiert. Ein Beispiel dafür ist die erweiterte Lagrange-Funktion

$$\begin{aligned} \mathcal{M} = \Psi(y) & - \sum_{j=1}^{n_h} \left(\mu_j h_j(y) - \frac{1}{2} \pi_j h_j(y)^2 \right) \\ & - \sum_{j=1}^{n_g} \frac{1}{2\rho_j} (\sigma_j^2 - \min(0, \rho_j g_j(y) - \sigma_j)^2) \end{aligned} \quad (31)$$

mit geeigneten Lagrange- und Penaltyparametern.

Eine effiziente Implementierung des SQP-Verfahrens ist das Paket NPSOL [6].

Hooke-Jeeves Verfahren. Der Hooke-Jeeves Algorithmus [7] ist ein Verfahren zur Mustersuche (Pattern Search), die dem Verfahren des steilsten Abstiegs ähnelt. Der Unterschied ist, dass zur Berechnung der Abstiegsrichtung nicht die Bestimmung von Ableitungen der Zielfunktion, sondern eine Abtastung in unmittelbarer Umgebung der aktuellen Iterierten vorgenommen wird.

Die Iterationen des Hooke-Jeeves Verfahrens zerfallen in zwei Phasen: In der Tastphase wird ein diskreter Suchzyklus in jede Koordinatenrichtung, d. h. Auswertungen von $\Psi(y + \epsilon e_i)$ mit den Koordinatenvektoren e_i und einer Abtastschrittweite ϵ , durchgeführt. Sofern der Tastschritt in einer Richtung nicht zur Verbesserung des Gütevalues führt, wird zusätzlich die entgegengesetzte Richtung untersucht. Wenn längere Zeit keine Verbesserung des Zielfunktionswertes erreicht wird, wird die Tastschrittweite ϵ reduziert. In der Extrapolationsphase, wird ein größerer Schritt in die zuvor ermittelte Richtung des steilsten Abstieges ausgeführt. Die Schrittweite der Extrapolation wird verdoppelt, so lange dabei eine Verbesserung des Zielfunktionswertes möglich ist. Sofern das Verfahren die Minimalstelle verfehlt, erfolgt eine schrittweise Verringerung der Extrapolationsschrittweite. Das Verfahren bricht in der Regel ab, wenn die Tastschrittweite die Größenordnung der stochastischen Fehler der Zielfunktion erreicht hat.

Die nachfolgenden Berechnungen basieren auf der Implementierung des Hooke-Jeeves Verfahrens im Programm iSIGHT [4].

Simulated Annealing. Das Simulated Annealing [9] gehört zur Klasse der naturwissenschaftlich motivierten Verfahren. Es soll das Erstarren eines Metalls nachgebildet werden, das beim Übergang vom flüssigen in den festen Zustand Kristalle bildet, deren Atome energieminimal angeordnet sind. Weitere Methoden dieser Klasse sind genetische Algorithmen und Particle Swarm Verfahren, in denen die Mechanismen der Evolution bzw. die raumfüllende Verteilung von Vogelschwärmen und Teilchenwolken emuliert werden sollen.

Bei der numerischen Optimierung wird in jedem Schritt eine zufällig gewählte benachbarte Parameterbelegung y^* der aktuellen Iterierten untersucht. Ob diese Belegung akzeptiert wird, hängt von deren Zielfunktionswert und einem Kontrollparameter T ab, der die Wahrscheinlichkeit

$$p = \exp\left(\frac{\Psi(y^{(l)}) - \Psi(y^*)}{T}\right) \quad (32)$$

bestimmt, dass Iterierte mit einem größeren Zielfunktionswert akzeptiert werden. Der Wert von T nimmt abnehmende Werte zwischen 0 und 1 an. Während am Anfang der Iteration die neuen Konfigurationen fast wahllos weiterverwendet werden, geht mit abnehmenden Werten von T auch die Bereitschaft zurück, eine schlechtere Konfiguration zu akzeptieren. Dies entspricht sinngemäß dem Verlauf der Kristallisierung, bei der mit steigender Zähflüssigkeit auch die Sprünge zu ungünstigeren energetischen Zuständen seltener werden. Um bei der Minimierung lokale Minima zu vermeiden, erfolgt die Reduktion von T und damit die Konvergenz des Verfahrens entsprechend langsam. Setzt man den Kontrollparameter von Anfang an auf 0, entspricht das Verfahren einem Random Walk.

Zur Optimierung verwenden wir die Implementierung im Programmpaket iSIGHT [4].

6 Numerische Ergebnisse

Wir präsentieren im folgenden Ergebnisse für die iterative Verbesserung der Sollgrößen mit Verfahren der nicht-linearen Optimierung. Exemplarisch führen wir die Rundenzeitoptimierung für ein Formel 1 Fahrzeug auf dem Rennkurs in Monza durch. Im Hinblick auf die Konvergenz der numerischen Verfahren unterteilen wir den gesamten Kurs in sechs Segmente, für die wir die Minimierung einzeln durchführen (Abb. 7). Durch geschickte Segmentaufteilung können leicht konsistente und hinreichend glatte Übergangsbedingungen erreicht werden, während die Anzahl der Unbekannten, die pro Optimierungslauf behandelt werden müssen, deutlich reduziert wird.

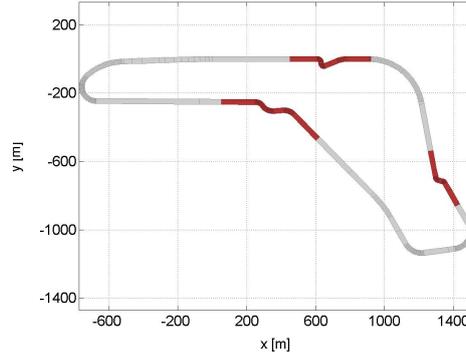


Figure 7: Segmentunterteilung für die Optimierung.

6.1 Optimierung der Ideallinie

Wir untersuchen zunächst die Optimierung der Ideallinie. Die Unbekannten sind in diesem Fall die Vektoren der Ortskoordinaten x_i und y_i sowie der Tangentenwinkeln α_i an den Stützstellen; die Diskretisierung der Bogenlänge s ergibt sich aus direkt aus den Stützwerten für die Ortskoordinaten.

Die Optimierung der Ideallinie bleibt jedoch ohne Ergebnis. Eine wesentliche Verbesserung der Rundenzeit ist nicht möglich. Der Grund dafür dürfte sein, dass die Querdynamik bereits durch das Einspurmodell und die realistischen Reifencharakteristiken sehr gut abgebildet werden kann. Diese Beobachtung deckt sich mit den Ergebnissen in [1, 19] für andere Fahrmanöver.

Auch bei Hinzunahme der Sollgeschwindigkeit liefert die Optimierung keine wesentlichen Änderungen der Ideallinie. Wir beschränken uns daher im weiteren auf die Optimierung des Geschwindigkeitsprofils. Unbekannte sind dabei jeweils die Stützstellen s_i und die Stützwerte v_i aus (24). Da für die Längsdynamik im Einspurmodell von Abschnitt 4.1 nur die statischen Reifenkräfte in Längsrichtung, nicht aber die Dynamik von Motor und Antriebsstrang berücksichtigt sind, ist hier großes Potential für Verbesserungen zu erwarten.

6.2 Optimierung des Geschwindigkeitsprofils für die Variante di Rettifilo

Zur Optimierung des Geschwindigkeitsprofils für die Variante di Rettifilo wird das 475 m lange Segment mit 18 Stützstellen diskretisiert. Neben den zugehörigen Geschwindigkeitswerten wird auch die Position der 16 inneren Knoten im Rahmen von zweckmäßigen einfachen Schranken freigegeben.

Abbildung 8 zeigt die Ideallinie für die Schikane.

In Tab. 1 sind die Ergebnisse der Optimierung mit den untersuchten Verfahren protokolliert. Mit allen drei Verfahren wird der Zielfunktionswert erheblich reduziert. Der minimale Funktionswert wird beim Simulated Annealing erreicht, bei dem im Vergleich zur Startlösung aus dem Einspurmodell die benötigte Fahrzeit um

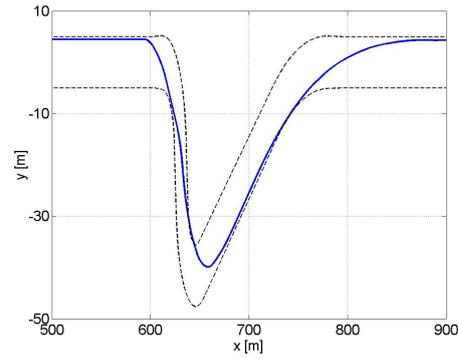


Figure 8: Ideallinie für die Variante di Rettifilo.

mehr als 10 % verringert werden kann. Der Hooke-Jeeves Algorithmus findet trotz vergleichbarer Anzahl von Funktionsauswertungen keinen Lösungspunkt mit ähnlich niedrigem Zielwert, das SQP Verfahren bricht bei einem benachbarten lokalen Minimum ab. Die Rechenzeiten auf einem Notebook mit 3 GHz Intel Prozessor und 1 GB Arbeitsspeicher bewegen sich in der Größenordnung von 1.5 Stunden.

Verfahren	Funktionsauswertungen	Rechenzeit [h]	t_f
Startlösung			13.179
SQP	506	0:43	11.920
Simulated Annealing	995	1:25	11.732
Hooke-Jeeves	1001	1:26	12.185

Table 1: Ergebnis der Optimierung für die Variante di Rettifilo.

Ein Vergleich zwischen der Startlösung sowie den optimierten Geschwindigkeitsprofilen ist aus Abb. 9 ersichtlich. Die Animation illustriert den mit der optimalen Geschwindigkeitsvorgabe herausgefahrenen Vorsprung, der sich auf mehr als 75 m beläuft.

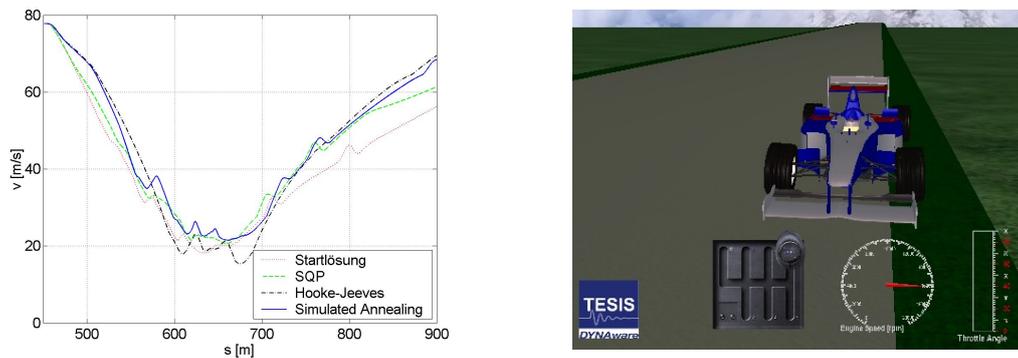


Figure 9: Vergleich zwischen Startlösung und optimierten Geschwindigkeitsprofilen für die Variante di Rettifilo (Starten der Animation durch Mausklick auf das Bild).

6.3 Optimierung des Geschwindigkeitsprofils für die Variante della Roggia

Für die Variante della Roggia definieren wir ein 550 m langes Teilstück und eine Diskretisierung mit 26 Stützstellen. Die Unbekannten der Optimierung sind die zugehörigen Stützstellen des Geschwindigkeitsprofils sowie die Abszissen der inneren Gitterpunkte.

Die Ideallinie ist in Abb. 10 dargestellt.

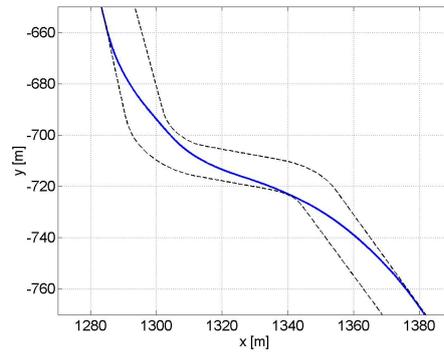


Figure 10: Ideallinie für die Variante della Roggia.

Tabelle 2 zeigt die Ergebnisse der Optimierung. Simulated Annealing und das Hooke-Jeeves Verfahren liefern zwei Lösungen mit fast identischer Fahrzeit. Die benötigten Rechenzeiten unterscheiden sich mit 2.5 bzw. 1.5 Stunden jedoch deutlich. Das SQP-Verfahren bricht schon nach verhältnismäßig wenigen Iterationen mit einer langsameren Lösung ab.

Verfahren	Funktionsauswertungen	Rechenzeit [h]	t_f
Startlösung			10.632
SQP	318	0:24	10.123
Hooke-Jeeves	1201	1:29	9.411
Simulated Annealing	1848	2:19	9.410

Table 2: Ergebnis der Optimierung für die Variante della Roggia.

In Abb. 11 vergleichen wir die optimierten Geschwindigkeitsprofile und den Anfangswert. Trotz fast identischem Zielfunktionswert ergeben sich für Simulated Annealing und Hooke-Jeeves Verfahren wesentlich unterschiedliche Lösungen. Im Gegensatz zu dem mäßigen Ergebnis bei der ersten Schikane liefert das Hook-Jeeves Verfahren hier eine Lösung, bei der die minimale Fahrgeschwindigkeit in der Schikane am höchsten liegt. Das Profil aus dem Simulated Annealing zeigt dagegen eine höhere Geschwindigkeit bei der Kurveneinfahrt.

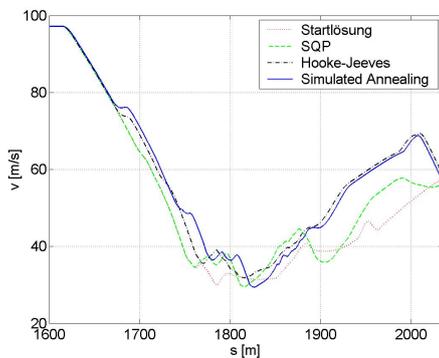


Figure 11: Vergleich zwischen Startlösung und optimierten Geschwindigkeitsprofilen für die Variante della Roggia (Starten der Animation durch Mausklick auf das Bild).

6.4 Optimierung des Geschwindigkeitsprofils für die Curva di Lesmo

Zur Bestimmung des optimalen Geschwindigkeitsprofils wird die 1600 m lange Curva di Lesmo ebenfalls in 26 Intervalle unterteilt. Dabei werden wieder die diskreten Geschwindigkeitswerte sowie die Abszissen der inneren Stützstellen zur Optimierung freigegeben.

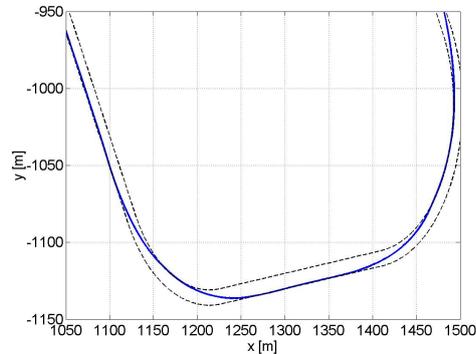


Figure 12: Ideallinie für die Curva di Lesmo.

Die Ideallinie für die eigentliche Kurve geht aus Abb. 12 hervor.

Abbildung 13 zeigt einen Vergleich der Startlösung aus dem erweiterten Einspurmodell mit den optimierten Verläufen der Sollgeschwindigkeit. Der Vorteil der Lösung mit minimalem Zeitbedarf gegenüber dem Startprofil beträgt am Ende des Segments fast 250 m.

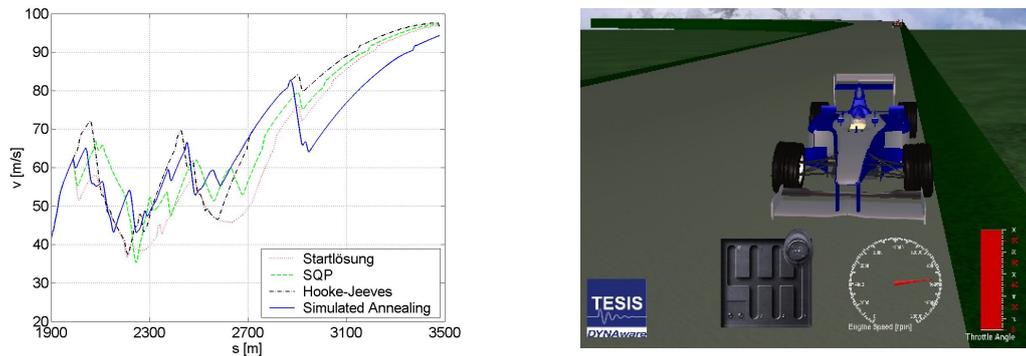


Figure 13: Vergleich zwischen Startlösung und optimierten Geschwindigkeitsprofilen für die Curva di Lesmo (Starten der Animation durch Mausklick auf das Bild).

Verfahren	Funktionsauswertungen	Rechenzeit [h]	t_f
Startlösung			26.828
SQP	482	0:50	24.527
Hooke-Jeeves	780	1:21	24.248
Simulated Annealing	1060	1:50	24.772

Table 3: Ergebnis der Optimierung für die Curva di Lesmo.

Die Ergebnisse der Optimierung sind in Tab. 3 dargestellt. Aufgrund der relativ einfachen Fahraufgabe erreichen alle drei Optimierungsverfahren eine Verbesserung der Ausgangszeit um 2 Sekunden und mehr. Den minimalen Zielfunktionswert findet das Hooke-Jeeves Verfahren; die benötigte Rechenzeit auf dem 3 GHz Notebook liegt in der Größenordnung von 1.5 Stunden. Trotz des größten Rechenzeitbedarfs wird durch Simulated Annealing nur

eine mittelmäßige Verbesserung erreicht, da das Bremsmanöver bei $s \approx 2900$ m unnötig stark ausfällt; zudem ist das Geschwindigkeitsprofil in den beiden Kurven sehr unruhig. Das gleiche gilt für das schnell abbrechende SQP-Verfahren, das aber eine etwas bessere Fahrzeit erreicht.

6.5 Optimierung des Geschwindigkeitsprofils für den gesamten Kurs

Zur Diskretisierung des Geschwindigkeitsprofils für den 5793 m langen Kurs verwenden wir eine Zerlegung in 222 Teilintervalle. Die Position des Anfangs- bzw. Endpunkts der Runde ist durch die Ausgangsposition des Fahrzeugs festgelegt; die Anfangs- bzw. Endgeschwindigkeit ist die Höchstgeschwindigkeit. Die Anzahl von Unbekannten für die Optimierung beläuft sich daher auf 442.

Die Ergebnisse der Optimierung sind aus Tab. 4 ersichtlich. Simulated Annealing erreicht keine wesentliche Verbesserung und bricht nach kurzer Zeit ab. Auch mit dem SQP-Verfahren wird die Rundenzeit nur um etwas mehr als eine Sekunde verringert; die Optimierung bleibt in einem lokalen Minimum stehen. Mit Abstand das beste Ergebnis liefert das Hooke-Jeeves Verfahren; die benötigte Rechenzeit von ca. 7.5 Stunden ist angesichts der Komplexität des Optimierungsproblems akzeptabel. Aufgrund der großen Zahl von Unbekannten erreicht keines der drei Verfahren den Minimalwert, der sich durch Aneinanderfügen der Optimierungsergebnisse für die Einzelsegmente ergibt und fast eine halbe Sekunde unter dem Ergebnis des Hooke-Jeeves Verfahrens liegt.

Verfahren	Funktionsauswertungen	Rechenzeit [h]	t_f
Startlösung			95.469
SQP	1209	4:07	94.279
Hooke-Jeeves	2235	7:35	89.440
Simulated Annealing	441	1:29	95.375
Optimierung der Teilsegmente			88.957

Table 4: Ergebnis der Optimierung für den gesamten Kurs.

6.6 Fazit

Die iterative Verbesserung der Rundenzeit mit Verfahren der nichtlinearen Optimierung ist prinzipiell machbar, wenn auch nicht völlig reibungslos. Aufgrund der hohen Nichtlinearität der Optimierungsaufgabe und der großen Anzahl von Unbekannten ist keines der untersuchten Verfahren vor einem vorzeitigem Abbruch der Iteration in einem lokalen Minimum geschützt. Insbesondere das gradientenbasierte SQP-Verfahren ist für dieses Problem anfällig: Neben der Tendenz zum Konvergieren gegen lokale Minima basiert die zur Zielfunktionsauswertung benötigte Fahrdynamiksimulation auf dem Verhalten des Positionsreglers, dessen Differenzierbarkeit zweifelhaft ist. Im Gegenzug zeigen die mittels SQP-Verfahren berechneten Geschwindigkeitsprofile stets plausibles Verhalten, während heuristische Suchverfahren zum Teil mit gänzlich unbrauchbaren Lösungen, wie z. B. einem unplausiblen Bremsmanöver in der Curva di Lesmo, abrechnen.

Durch Zerlegung des Kurses in Teilsegmente mit Längen zwischen 475 m und 2100 m lässt sich die Komplexität der Rundenzeitoptimierung verringern und die Konvergenz der numerischen Verfahren verbessern. Mit allen drei Verfahren wird eine, wenn auch teilweise nur geringe Reduktion des Zielfunktionswerts erreicht. Die insgesamt besten Ergebnisse liefert das Hooke-Jeeves Verfahren. Gemessen am Erfolg der Optimierung gehen die Rechenzeiten einen Mittelweg zwischen gradientenbasierten Algorithmen und heuristischen Suchverfahren. Im Verhältnis zur Modellgüte des Vollfahrzeugmodells benötigen die Auswertungen der Zielfunktion sehr wenig Rechenzeit. Daher ist die Verwendung von veDYNA als Simulationskernel für nichtlineare Optimierungsverfahren praktikabel.

7 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Verwendbarkeit des Fahrdynamikprogramms veDYNA zur iterativen Verbesserung von Rundenzeiten mittels Verfahren der nichtlinearen Optimierung untersucht. Die besten Ergebnisse wurden mit einer Implementierung des Hooke-Jeeves Verfahrens erzielt, das eine robuste Version des Verfahrens des steilsten Abstiegs darstellt. Damit sind nahezu zeitoptimale Simulationen mit einem komplexen MKS-Fahrdynamikmodell möglich, das detaillierte Modelle für Achsgeometrie, Reifen, Antriebsstrang und Lenkung enthält. Eine weitere Verbesserung der Ergebnisse ist denkbar, wenn die mittels SQP-Verfahren ermittelten Sollwerte als Startlösung für erneute Optimierungsläufe, etwa mit dem Hooke-Jeeves Verfahren, verwendet werden. Durch Aufteilung des Rundkurses in mehrere Teilsegmente konnten die Konvergenzeigenschaften der numerischen Algorithmen wesentlich verbessert werden. Aufgrund des verhältnismäßig niedrigen Rechenzeitbedarfs bei der Simulation mit veDYNA ist der zeitliche Aufwand für die Optimierungen auf die Größenordnung von ein paar Stunden beschränkt.

References

- [1] T. Butz: *Optimaltheoretische Modellierung und Identifizierung von Fahrereigenschaften*, Fortschritt-Berichte VDI, Reihe 8, Nr. 1080, VDI-Verlag, Düsseldorf, 2005.
- [2] T. Butz, M. Ehmann, O. von Stryk, T.-M. Wolter: *Realistische Straßenmodellierung für die Fahrdynamiksimulation in Echtzeit*, ATZ Automobiltechnische Zeitschrift, 106, 2, 2004, S. 118-125.
- [3] M. Ehmann, T. Butz: *Raceline Optimierung und Fahrermodellierung für die Simulation von Rennfahrzeugen in Echtzeit*, in: Tagungsunterlagen race.tech 2004, TÜV Süd Gruppe, Garching, 14.-15. Oktober 2004.
- [4] Engineous Software: *iSIGHT User's Guide*, Cary, NC, 2004.
- [5] P. E. Gill, W. Murray, M. A. Saunders: *SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization*, SIAM Journal on Optimization, 12, 2002, S. 979-1006.
- [6] P. E. Gill, W. Murray, M. A. Saunders, M. H. Wright: *User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming*, Report NA 98-2, Department of Mathematics, University of California, San Diego, 1998.
- [7] R. Hooke, T. A. Jeeves: *Direct Search Solution of Numerical and Statistical Problems*, Journal of the ACM, 8, 1961, S. 212-229.
- [8] M. Irmscher, M. Ehmann: *Driver Classification using veDYNA Advanced Driver*, Society of Automotive Engineers, SAE Paper 2004-01-0451, 2004.
- [9] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi: *Optimization by Simulated Annealing*, Science, 220, 1983, S. 671-680.
- [10] R. Mayr: *Verfahren zur Bahnfolgeregelung für ein automatisch geführtes Fahrzeug*, Dissertation, Fakultät für Elektrotechnik, Universität Dortmund, 1991.
- [11] P. Rieckert, T. E. Schunck: *Zur Fahrmechanik des gummibereiften Kraftfahrzeugs*, Ingenieur Archiv, 11, 1940, S. 210-224.
- [12] G. Rill: *Simulation von Kraftfahrzeugen*, Vieweg, Braunschweig, 1994.
- [13] J. Stoer: *Numerische Mathematik 1*, Springer Verlag, Berlin, Heidelberg, 8. Auflage, 1999.
- [14] TESIS DYNAware: *veDYNA Driver Reference Manual*, München, 2004.
- [15] TESIS DYNAware: *veDYNA User Manual*, München, 2004.

- [16] C. van Ruten: *Accessible Simulations*, Race Tech Magazine, 51, 2003, S. 66-72.
- [17] O. von Stryk: *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen*, Fortschritt-Berichte VDI, Reihe 8, Nr. 441, VDI-Verlag, Düsseldorf, 1995.
- [18] M. Vögel: *Fahrbahnmodellierung und Kursregelung für ein echtzeitfähiges Fahrdynamikprogramm*, Diplomarbeit, Mathematisches Institut, Technische Universität München, 1997.
- [19] M. Vögel, O. von Stryk, R. Bulirsch, T.-M. Wolter, C. Chucholowski: *An Optimal Control Approach to Real-Time Vehicle Guidance*, in: W. Jäger, H.-J. Krebs (Hrsg.): *Mathematics – Key Technology for the Future*, Springer Verlag, Berlin, Heidelberg, 2003, S. 84-102.