

Modellprädiktive Regelung zum verbrauchsoptimalen Betrieb eines PKWs mit Verbrennungskraftmaschine

Masterarbeit

Zur Erlangung des akademischen Grades eines
Diplom-Ingenieurs (Dipl.-Ing.)



Verfasser: Benjamin Winkler BSc
Begutachter: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn.tit.Univ.-Prof. Anton Hofer
Betreuer: Dipl.-Ing. Dr.techn. Josef Girstmair
Vorgelegt am: 10.06.2013

Institut für Regelungs- und Automatisierungstechnik
Technische Universität Graz
Kopernikusgasse 24/II, A-8010 Graz



Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort

Datum

Unterschrift

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place

Date

Signature

Kurzfassung

Das Ziel dieser Masterarbeit ist die Implementierung einer modellprädiktiven Regelungsstrategie bezüglich des verbrauchsoptimalen Betriebs eines PKWs mit einer Verbrennungskraftmaschine und einem hydrodynamischen Getriebe.

Die modellprädiktive Regelung basiert auf einem vom GPS-System generierten Fahrzyklus für die Fahrzeuggeschwindigkeit mit Zusatzinformation über den zu erwartenden Höhenverlauf der Strecke, wobei von der Regelung die Stellgrößen Gaspedalstellung, Bremspedalkraft und der zu wählender Gang an das Fahrzeug ausgegeben werden.

Es wird ein, für die Methode der dynamischen Programmierung als gewähltes Optimierungsverfahren geeignetes, vereinfachtes Fahrzeugmodell formuliert und anhand eines bereits vorhandenen, detaillierten Fahrzeugmodells validiert. Um die vorhandenen Modellabweichungen des vereinfachten Modells und mögliche unbekannte Störungen auszugleichen, wird ein geschlossener Regelkreis aufgebaut und die Funktionstauglichkeit an realen Straßenverläufen und daraus gewonnenen GPS-Daten gezeigt.

Um das Konzept der modellprädiktiven Regelung in Bezug auf eine potenzielle Kraftstoffersparnis zu untersuchen, wird der resultierende Kraftstoffverbrauch mit den Ergebnissen einer bereits bestehenden PI-Regelung für dasselbe Fahrzeugmodell untersucht und verglichen.

Abstract

The aim of this thesis is to implement an optimal control strategy regarding fuel consumption for a standard passenger car with a hydrodynamic torque converter.

The Model Predictive Control (MPC) is based on GPS-data from the navigation system, which generates the desired velocity trajectory of a given route. With this given driving cycle and additional information about the road's height profile, the controller calculates each timestep the optimal throttle, brake pedal force and desired gear for a fixed receding horizon.

A simplified vehicle model is formulated and evaluated with data from an already existing complex vehicle model. To compensate model deviations and possible unknown disturbances, a closed-loop Model Predictive Control is implemented and evaluated with GPS-data derived from real routes.

To investigate the MPC's potential fuel savings, the resulting fuel consumption is compared to the results of an already existing PI-control algorithm implemented for the same passenger car and route data.

Danksagung

Mein Dank gilt Herrn Prof. Anton Hofer vom Institut für Regelungs- und Automatisierungstechnik der Technischen Universität Graz für seine kompetente und verlässliche Unterstützung, sowie die konstruktiven Verbesserungsvorschläge diese Arbeit betreffend.

Des Weiteren möchte ich mich bei Herrn Josef Girstmair vom Kompetenzzentrum "Das virtuelle Fahrzeug Forschungsgesellschaft mbH" für die Ermöglichung dieser Arbeit, die engagierte Betreuung und stetige Verfügbarkeit als Ansprechpartner bedanken.

Weiterhin danke ich meinen Eltern für ihren Rückhalt und ihre Unterstützung während des Studiums und meiner Schwester für den fachkundigen Einsatz bei der Illustration dieser Arbeit.

Inhaltsverzeichnis

Abkürzungsverzeichnis	VI
1 Einleitung	1
2 Fahrzeugmodell.....	4
2.1 Beschreibung der Fahrzeugkomponenten	4
2.1.1 Die Verbrennungskraftmaschine	5
2.1.2 Der Hydrodynamischer Wandler	8
2.1.3 Die Wandlerüberbrückungskupplung.....	12
2.1.4 Das Schaltgetriebe	12
2.1.5 Der Antriebsstrang.....	13
2.1.6 Translatorische Beschreibung des Fahrzeugs.....	15
2.2 Modellerstellung für das Optimierungsproblem.....	15
2.3 Quasistatischer Lösungsansatz	18
2.4 Modellvalidierung	22
2.5 Einbindung von realen Fahrstrecken	25
3 Dynamische Programmierung.....	30
3.1 Grundlagen der dynamischen Programmierung	30
3.1.1 Einführendes Beispiel - Wegenetz	31
3.1.2 Anwendung auf dynamische Systeme.....	36
3.1.3 Abschätzung des Rechenaufwands	37
3.2 Beschreibung der MATLAB®-Funktion <i>dpm.m</i>	38
3.3 Anwendung auf das Fahrzeugmodell	43
3.3.1 Anpassung des Fahrzeugmodells.....	43
3.3.2 Einstellungen für die MATLAB®-Funktion <i>dpm.m</i>	48
3.3.3 Simulationsergebnisse	50

4	Modellprädiktive Regelung.....	55
4.1	Grundlagen der modellprädiktiven Regelung.....	55
4.2	Anwendung der modellprädiktiven Regelung auf das Fahrzeugmodell	60
4.3	Evaluierung der Modell Prädiktiven Regelung	63
5	Vergleich modellprädiktive Regelung - PI-Regelung.....	70
6	Zusammenfassung und Ausblick	76
	Tabellenverzeichnis.....	77
	Abbildungsverzeichnis.....	78
	Anhang	80
	A. MATLAB® MPC – S-Funktion.....	80
	Literaturverzeichnis.....	85

Abkürzungsverzeichnis

Griechische und mathematische Symbole

α_{GP}	Gas-/Fahrpedalstellung in %
ω	Winkelgeschwindigkeit in rad/s
ν	Drehzahlverhältnis hydrodynamischer Wandler
$\lambda(\cdot)$	Proportionalitätsfaktor hydrodynamischer Wandler
$\mu(\cdot)$	Momentenverhältnis hydrodynamischer Wandler
μ_R	Reibbeiwert
ρ	Dichte in kg/m ³
η	Wirkungsgrad
Θ	Massenträgheitsmoment in kgm ²
γ	Steigungswinkel in rad
M	Drehmoment in Nm
V	Verbrauch in l
D	Durchmesser in m
i_G	Getriebeübersetzungsverhältnis
g_G	Gang
s_R	Radschlupf
r_{Rad}	Radradius in m
F	Kraft in N
k	Konstante
v	Fahrzeuggeschwindigkeit in m/s
A	Fläche in m ²
m	Fahrzeugmasse in kg
t	Zeit in s

u	Stellgröße
x	Zustandsgröße
J	Kostenfunktion
h	Kostenwert
T	Zeitkonstante in s
N	Problemlänge des Optimierungsproblems
s_f	Schaltfreigabe
S	Wegschritt in m
g	Erdbeschleunigung in m/s^2

Abkürzungen, Indizes

GP	Gas-/Fahrpedal
BP	Bremspedal
VM	Verbrennungskraftmaschine
G	Gang
P	Pumpe
T	Turbine
k	Zeitpunkt
D	Diskretisierungszeit
H	Prädiktionshorizont
MPC	Model Predictive Control
PI	Proportional-Integral
DW	Drehmomentwandler
FTP	Federal Test Procedure
$EUDC$	Extra-Urban Drive Cycle
ATS	Antriebsstrang
PKW	Personenkraftwagen

1 Einleitung

Bedingt durch die kontinuierlich steigenden Kraftstoffpreise, strengere gesetzliche Rahmenbedingungen und ein immer stärker werdendes Verlangen nach CO₂-Emissionsreduktion in der Gesellschaft, werden in der Fahrzeugindustrie große Anstrengungen unternommen, um die Effizienz von Verbrennungskraftfahrzeugen zu steigern.

Eine Möglichkeit zur Steigerung der Effizienz liegt darin, vorhandene, jedoch bisher nicht berücksichtigte Informationen über die zu fahrende Strecke z.B. bei der Wahl des zu verwendenden Gangs zu berücksichtigen.

Ziel dieser Arbeit ist es nun, einen geeigneten Regelalgorithmus zu implementieren, der es ermöglicht, einem gegebenen Geschwindigkeitsprofil $v(t)$ mit zusätzlicher Information über das Höhenprofil $\gamma(t)$ unter minimalem Kraftstoffverbrauch V_{VM} zu folgen. Der Regler soll dabei die Gaspedalstellung α_{GP} , Bremskraft F_{BP} und den richtigen Gang g_G wählen, und damit die Tätigkeit des Fahrers in Bezug auf die Beschleunigung und Verzögerung des Fahrzeugs vollständig ersetzen. Der prinzipielle Aufbau des angestrebten Regelkreises ist in Abbildung 1.1 dargestellt.

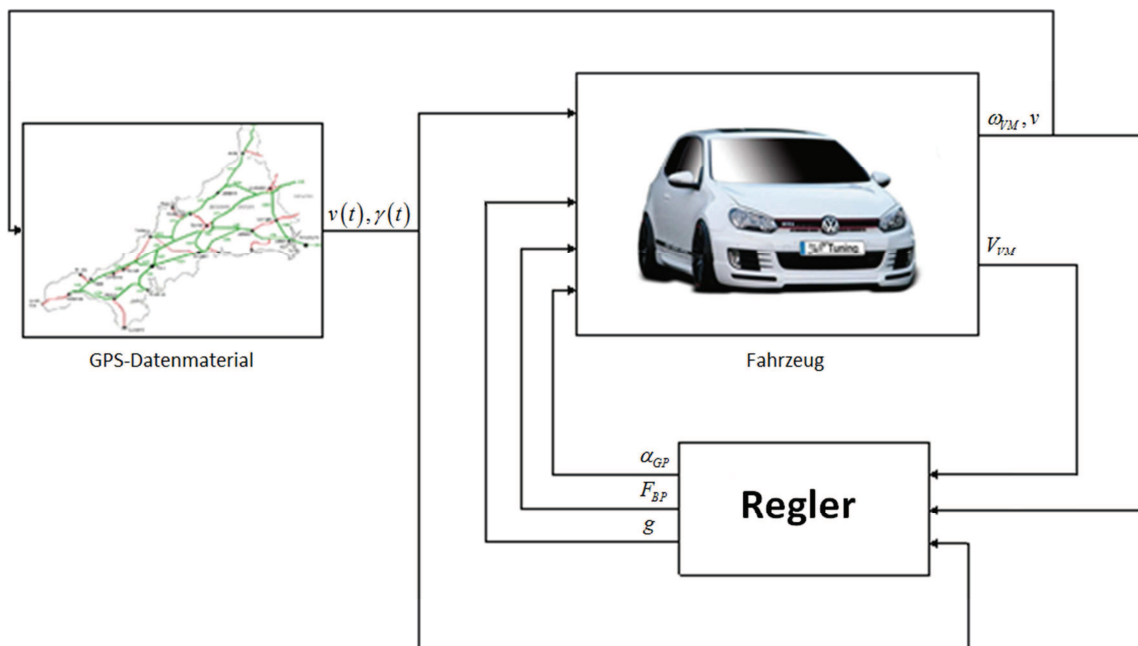


Abbildung 1.1: Prinzipieller Aufbau der geforderten Regelung

Es soll dabei eine möglichst einfache und robuste Implementierung gewählt werden, um den erstellten Algorithmus ohne großen Zusatzaufwand auf Fahrzeuge des verwendeten Typs anwenden zu können. Um diese Aufgabe bewältigen zu können, sind mehrere Schritte notwendig. Zunächst muss ein geeignetes Fahrzeugmodell, welches die Längsdynamik des Fahrzeugs ausreichend genau beschreibt, gefunden werden. Dieses Fahrzeugmodell, erläutert in Kapitel 2.3, dient als Basis für das Optimierungsproblem, welches mit Hilfe der Methode der dynamischen Programmierung implementiert wird. Die Methode der dynamischen Programmierung stellt eine sehr effiziente Optimierungsmethode für Probleme mit beschränkten Zustands- und Eingangsgrößen dar und wird in Kapitel 3 näher erläutert. Diese Methode liefert jedoch lediglich eine optimale Steuerfolge für die 3 Eingangsgrößen des Fahrzeugs und ist daher sehr anfällig für etwaige Störungen und Abweichungen im Modell. Aus diesem Grund ist es erforderlich, einen geschlossenen Regelkreis nach dem Konzept der modellprädiktiven Regelung (Kapitel 4) zu implementieren. Der dabei eingesetzte Regler verwendet zu jedem Abtastzeitpunkt die dynamische Programmierung zur Optimierung über einen endlichen Prädiktionshorizont. Durch diese Regelstruktur können externe Störungen und Modellabweichungen für die bekannte Strecke berücksichtigt werden und somit das Fahrzeug über den gesamten

Fahrverlauf bei minimalem Kraftstoffverbrauch betrieben werden. In Kapitel 4.3 werden verschiedene Parameter der modellprädiktiven Regelung und ihr Einfluss auf die Kraftstoffersparnis und die Berechnungsdauer für eine mögliche Echtzeit-Implementierung untersucht. Um die Funktion der Regelung zu überprüfen sollen mehrere Streckenszenarien mit realen GPS-Daten abgefahren werden.

Abschließend wird in Kapitel 5 die implementierte modellprädiktive Regelung mit einer bereits bestehenden PI-Regelung des Fahrzeugs mit automatischer Wahl des Gangs verglichen und insbesondere auf potentielle Kraftstoffersparnis untersucht.

In Kapitel 6 werden die Erkenntnisse der Arbeit zusammengefasst und ein Ausblick über die mögliche zukünftige Erweiterung der Arbeit und Anwendungsmöglichkeiten gegeben.

2 Fahrzeugmodell

In dem nun folgenden Kapitel werden die Hauptkomponenten eines modernen PKWs mit Verbrennungskraftmaschine und Automatikgetriebe beschrieben und näher erläutert (Abschnitt 2.1.1 - 2.1.6). Des Weiteren wird die für die Optimierung nötige Modellerstellung in Abschnitt 2.2 für ein möglichst einfaches dynamisches Modell des Fahrzeugs beschrieben und der Ansatz der quasistatischen Modellbildung im folgenden Abschnitt 2.3 vorgestellt. Zum Abschluss des Kapitels erfolgt in Abschnitt 2.4 eine Modellvalidierung des vereinfachten Modells, wobei als Referenz das bereits vorhandene detaillierte Fahrzeugmodell in MATLAB®/Simulink dient.

2.1 Beschreibung der Fahrzeugkomponenten

Um ein für das Optimierungsproblem geeignetes Modell erstellen zu können, muss zunächst einmal das detaillierte Fahrzeugmodell analysiert werden. In diesem Abschnitt werden die Hauptkomponenten des Fahrzeugs beschrieben und näher erläutert.

2.1 Beschreibung der Fahrzeugkomponenten

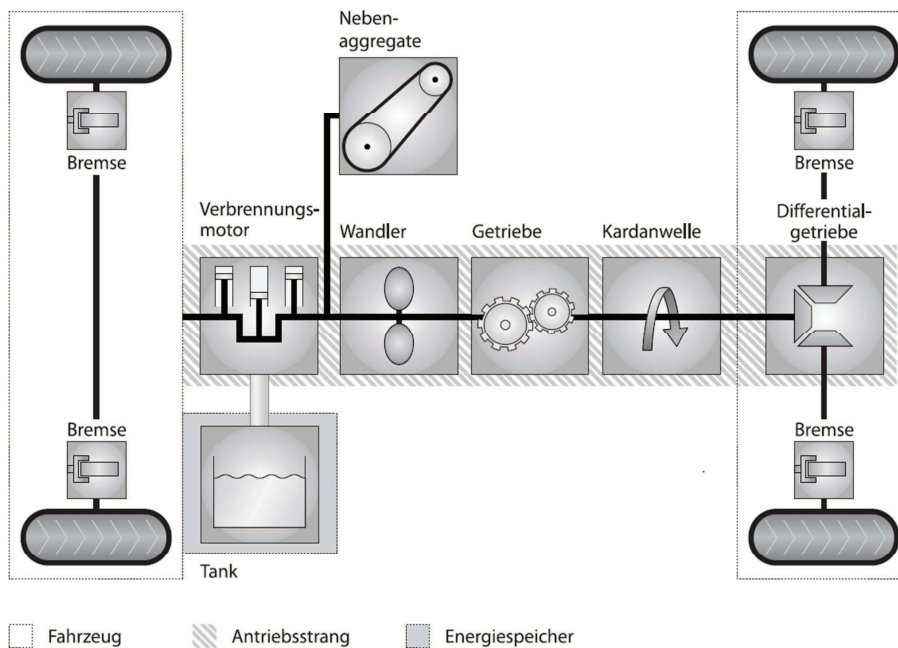


Abbildung 2.1: Aufbau eines Verbrennungskraftfahrzeugs¹

Das in Abbildung 2.1 dargestellte Fahrzeugmodell beschreibt im Wesentlichen die Hauptkomponenten des Fahrzeugs, wobei im Folgenden näher auf die relevanten Komponenten eingegangen wird.

2.1.1 Die Verbrennungskraftmaschine

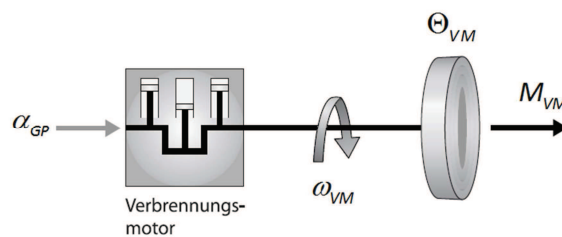


Abbildung 2.2: Schematische Darstellung des Motorblocks²

Die Beschreibung der Verbrennungskraftmaschine erfolgt in der Form eines Mittelwertmodells, welches auf Messdaten eines realen Verbrennungsmotors basiert. Im

¹ Überarbeitet aus [15]

² Überarbeitet aus [15]

für die Simulation des Fahrzeugs nötigen Simulink-Modell wurden die gesammelten Messdaten in einem Kennfeld für den gesamten Betriebsbereich zusammengefasst und wie in Abbildung 2.3 gezeigt eingebunden.

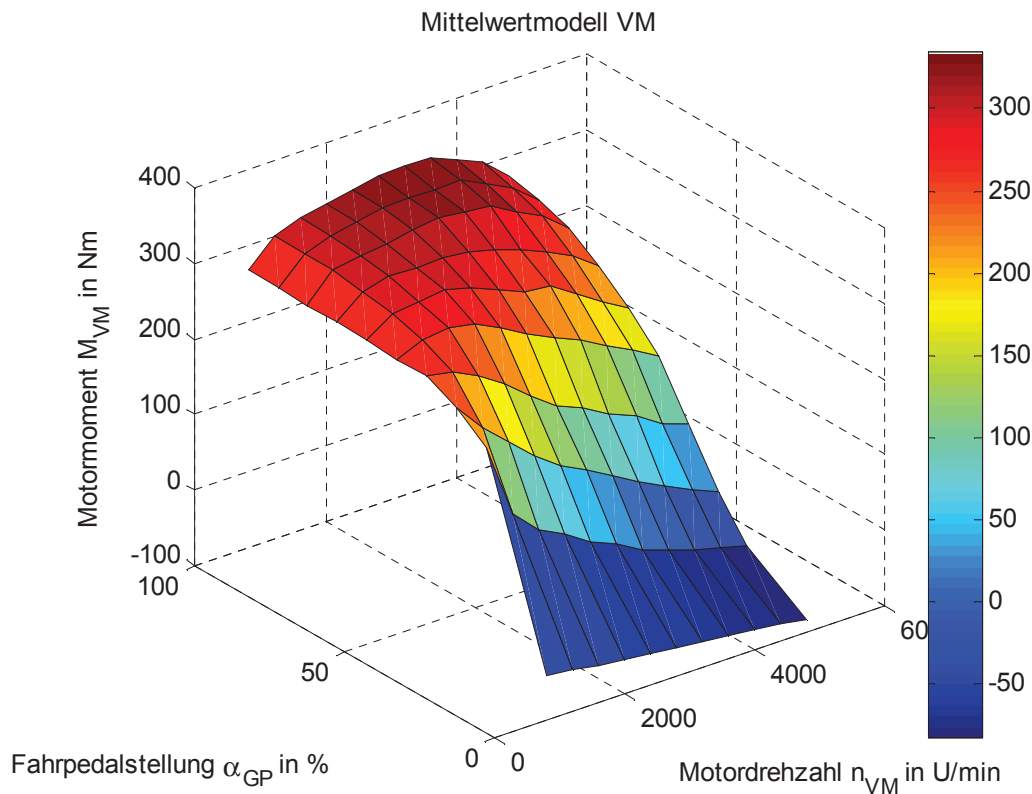


Abbildung 2.3: Kennfeld für das Motormoment der verwendeten VKM³

Das in Abbildung 2.3 dargestellte Kennfeld bildet abhängig von der Fahrpedalstellung α_{GP} gemessen in % und der aktuellen Motordrehzahl in U/min das vom Motor abgegebene Drehmoment in Nm. Für die Fahrpedalstellung gilt dabei folgende harte Stellgrößenbeschränkung:

$$0\% \leq \alpha_{GP} \leq 100\% \quad (2.1)$$

Der Betriebsbereich von $800 \frac{U}{min} \leq n_{VM} \leq 4800 \frac{U}{min}$ sollte nicht verletzt werden.

Für die Modellbildung kann das Motormoment mit der Beziehung

³ 4-Zylinder Reihendieselmotor VW 2.0l TDI

$$M_{VM} = f(\alpha_{GP}, \omega_{VM}) = M_{VM}(\alpha_{GP}, \omega_{VM}) \quad (2.2)$$

beschrieben werden. Um die Verbrennungskraftmaschine vollständig beschreiben zu können, wird noch das mittlere Trägheitsmoment Θ_{VM} des Kurbeltriebs benötigt. Dieses kann über die kinematischen Zusammenhänge und Massen berechnet werden und als gegeben angenommen werden.

Mit dem berechneten Drehmoment M_{VM} und der Winkelgeschwindigkeit ω_{VM} kann ebenfalls mit Hilfe eines auf Messdaten beruhenden Kennfelds der aktuelle Verbrauch V_{VM} der VM bestimmt werden. Dabei gilt die Beziehung:

$$V_{VM} = f(\omega_{VM}, M_{VM}) = V_{VM}(\omega_{VM}, M_{VM}) \quad (2.3)$$

Diese wird durch die Kennlinien⁴ in Abbildung 2.4 beschrieben. Die schwarze Linie stellt die Vollastlinie des Motors dar, d.h. $\alpha_{GP} = 100\%$ für den gesamten Drehzahlbereich und entspricht der Betriebsgrenze des Motors. Der Bereich oberhalb der Vollastlinie kann im Betrieb nicht erreicht werden.

⁴ Messdaten wurden für den Einsatz in Simulink extrapoliert

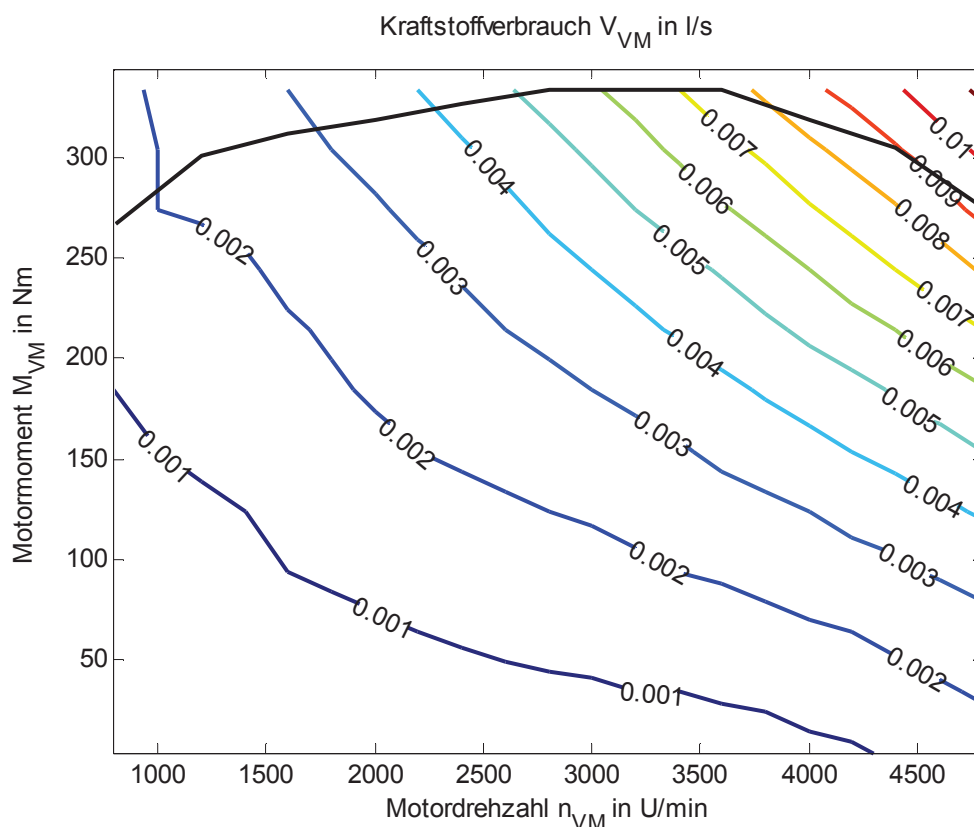


Abbildung 2.4: Kraftstoffverbrauch für gesamten Betriebsbereich

2.1.2 Der Hydrodynamischer Wandler

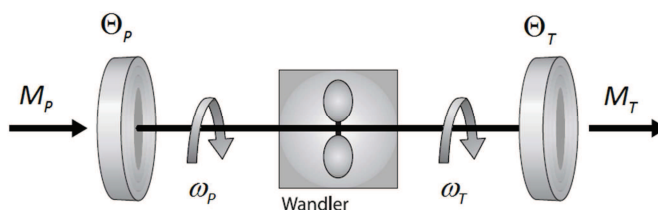


Abbildung 2.5: Schematische Darstellung des hydrodynamischen Wandlers⁵

Ein hydrodynamischer Drehmomentwandler (Abbildung 2.5), näher beschrieben in [1] besteht aus einem Pumpenrad, Turbinenrad und Leitrad (Abbildung 2.6) und dient zur Drehzahlwandlung für den Anfahr- und Schaltvorgang. Als Arbeitsflüssigkeit dient ATF-Öl⁶, welches nacheinander Pumpe, Turbine und das feststehende Leitrad durchströmt. Der

⁵ Überarbeitet aus [15]

⁶ Automatic Transmission Fluid

2.1 Beschreibung der Fahrzeugkomponenten

hydrodynamische Wandler bildet gemeinsam mit dem Schaltgetriebe ein so genanntes hydrodynamisches Getriebe, welches in modernen Automatikgetrieben meist Verwendung finden. Einer der Hauptvorteile liegt in der elastischen Verbindung zwischen Motor und Antriebsstrang, welche zu einer Komforterhöhung beim Schalt- und Anfahrvorgang für den Fahrer führt. Es findet dabei stets eine Anpassung des Übersetzungsverhältnisses an die Belastung durch den Antriebsstrang statt. Da der hydrodynamische Wandler jedoch in weiten Betriebsbereichen einen schlechten Wirkungsgrad aufweist, ist der Einsatz eines nachgeschalteten Schaltgetriebes nötig. Dieses Schaltgetriebe muss lastschaltbar sein und erhöht dadurch den Bauaufwand des hydrodynamischen Schaltgetriebes. Das Schaltgetriebe kann zur weiteren Steigerung der Effizienz eine Wandlerüberbrückungskupplung erweitert werden, welche im Abschnitt 2.1.3 beschrieben wird.

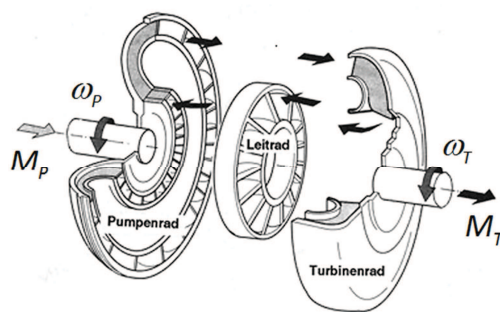


Abbildung 2.6: Hauptkomponenten eines hydrodynamischen Wandlers⁷

Um den abfallenden Zweig der Wirkungsgradparabel zu umgehen, werden in PKWs häufig so genannte Trilok-Wandler⁸ eingesetzt. Diese verbinden die Vorteile von hydrodynamischen Wandlern und hydrodynamischen Kupplungen. Es handelt sich dabei um einen Zweiphasen-Wandler. In der ersten Phase wirkt der hydrodynamische Wandler mit feststehendem Leitrad und in der zweiten Phase löst sich das Leitrad über einen Freilauf, wodurch der Trilok-Wandler als hydrodynamische Kupplung zu wirken beginnt.

⁷ Überarbeitet aus

http://www.lehrerfreund.de/xinha/plugins/ImageManager/demo_images/tec/drehmomentwandler_440.png März 2013

⁸ Benannt nach der TRILOK-Forschungsgesellschaft [1]

Das Verhalten des hydrodynamischen Wandlers wird durch das Ähnlichkeitsgesetz [1] für das Pumpenmoment M_p und das Turbinenmoment M_t wie folgt beschrieben:

$$M_p = \lambda(\nu) \cdot D^5 \cdot \rho \cdot \omega_p^2 \quad (2.4)$$

$$M_t = \mu(\nu) \cdot \lambda(\nu) \cdot D^5 \cdot \rho \cdot \omega_p^2 = \mu(\nu) \cdot M_p \quad (2.5)$$

wobei stets gilt

$$\nu = \frac{\omega_t}{\omega_p}. \quad (2.6)$$

Das Momentenverhältnis μ und der Proportionalitätsfaktor λ ⁹ sind dabei Funktionen des Drehzahlverhältnisses ν von Turbinendrehzahl ω_t und Pumpendrehzahl ω_p nach Gl. (2.6). Die beiden Funktionen sind dabei je nach Wandlertyp verschieden und wurden vom Hersteller zur Verfügung gestellt. Mit dem Durchmesser D (in m) wird der strömungstechnisch relevante Durchmesser des Wandlers beschrieben. Die Dichte ρ des verwendeten ATF-Öls beträgt in der Regel $\rho = 800 - 900 \text{ kg / m}^3$ [1].

⁹ auch als Leistungsziffer bezeichnet [1]

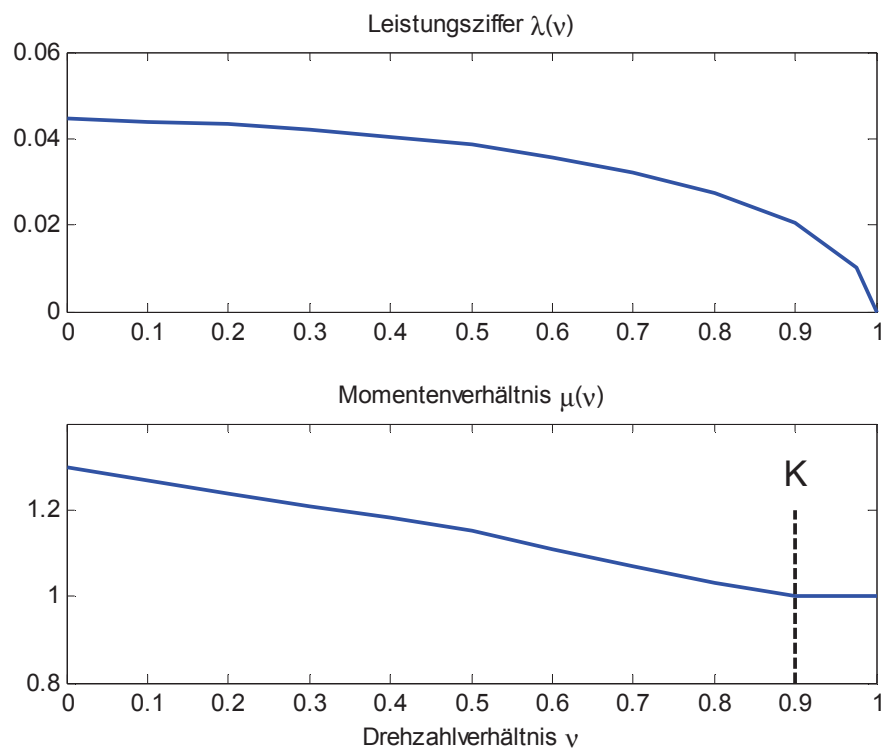


Abbildung 2.7: Kennlinien des hydrodynamischen Wandlers

In Abbildung 2.7 ist mit dem Buchstaben K der Kupplungspunkt dargestellt. Ab diesem Punkt gilt $M_p = M_T$ und der Trilok-Wandler wirkt als hydrodynamische Kupplung.

Im Schubbetrieb muss darauf geachtet werden, dass das Turbinenrad zum antreibenden Rad wird und der Trilok-Wandler somit ein anderes Verhalten aufweist, welches im detaillierten Simulink-Modell gesondert behandelt wird.

2.1.3 Die Wandlerüberbrückungskupplung

In vielen modernen Automatikgetrieben wird zur weiteren Effizienzsteigerung eine Wandlerüberbrückungskupplung parallel zum verwendeten hydrodynamischen Wandler verbaut. Diese Kupplung kann zum einen im schlupfenden bzw. offenen Betrieb und zum anderen im geschlossenen Betrieb verwendet werden. Dabei ist der geschlossene Betrieb nach Schaltvorgängen sinnvoll, um die hydrodynamischen Verluste des Wandlers in Betriebszuständen mit geringem Wirkungsgrad zu umgehen. Der weitere Freiheitsgrad einer geschlossenen oder offenen Wandlerüberbrückungskupplung wird im Zuge dieser Arbeit nicht berücksichtigt und auf eine Implementierung im mathematischen Modell daher verzichtet.

2.1.4 Das Schaltgetriebe

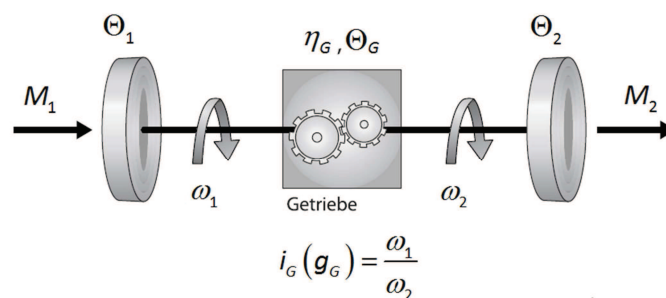


Abbildung 2.8: Schematische Darstellung des Schaltgetriebes¹⁰

Bei dem im detaillierten Fahrzeugmodell vorhandenen Getriebe handelt es sich um ein 4-Gang Getriebe. Dieses Modell beinhaltet den vom Eingangsmoment M_1 und Eingangsdrehzahl ω_1 abhängigen Wirkungsgrad η_G der einzelnen Gänge g_G , welcher messtechnisch erfasst wurde.

Das Getriebe wie in Abbildung 2.8 kann mit folgenden Gleichungen für die Ausgangsdrehzahl ω_2 und das Ausgangsmoment M_2 beschrieben werden:

¹⁰ Überarbeitet aus [15]

$$\omega_2 = \frac{\omega_1}{i_G(g_G)} \quad (2.7)$$

$$M_2 = \eta_G(g_G, M_1, \omega_1) \cdot i_G(g_G) \cdot M_1 \quad (2.8)$$

$$g_G = \{1, 2, 3, 4\} \quad (2.9)$$

2.1.5 Der Antriebsstrang

Im restlichen Antriebsstrang ab dem Getriebe sind alle vorhandenen Wellen, Übersetzungen, ein schlupfbasiertes Reifenmodell und die mechanischen Bremsen modelliert. Im Wesentlichen handelt es sich bei der Verbindung zwischen Getriebe und Antriebsräder (ohne schlupfbasiertes Reifenmodell) um eine Drehschwingerkette [2], welche als lineares Differentialgleichungssystem 6. Ordnung abgebildet werden kann. Mit Hilfe dieser Drehschwingerkette kann die aktuelle Raddrehzahl ω_{Rad} bestimmt werden. Aus der Raddrehzahl, dem mittleren Radradius r_{Rad} und der aktuellen Fahrzeuggeschwindigkeit v kann der Schlupf s_R nach Formel (2.10) bestimmt werden. Über das für den verwendeten Reifen typische Umfangskraftdiagramm (Abbildung 2.9) zur Ermittlung des Kraftschlusses μ_R und der Radlast F_n kann die, dem Schlupf entsprechende, auf die Straße übertragene Antriebskraft F_{Rad} nach Formel (2.11) berechnet werden.

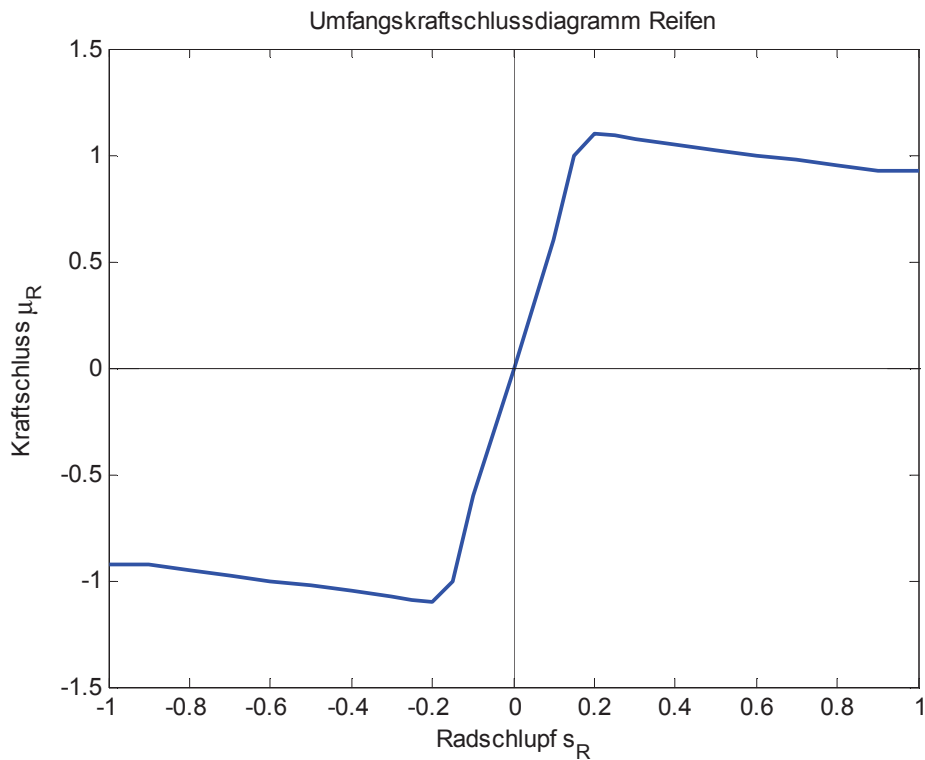


Abbildung 2.9: Umfangskraftdiagramm des verwendeten Reifens

$$s_R = \frac{\omega_{Rad} \cdot r_{Rad} - v}{v} \quad (2.10)$$

$$F_{Rad} = \mu_R(s_R) \cdot F_n \quad (2.11)$$

Das durch die mechanische Bremse erzeugte Bremsmoment M_{BR} kann über die auf das Bremspedal wirkende Kraft F_{BP} berechnet und durch folgende Gleichung beschrieben werden:

$$M_{BR} = 2 \cdot \mu_{BR} \cdot r_{Rad} \cdot F_{BR} \quad (2.12)$$

$$F_{BR} = k_{BP} \cdot F_{BP} \quad (2.13)$$

Der Reibkoeffizient μ_{BR} wird dabei für den gesamten Betriebsbereich als konstant angenommen. Der konstante Faktor k_{BP} bildet jene Übersetzung zwischen der Kraft auf das Bremspedal und der Normalkraft auf die Bremsbacken der Bremsen.

2.1.6 Translatorische Beschreibung des Fahrzeugs

Die Beschreibung der Längsdynamik des Fahrzeugs erfolgt in Form der Bewegungsgleichung (2.18). Diese Bewegungsgleichung enthält alle auftretenden Verluste durch den Luftwiderstand (2.14), den Rollwiderstand (2.15) und der Kraft durch die Steigung γ im Straßenverlauf (2.16).

$$F_{Luft} = \frac{1}{2} \cdot \rho \cdot A \cdot c_w \cdot v^2 = k_{Luft} \cdot v^2 \quad (2.14)$$

$$F_{Roll} = m \cdot g \cdot k_{Roll1} + m \cdot g \cdot k_{Roll2} \cdot v \quad (2.15)$$

$$F_{Steigung} = m \cdot g \cdot \sin(\gamma) \quad (2.16)$$

Die Konstanten k_{Luft} , k_{Roll1} und k_{Roll2} stellen fahrzeugspezifische Parameter dar, g die Erdbeschleunigung und m die Gesamtfahrzeugmasse.

In Summe ergibt sich demnach die Kraft

$$F_{Verlust} = F_{Luft} + F_{Roll} + F_{Steigung}, \quad (2.17)$$

und damit die folgende Differenzialgleichung:

$$\dot{v} \cdot m = F_{Rad} - F_{Luft} - F_{Roll} - F_{Steigung}. \quad (2.18)$$

2.2 Modellerstellung für das Optimierungsproblem

Um das bei der modellprädiktiven Regelung verwendete Optimierungsproblem möglichst schnell (Echtzeitfähigkeit) und einfach lösen zu können, muss das Fahrzeugmodell stark vereinfacht werden. Dabei gilt in der Regel, dass bei der dynamischen Programmierung als Optimierungsmethode ein nichtlineares Modell 1. oder 2. Ordnung angestrebt werden sollte. Modelle höherer Ordnung führen meist zu einer zu langen Rechenzeit, um das Problem, in dem bei der modellprädiktiven Regelung begrenzten Zeitabschnitt, lösen zu können. Wie in Kapitel 3.1.3 beschrieben steigt der Rechenaufwand exponentiell mit der Ordnung des Systems.

Auf Grund des im Fahrzeug vorhandenen Trilok-Wandlers (Kapitel 2.1.2) ist die Kopplung zwischen Motorachse und Antriebsstrang stark nichtlinear und kann lediglich mit dem Verhältnis der beiden Drehzahlen beschrieben werden. Aus diesem Grund liegt es nahe, ein dynamisches System 2. Ordnung für das Gesamtfahrzeug zu wählen. Es müssen dabei die folgenden Vereinfachungen getroffen werden:

- Vernachlässigung des Reifenschlupfs
- starrer Antriebsstrang
- reduziertes Wandlermodell
- lineare Verluste im Getriebe

Betrachtet man den Motorblock mit dem Pumpenrad des hydrodynamischen Wandlers und setzt $\omega_{VM} = \omega_p$, ergibt sich folgende Differenzialgleichung zur Beschreibung der Motordrehzahl:

$$\dot{\omega}_{VM} = \frac{1}{\Theta_{VM} + \Theta_p} [M_{VM}(\alpha_{GP}, \omega_{VM}) - M_p(\omega_T, \omega_{VM})] \quad (2.19)$$

Das Turbinenmoment M_T des hydrodynamischen Wandlers ist durch Gl. (2.5) beschrieben und bildet das Eingangsmoment M_1 aus Gl. (2.8) des Schaltgetriebes. Unter der Vernachlässigung der Drehzahl- und Momentenabhängigkeit des Getriebewirkungsgrades kann für das Ausgangsmoment M_2 folgendes angenommen werden:

$$M_2 = \eta_G \cdot i_G(g_G) \cdot M_T \quad (2.20)$$

Die 2-stufige Drehschwingerkette des Antriebsstrangs kann unter Vernachlässigung des dynamischen Schwingungsverhaltens als starre Verbindung bestehend aus 2 Stufen mit fixem Übersetzungsverhältnis angenommen werden. Dabei müssen die Massenträgheitsmomente der einzelnen Stufen auf die Ein- oder Ausgangsseite zusammengefasst werden [2]:

$$\Theta_{Rad} = \frac{\Theta_1}{i_1^2 \cdot i_2^2} + \frac{\Theta_2}{i_2^2} + \Theta_{Reifen} \quad (2.21)$$

2.2 Modellerstellung für das Optimierungsproblem

Das Übersetzungsverhältnis i_1 entspricht der Sekundärübersetzung des Getriebes mit einem Massenträgheitsmoment Θ_1 und i_2 der Achsgetriebeübersetzung und Θ_2 dem Massenträgheitsmoment der 2.Stufe. Die Gesamtübersetzung des Antriebsstrangs ist somit konstant und kann mit $i_{ATS} = i_1 \cdot i_2$ angenommen werden. Zusammengefasst mit der vom Gang abhängigen Getriebeübersetzung ergibt sich daher für die Gesamtübersetzung in Abhängigkeit vom Gang g_G :

$$i(g_G) = i_G(g_G) \cdot i_1 \cdot i_2 \quad (2.22)$$

Bei Vernachlässigung des Schlupfs ergibt sich für die Raddrehzahl ω_{Rad} in Abhängigkeit der Fahrzeuggeschwindigkeit v zu:

$$\begin{aligned} \omega_{Rad} &= \frac{v}{r_{Rad}} \\ \dot{\omega}_{Rad} &= \frac{\dot{v}}{r_{Rad}} \end{aligned} \quad (2.23)$$

Die Bewegungsgleichung für das Rad lautet:

$$\dot{\omega}_{Rad} \cdot \Theta_{Rad} = i_1 \cdot i_2 \cdot M_2 - M_{BR} - M_{Rad} \quad (2.24)$$

Nach Einsetzen von Gl.(2.23), Gl.(2.22) und Gl.(2.20) in Gl.(2.24) ergibt sich:

$$\frac{\dot{v} \cdot \Theta_{Rad}}{r_{Rad}} = \eta_G \cdot i(g_G) \cdot M_T - M_{BR} - M_{Rad} \quad (2.25)$$

Durch die Vernachlässigung des Schlupfs gilt weiter:

$$M_{Rad} = r_{Rad} \cdot F_{Rad} \quad (2.26)$$

und Gl. (2.11) verliert in diesem Zusammenhang ihre Gültigkeit. Durch Einsetzen von Gl. (2.25) und (2.26) in Gl.(2.18) ergibt sich die Bewegungsgleichung des Gesamtfahrzeugs wie folgt:

$$\dot{v} = \frac{r_{Rad}}{\Theta_{Rad} + m \cdot r_{Rad}^2} \left[\eta_G \cdot i(g_G) \cdot M_T(\omega_{VM}, g_G, v) - r_{Rad} \cdot F_{Verlust}(v, \gamma) - M_{BR} \right] \quad (2.27)$$

Zusammengefasst ergibt sich nun aus Gl. (2.19) und (2.27) ein Differentialgleichungssystem 2. Ordnung der Form:

$$\begin{bmatrix} \dot{\omega}_{VM} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Theta_{VM} + \Theta_P} [M_{VM}(\alpha_{GP}, \omega_{VM}) - M_P(\omega_{VM}, g_G, v)] \\ \frac{r_{Rad}}{\Theta_{Rad} + m \cdot r_{Rad}^2} [\eta_G \cdot i(g_G) \cdot M_T(\omega_{VM}, g_G, v) - r_{Rad} \cdot F_{Verlust}(v, \gamma) - M_{BR}] \end{bmatrix} \quad (2.28)$$

2.3 Quasistatischer Lösungsansatz

Um nun den Lösungsaufwand zur Bestimmung der optimalen Lösung des Differentialgleichungssystems nach Gl. (2.28) weiter zu reduzieren, bietet sich der quasistatische Lösungsansatz, vorgestellt von Guzzella und Sciarretta [3], an.

Normalerweise werden dynamische Systeme mit gewöhnlichen Differentialgleichungen wie in Gl. (2.19) und (2.27) beschrieben und durch Integration gelöst. Im Falle der translatorischen Bewegung einer bewegten Masse m unter Einfluss einer Kraft $F(t)$ gilt im Allgemeinen:

$$\frac{dv}{dt} = \frac{1}{m} \cdot F(t) \quad (2.29)$$

Dabei wird die auf die Masse m wirkende Kraft als Ursache für eine Bewegungsänderung in Form einer Beschleunigung betrachtet und mittels Integration die Geschwindigkeit der Masse m zu jedem Zeitpunkt bestimmt.

Das Lösen von Differentialgleichungssystemen höherer Ordnung ist jedoch mit einem erheblichen Rechenaufwand verbunden und sollte bei zeitkritischen Anwendungen, so weit wie möglich, umgangen werden. Bei bestimmten Anwendungen ist nun die angestrebte Lösung einzelner Zustandsgrößen a-priori bekannt und kann als Information für den quasistatischen Lösungsansatz verwendet werden.

Liegen nun zum Beispiel Informationen über den angestrebten Verlauf der Geschwindigkeit v der Masse m zu jedem Zeitpunkt vor, kann daraus mittels Gl.(2.30)

die Beschleunigung näherungsweise im Zeitintervall $T_D = t_2 - t_1$, also quasistationär bestimmt werden.

$$\frac{dv}{dt} \approx \frac{v(t_2) - v(t_1)}{t_2 - t_1} \quad (2.30)$$

Unter der Voraussetzung, dass der Zeitschritt T_D klein genug gewählt wurde, kann die gewöhnliche Differentialgleichung (2.29) quasistationär gelöst werden.

$$F = m \cdot \frac{dv}{dt} \approx m \cdot \frac{v(t_2) - v(t_1)}{t_2 - t_1} \quad (2.31)$$

Beim quasistatischen Ansatz wird demnach die Wirkungsrichtung umgekehrt, aus der Durchschnittsgeschwindigkeit im Zeitschritt T_D folgt eine konstante Beschleunigung und letztendlich die Kraft als Wirkung nach Gl. (2.31).

Im Falle der Anwendung auf das dynamische System des Fahrzeugmodells (2.28) können die im Vorfeld bekannten Informationen über den Fahrzyklus $v(t), \dot{v}(t)$ und den Höhenverlauf $\gamma(t)$ der Strecke zur weiteren Vereinfachung und somit effizienteren Lösung des Differentialgleichungssystem herangezogen werden. Dadurch reduziert sich die Ordnung des Differentialgleichungssystem auf ein System 1. Ordnung mit der Motorwinkelgeschwindigkeit ω_{VM} als dynamische Größe. Ein wichtiger Aspekt für den Einsatz des quasistatischen Ansatzes stellt die „Fahrbarkeit“ des Fahrzyklus mit dem jeweiligen Fahrzeug dar, d.h. es muss vor dem Einsatz des quasistatischen Ansatzes sichergestellt sein, dass der angestrebte Fahrzyklus mit dem Fahrzeug gefahren werden kann. Im mathematischen Sinne muss der beim quasistatischen Ansatz verwendete zeitliche Verlauf der dynamischen Größen auch eine mögliche Lösung des Differentialgleichungssystem darstellen.

In Abbildung 2.10 ist die Vereinfachung durch den quasistatischen Ansatz dargestellt und zeigt sowohl den dynamischen (volle Linien), als auch den quasistatischen (halbvolle Linien) Teil des Modells.

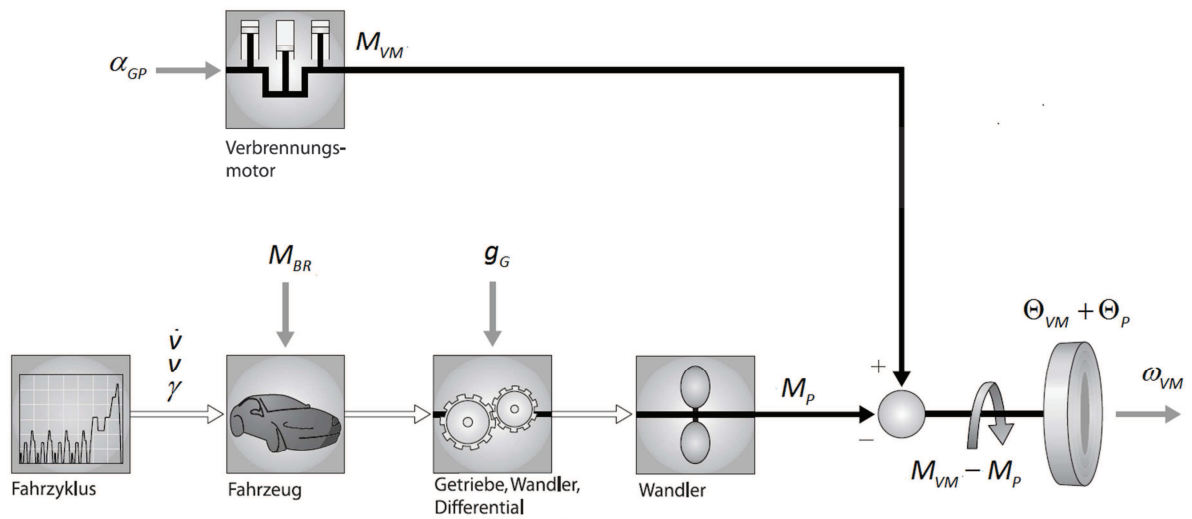


Abbildung 2.10: Anwendung des quasistatischen Ansatzes auf das Fahrzeugmodell¹¹

Für die Implementierung des Modells in einer geeigneten Form zur Verwendung des quasistatischen Lösungsansatzes muss das nichtlineare Differentialgleichungssystem nach Gl. (2.28) zeitdiskretisiert werden. Das einfachste Verfahren zur Lösung nichtlinearer Differentialgleichungssysteme mit gegebenen Anfangswerten stellt das explizite Euler-Verfahren [4] dar. Es handelt sich dabei um ein Einschrittverfahren erster Ordnung, d.h. es ist nur eine Funktionsauswertung des Modells nötig, um die Lösung für den nächsten Zeitschritt zu bestimmen. Die Nachteile liegen in der geringen Fehlerordnung und dem begrenzten Konvergenzgebiet. Die Wahl der Diskretisierungszeit ist dabei entscheidend für die Qualität und Stabilität der Lösung mit dem Euler-Verfahren. Im Falle von linearzeitinvarianten (LZI) Systemen können die Stabilitätsgrenzen des Euler-Verfahrens sehr einfach, über die Eigenwerte der Systemmatrix, bestimmt werden, im vorliegenden Falle eines nichtlinearen Systems müssen die Grenzen für die Lösbarkeit mittels Simulationsstudien überprüft werden.

¹¹ Überarbeitet aus [15]

2.3 Quasistatischer Lösungsansatz

Auf das dynamische System des Gesamtfahrzeugmodells aus Gl.(2.28) angewendet ergibt sich das folgende System von Differenzgleichungen:

$$\begin{aligned} \omega_{VM,k+1} &= \omega_{VM,k} + T_D \cdot \frac{1}{\Theta_{VM} + \Theta_P} \left[M_{VM}(\alpha_{GP,k}, \omega_{VM,k}) - M_P(\omega_{VM,k}, g_{G,k}, v_k) \right] \\ v_{k+1} &= v_k + T_D \cdot \frac{r_{Rad}}{\Theta_{Rad} + m \cdot r_{Rad}^2} \left[\eta_G \cdot i(g_{G,k}) \cdot M_T(\omega_{VM,k}, g_{G,k}, v_k) - r_{Rad} \cdot F_{Verlust}(v_k, \gamma_k) - M_{BR,k} \right] \end{aligned} \quad (2.32)$$

Nach dem Einsetzen der im Kapitel 2.3 getroffenen Vereinfachungen ergibt für das Pumpenmoment $M_{P,k}$ des Drehmomentwandlers

$$M_{P,k} = \frac{1}{\eta_G \cdot i(g_{G,k}) \cdot \mu_k} \left[\dot{v}_k \frac{\Theta_{Rad} + m \cdot r_{Rad}^2}{r_{Rad}} + r_{Rad} \cdot F_{Verlust,k} + M_{BR,k} \right] \quad (2.33)$$

und durch Einsetzen von Gl.(2.33) in Gl.(2.32) die folgende Differenzgleichung für die Motordrehzahl:

$$\omega_{VM,k+1} = \omega_{VM,k} + \frac{T_D}{\Theta_{VM} + \Theta_P} \cdot \left[M_{VM,k} - \frac{1}{\eta_G \cdot i(g_{G,k}) \cdot \mu_k} \left[\dot{v}_k \frac{\Theta_{Rad} + m \cdot r_{Rad}^2}{r_{Rad}} + r_{Rad} \cdot F_{Verlust,k} + M_{BR,k} \right] \right] \quad (2.34)$$

Die Beziehung (2.34) stellt dabei die Differenzgleichung für die Systembeschreibung mit Hilfe der quasistatischen Methode nach Abbildung 2.10 dar. Im weiteren Verlauf dieser Arbeit wird diese Systembeschreibung für die Optimierung mit Hilfe der in Kapitel 3 vorgestellten Methode verwendet.

2.4 Modellvalidierung

Im Anschluss an die Modellierung und Vereinfachung durch den quasistatischen Ansatz ist es sinnvoll, das erstellte Fahrzeugmodell anhand von Simulationen zu verifizieren. Durch den Vergleich des Verhaltens der Modelle (vereinfachtes und detailliertes Modell) bei gleicher Wahl der Gaspedalstellung α_{GP} , der Bremspedalkraft F_{BP} und des gewählten Gangs g_G über die Zeit, kann die Güte des vereinfachten Modells bestimmt werden. Bei der Wahl der Eingangsgrößen sollte versucht werden, einen möglichst großen Bereich des Zustandsraums des Fahrzeugs abzudecken. Es werden dabei die folgenden 3 Modelle verglichen:

- grün** Fahrzeugmodell 1. Ordnung mit quasistatischem Ansatz (Kapitel 2.3)
 $T_D = 0.5s$ nach Gl. (2.34)
- blau** Fahrzeugmodell 2.Ordnung, $T_D = 0.1s$ nach Gl. (2.32)
- rot** detailliertes Simulink-Fahrzeugmodell als Referenz

Die zur Validierung verwendeten Fahrzyklen EUDC¹² und FTP-75¹³ [5] wurden im Bereich der vorgesehen Stillstände des Fahrzeugs modifiziert, da diese Fahrsituationen in weiterer Folge nicht vorgesehen sind.

In Abbildung 2.11 und Abbildung 2.12 sind jeweils die Verläufe der Motordrehzahl ω_{VM} und der Fahrzeuggeschwindigkeit v dargestellt. Durch den Vergleich der 3 Verläufe ist klar zu erkennen, dass sich alle 3 Modelle ähnlich verhalten und vor allem bei der Fahrzeuggeschwindigkeit die Abweichungen marginal sind.

Bei Betrachtung der Verläufe der Motordrehzahl ω_{VM} ergibt sich lediglich bei statischen Betriebszuständen, d.h. bei gleichbleibender Geschwindigkeit, eine Abweichung. Diese lässt sich durch, die im vereinfachten Modell nicht modellierten, höheren Verluste im Getriebe bei geringem Eingangs- bzw. Motormoment erklären.

¹² Extra-Urban Drive Cycle

¹³ Federal Test Procedure

2.4 Modellvalidierung

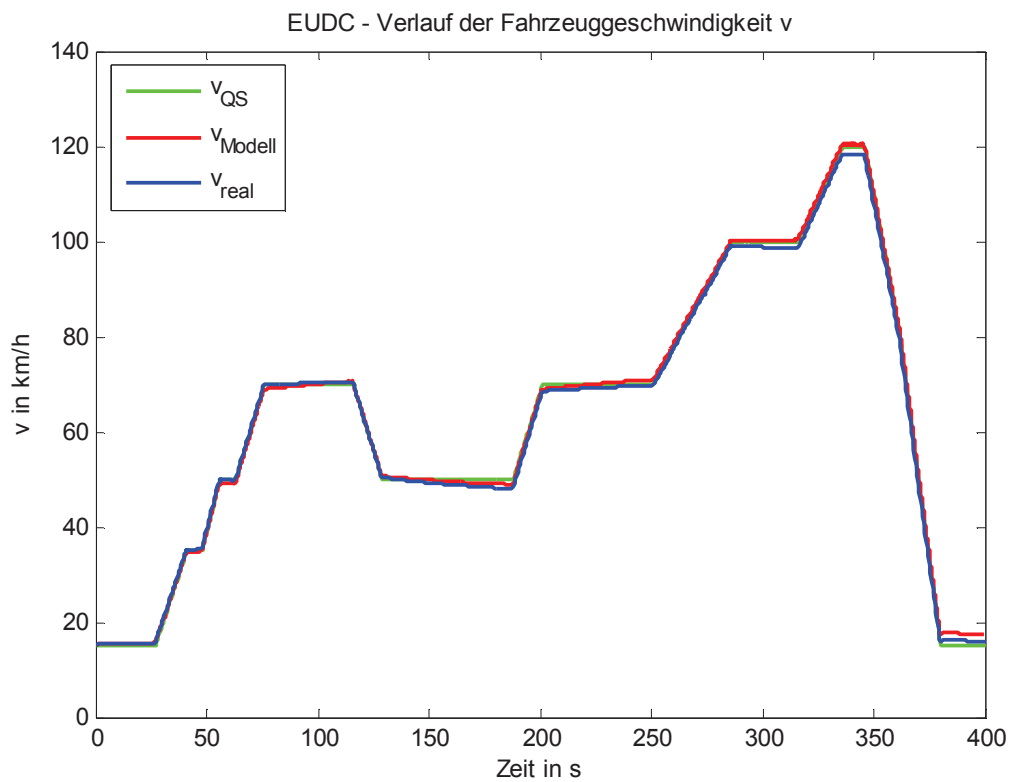
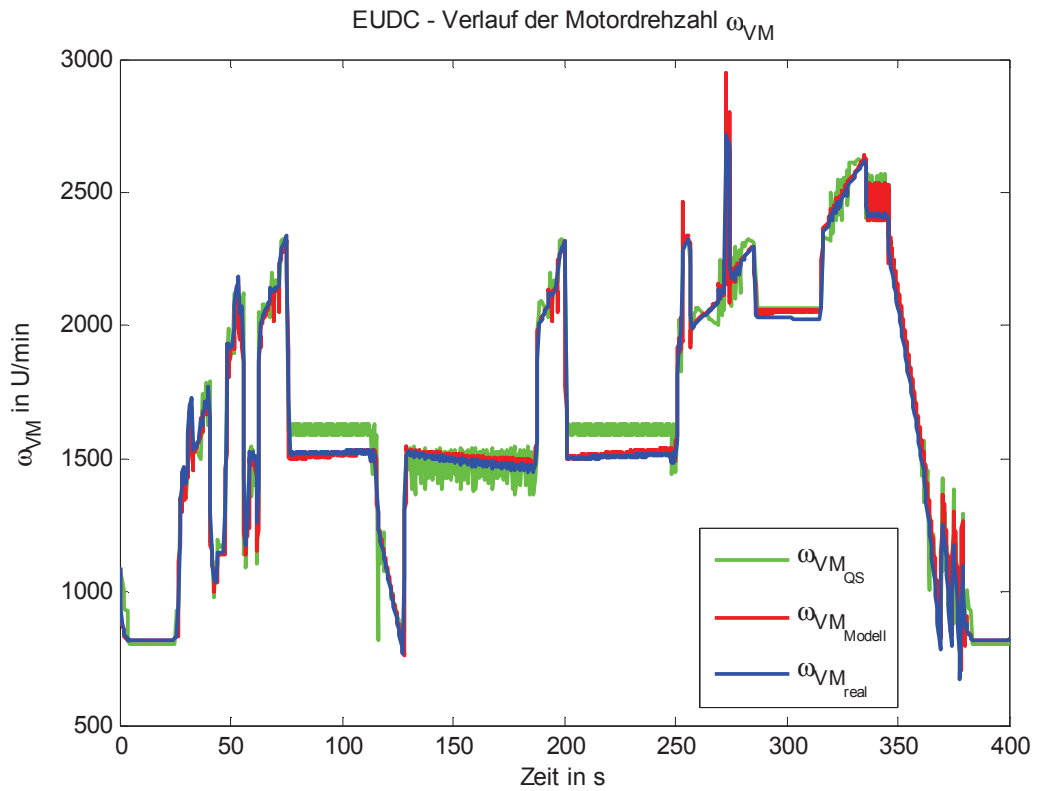


Abbildung 2.11: Modellvalidierung anhand des EUDC-Fahrzyklus

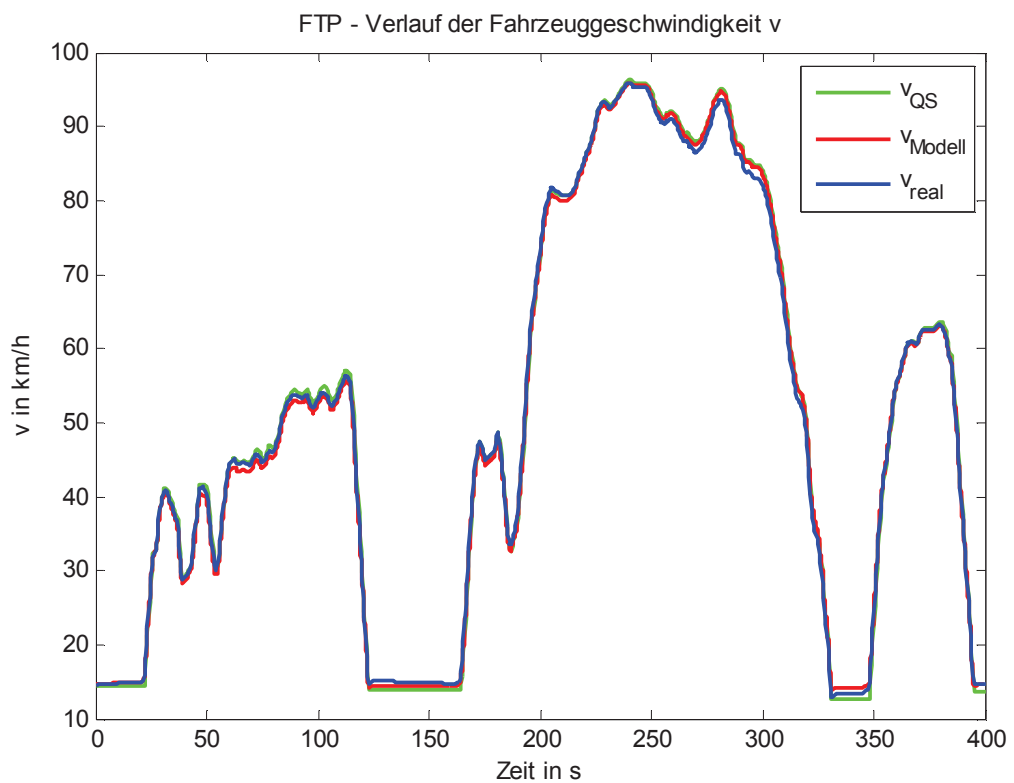
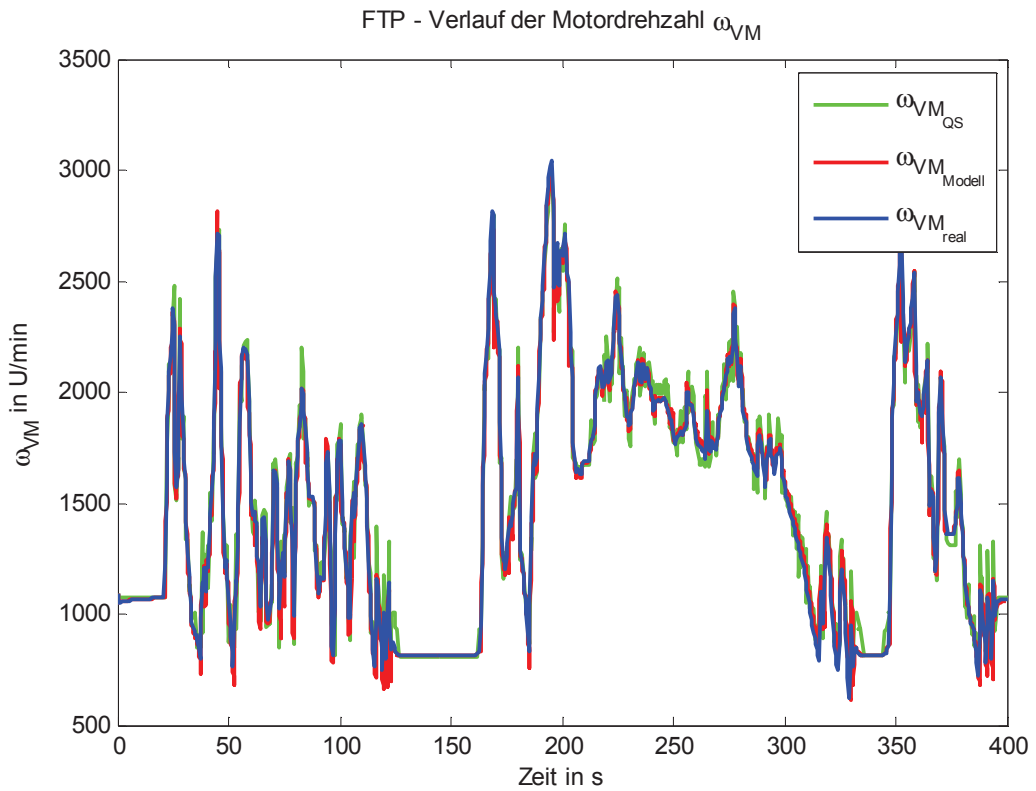


Abbildung 2.12: Modellvalidierung anhand des FTP-Fahrzyklus

2.5 Einbindung von realen Fahrstrecken

Um die in Kapitel 4 entworfene Regelung so realitätsnahe wie möglich testen und auf die potentielle Treibstoffersparnis untersuchen zu können, ist es sinnvoll, die Simulationen anhand von realen Strecken durchzuführen. Die aufgenommenen Streckendaten müssen, um für die Regelung verwendet werden zu können, sowohl in ein Geschwindigkeitsprofil, als auch ein Höhenprofil der Strecke umgerechnet werden. Die Daten einer aufgenommenen Strecke liegen dabei im GPX-Format [6] vor und müssen für die Simulation in geeignete Daten konvertiert werden. Das GPX-Format ist ein lizenzfreies Format zur Verarbeitung und zum Austausch von GPS-Daten. Die in der GPX-Datei gespeicherten Daten liegen dabei im Textformat vor und bestehen aus den geografischen Daten (Längen- und Breitengrad), der Seehöhe und einer Zeitinformation. Aus diesen Informationen lässt sich durch einfache mathematische Umrechnungen¹⁴ mit Hilfe von MATLAB® ein Streckenprofil in Abhängigkeit von der Zeit oder dem Weg berechnen.

Abbildung 2.13 und Abbildung 2.15 zeigen jeweils eine der verwendeten Routen in Google Maps¹⁵, welche mittels eines GPS-Geräts aufgenommen wurden. Aus der vom GPS-Gerät erstellten GPX-Datei können in Folge, wie in Abbildung 2.14 und Abbildung 2.16 dargestellt, die Verläufe für das Geschwindigkeits- und Höhenprofil der abgefahrenen Strecke berechnet werden.

Es stehen in der Folge 5 verschiedene Routen für Simulationen zur Verfügung, 2 verschiedene Routen von Graz nach Wolfsberg und 3 Routen in Budapest. Abbildung 2.13 und Abbildung 2.15 dienen als Beispielrouten, alle anderen verwendeten Routen können aus dem Anhang entnommen werden.

¹⁴Mittels MATLAB®-Funktion `loadgpx.m` als Download:

<http://www.mathworks.com/matlabcentral/fileexchange/24154-gpx-file-reader/content/loadgpx.m> Jänner 2013

¹⁵ Mittels MATLAB®-Funktion `plot_google_map.m` als Download:

<http://www.mathworks.com/matlabcentral/fileexchange/27627> Jänner 2013

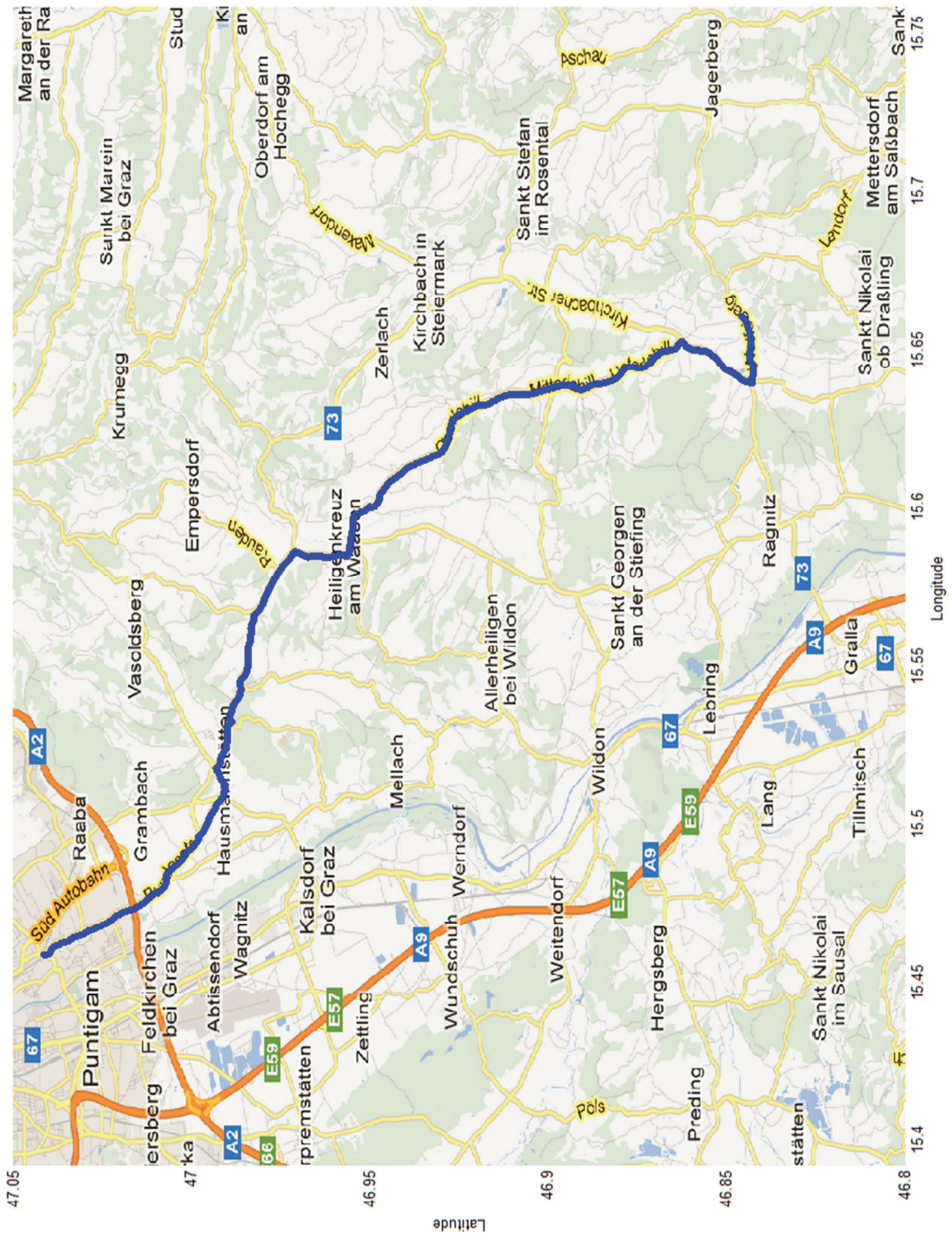


Abbildung 2.13: Route Graz- Wolfsberg 1 in Google Maps

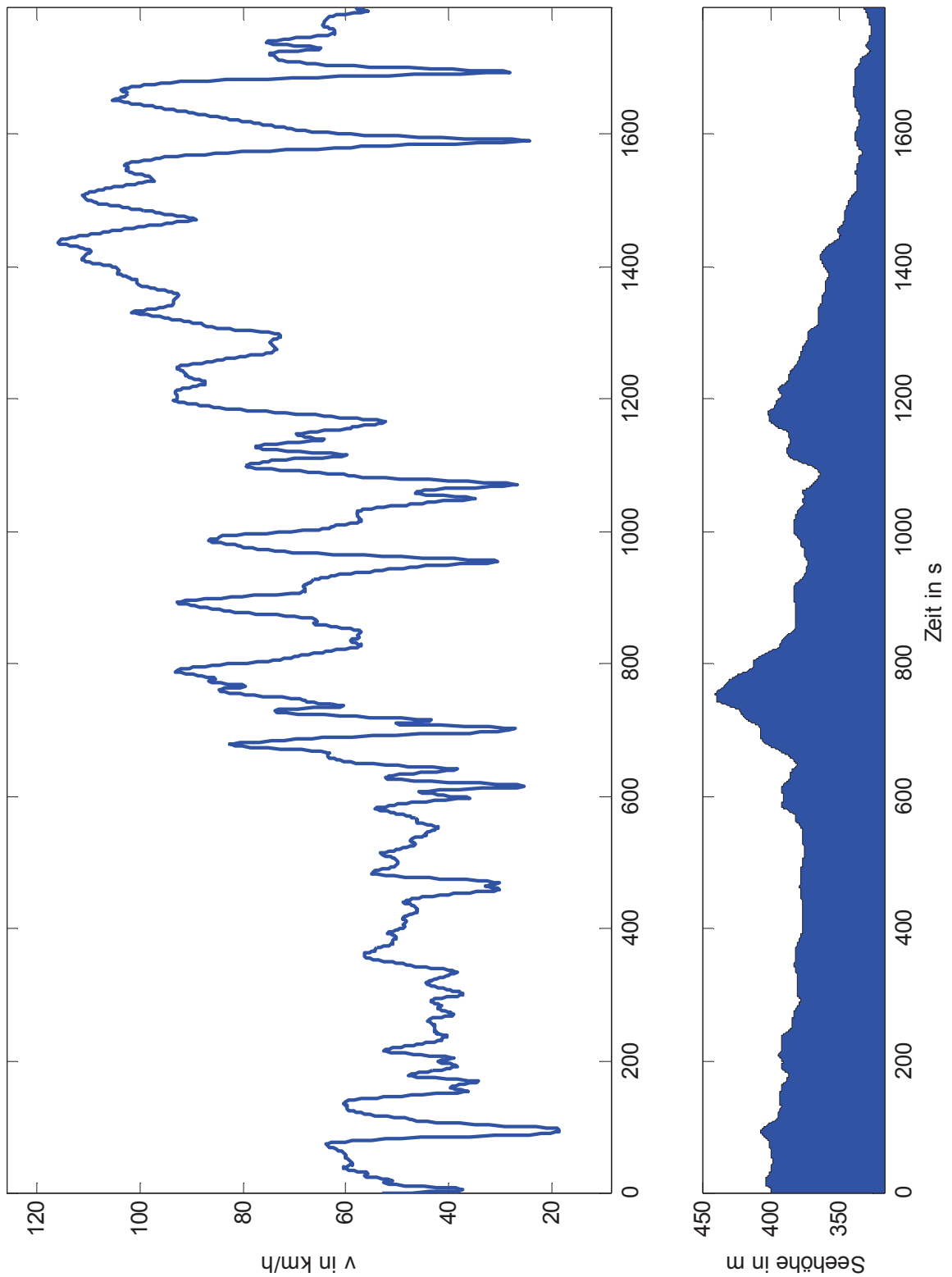


Abbildung 2.14: Geschwindigkeits- und Höhenprofil der Route Graz – Wolfsberg 1

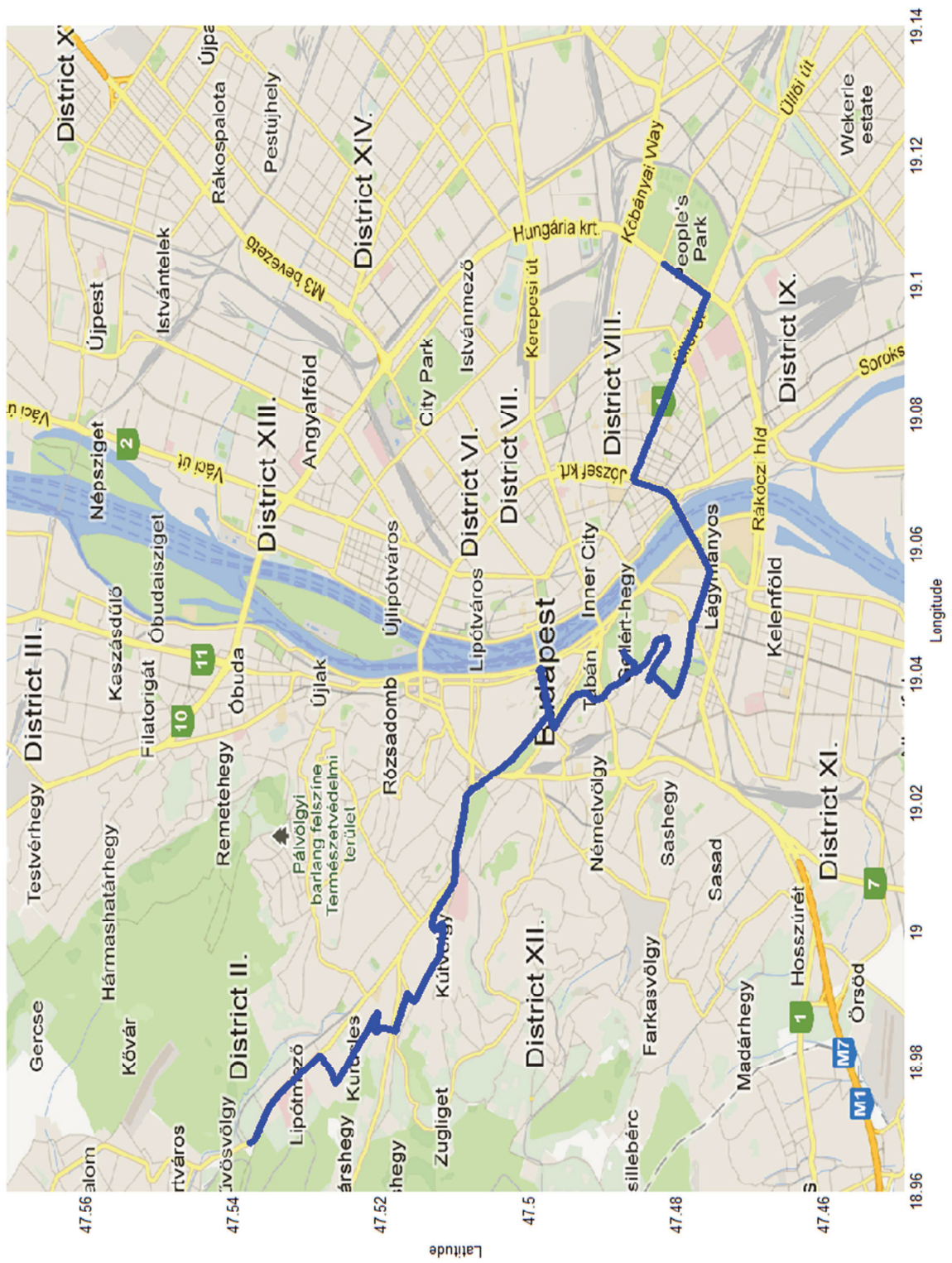


Abbildung 2.15: Route Budapest 1 in Google Maps

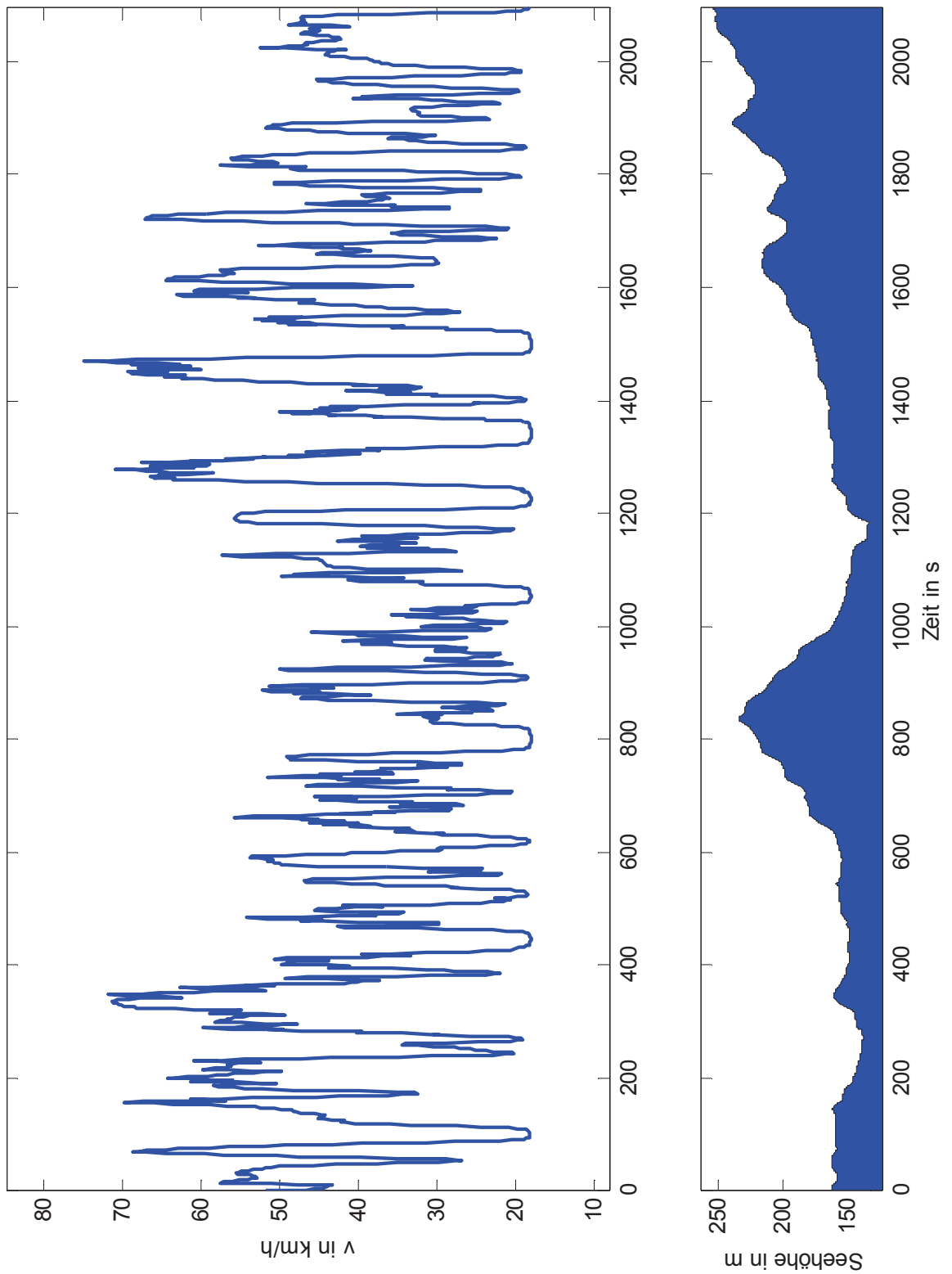


Abbildung 2.16: Geschwindigkeits- und Höhenprofil der Route Budapest 1

3 Dynamische Programmierung

In diesem Kapitel wird die dynamische Programmierung als Methode zur Lösung des vorliegenden Optimierungsproblems vorgestellt. Die grundlegende Beschreibung erfolgt dabei durch das Optimalitätsprinzip von Bellman [7]. Im Abschnitt 3.1 werden die Grundlagen der dynamischen Programmierung anhand eines einfachen Beispiels erläutert. Des Weiteren wird auf die Anwendung des Prinzips auf dynamische Systeme näher eingegangen. Die Implementierung der Optimierungsmethode in MATLAB® wird in Abschnitt 3.2 beschrieben. Abschließend werden im Abschnitt 3.3 die Anwendung der dynamischen Programmierung auf das in dieser Arbeit verwendete Fahrzeugmodell beschrieben und Simulationsergebnisse anhand von Fahrzyklen aus Abschnitt 2.5 vorgestellt.

3.1 Grundlagen der dynamischen Programmierung

Das Prinzip der dynamischen Programmierung beruht auf dem Optimalitätsprinzip von Bellman, welches in [7] von Richard Bellman im Jahre 1957 erstmals vorgestellt wurde:

„Eine optimale Entscheidungsfolge hat die Eigenschaft, dass, wie auch immer der Anfangszustand war und die erste Entscheidung ausfiel, die verbleibenden Entscheidungen ebenfalls eine optimale Entscheidungsfolge bilden müssen, betrachtet über alle möglichen Entscheidungsfolgen, deren Anfang bei dem Zustand liegt, der aus der ersten Entscheidung resultiert.“ - Bellman, 1957

Die grundlegende Idee liegt in der Betrachtung von Problemen, bei denen schrittweise Entscheidungen erforderlich sind. Kann nun das Gesamtproblem in eine Reihe von kleineren Teilproblemen aufgeteilt werden und für jedes der Teilprobleme eine optimale

Lösung gefunden werden, ergibt sich aus der Summe der optimalen Teillösungen eine optimale Gesamtlösung für das Optimierungsproblem. Eine wichtige Voraussetzung ist dabei die unabhängige Lösbarkeit der Teilsysteme. Die jeweilige Lösung der Teilprobleme wird dabei gespeichert und bei erneutem Bedarf zur Bestimmung der Gesamtlösung wieder aufgerufen.

3.1.1 Einführendes Beispiel - Wegenetz

Die Suche nach dem optimalen Weg in einem Wegenetz stellt ein mögliches Problem dar, welches mit Hilfe der dynamischen Programmierung gelöst werden kann. Diese Klasse von Problemen eignet sich dabei sehr gut für eine anschauliche Demonstration des Prinzips der dynamischen Programmierung und wird in [8] ausführlich erklärt. Dabei unterscheidet man zwischen 4 verschiedenen Problemfällen, jeweils mit freiem/ festem Anfangspunkt und freiem/ festem Endpunkt.

In Anlehnung an die spätere Anwendung der dynamischen Programmierung auf das Fahrzeugmodell soll die optimale Trajektorie für das in Abbildung 3.1 dargestellte Wegenetz mit fixem Anfangspunkt und freiem Endpunkt gefunden werden.

Gesucht ist die von einem fixen Startpunkt aus minimal nötige Zeit, um in 5 Entscheidungsschritten ein in Abbildung 3.1 dargestelltes Wegenetz zu durchschreiten. Die Endposition am Ende des Wegenetzes ist dabei frei. Jedem einzelnen Wegstück im Wegenetz sind verschiedene Gewichtungen $h_k(x_k, u_k)$ zugeordnet, welche der benötigten Zeit für die Bewegung von einem Punkt zum nächsten entsprechen. Es gibt dabei in jedem Punkt $P_k(x_k)$, abgesehen von den Randpunkten, jeweils die Möglichkeit für $u_k(x_k)$

$$u_k(x_k) = \begin{cases} +1 & \hat{=} \text{aufwärts} \\ 0 & \hat{=} \text{eben} \\ -1 & \hat{=} \text{abwärts,} \end{cases} \quad (3.1)$$

um zum nächsten Wegepunkt entlang der Abszisse zu gelangen.

Ziel ist die Minimierung der Kostenfunktion

$$J_0(x_0) = \sum_{k=0}^4 h_k(x_k, u_k) + J_5(x_5). \quad (3.2)$$

Dies entspricht der entlang des gewählten Weges zur anderen Seite des Wegenetzes vergangenen Zeit, bestehend aus der Summe der benötigten Zeit je Wegstück $h_k(x_k, u_k)$ und der Bewertung der Endpunkte $J_5(x_5)$.

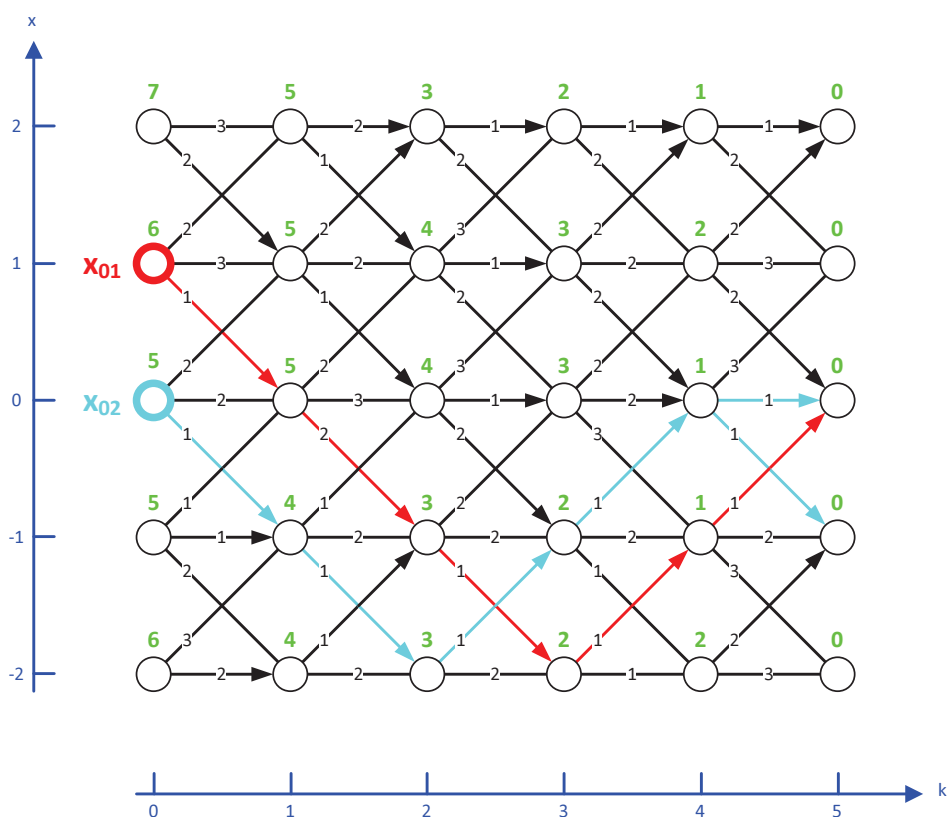


Abbildung 3.1: Rückwärts-Rechnung bei freiem Endpunkt.

Optimale Trajektorie mit festem Anfangspunkt

Eine Möglichkeit der Lösung dieses Optimierungsproblems nennt man Rückwärts-Rechnung. Dabei wird jeweils von den möglichen Endpunkten rückwärts im Wegenetz das Problem mit $N = 5$ Schritten entlang der Abszisse wie folgt gelöst:

- (1) Da es sich bei dem betrachteten Beispiel um ein Problem mit freiem Endpunkt handelt, wird jedem der Endpunkte $P_5(x_5)$ ein Kostenwert $J_5(x_5) = 0$ (in

Abbildung 3.1 grün beschriftet) zugewiesen. Somit ist jeder der Endpunkte gleich attraktiv.

- (2) Ausgehend vom letzten Schritt $k=5$ werden nun bei jedem Punkt $P_5(x_5)$ alle möglichen Bewegungsrichtungen, welche durch die Gewichtung $h_4(x_4, u_4)$ beschrieben sind, einen Schritt rückwärts betrachtet. Um zum Beispiel vom Punkt $P_4(-1)$ zu einem Endpunkt zu gelangen, gibt es 3 Alternativen mit den resultierenden Kosten:

$$\begin{aligned} \text{aufwärts} & : J_4(-1) = J_5(0) + h_4(-1, 1) = 0 + 1 = 1 \longrightarrow \text{Minimum} \\ \text{eben} & : J_4(-1) = J_5(-1) + h_4(-1, 0) = 0 + 2 = 2 \\ \text{abwärts} & : J_4(-1) = J_5(-2) + h_4(-1, -1) = 0 + 3 = 3 \end{aligned} \quad (3.3)$$

Da die Kostenfunktion minimiert werden soll, wird im Punkt $P_4(-1)$ der minimale Kostenwert $J_4(-1) = 1$ ($\hat{=}$ minimaler Zeitwert um zu einem beliebigen Endpunkt zu gelangen) und die entsprechende optimale Bewegungsrichtung $u_4(-1) = 1$ gespeichert. In Abbildung 3.1 wird dies in Form von Pfeilen dargestellt.

Diese Vorgangsweise wird für jeden Punkt $P_4(x_4)$ wiederholt und es ergeben sich daraus jeweils die laufenden Kosten¹⁶ $J_4(x_4)$ und die optimale(n) Bewegungsrichtung(en) $u_4(x_4)$ für jeden Punkt, um in minimaler Zeit zu einem beliebigen Endpunkt zu gelangen.

- (3) Analog zum vorhergehenden Schritt werden die noch verbleibenden Wegpunkte im Wegenetz für $k=3, \dots, 0$ rückwärts ausgewertet. Diese Auswertung führt schließlich zu den optimalen Kostenwerten $J_k(x_k)$ und optimalen Bewegungsrichtungen $u_k(x_k)$ in jedem Punkt $P_k(x_k)$ des Wegenetzes.

¹⁶ engl. cost-to-go function

(4) Durch die Rückwärts-Rechnung kann nun für jeden beliebigen Startwert $P_0(x_0)$ eine optimale Steuerfolge zum Durchschreiten des Wegenetzes in minimaler Zeit gefunden werden. Ist zum Beispiel der Startpunkt $P_0(1) = x_{01}$ vorgegeben ergibt sich eine optimale Bewegungsfolge durch die Auswertung in Vorwärts-Richtung zu:

$$u_{1,k} = (-1, -1, -1, +1, +1), \quad k = 0, 1, \dots, N-1 \quad (3.4)$$

Daraus folgt der in Abbildung 3.1 in Rot dargestellte optimale Weg:

$$P_0(1) \rightarrow P_1(0) \rightarrow P_2(-1) \rightarrow P_3(-2) \rightarrow P_4(-1) \rightarrow P_5(0) \quad (3.5)$$

welcher, dem Startwert entsprechend, in $J_0(1) = 6$ Zeiteinheiten durchschritten werden kann.

Wird nun ein anderer Startwert $P_0(0) = x_{02}$ gewählt ergeben sich 2 wie in Abbildung 3.1 in blau dargestellt Wege, die in unterschiedliche Endpunkte führen zu:

$$\begin{aligned} P_0(0) &\rightarrow P_1(-1) \rightarrow P_2(-2) \rightarrow P_3(-1) \rightarrow P_4(0) \rightarrow P_5(0) \\ P_0(0) &\rightarrow P_1(-1) \rightarrow P_2(-2) \rightarrow P_3(-1) \rightarrow P_4(0) \rightarrow P_5(-1) \end{aligned} \quad (3.6)$$

Die Endpunkte $P_5(0)$ und $P_5(-1)$ können dabei entsprechend dem optimalen Kostenfunktionswert $J_0(0) = 5$, dem Startwert $P_0(0)$, in derselben Zeit erreicht werden und bilden daher eine gleichwertige optimale Lösung.

Ein Vorteil der Rückwärts-Rechnung liegt nun darin, dass der Startwert für die Optimierung erst nach Beendigung der Rückwärts-Rechnung bekannt sein muss. Die Rückwärts-Rechnung stellt dabei den rechentechnisch aufwändigeren Teil der Optimierung mit Hilfe der dynamischen Programmierung dar. Somit kann ein Teil des Problems schon vorweg berechnet werden und der jeweilige Startpunkt muss erst für die eigentliche Bestimmung der optimalen Steuerfolge bekannt sein.

3.1 Grundlagen der dynamischen Programmierung

Die Anwendung des oben beschriebenen zweiten Lösungsschritts auf das Wegenetz zeigt das Optimalitätsprinzip von Bellman anhand der Unterteilung in Teilprobleme in Form von Schritten rückwärts im Wegenetz. Formuliert für einen festen Anfangs- und freien Endpunkt besagt es im Fall der Rückwärts-Rechnung [8]:

Durchläuft ein System nacheinander die $N+1$ Zustände x_0, x_1, \dots, x_N ausgehend vom Anfangszustand x_0 , hervorgerufen durch die N -Entscheidungen $(u_0, u_1, \dots, u_{N-1})$, optimal

bezüglich der Kostenfunktion $J_0(x_0) = \sum_{k=0}^{N-1} h_k(x_k, u_k) + J_N(x_N)$, so ist auch der $(N-i)$ -

stufige Entscheidungsprozess, beschrieben durch die $(N-i)$ -Entscheidungen $(u_i, u_{i+1}, \dots, u_{N-1})$, ausgehend vom Zustand x_i , bezüglich der sogenannten

Restkostenfunktion $J_i(x_i) = \sum_{k=i}^{N-1} h_k(x_k, u_k) + J_N(x_N)$ optimal.

Übertragen auf die allgemeine Formulierung für dynamische Systeme ergibt sich folgender Vergleich zum zuvor beschriebenen Problem für das Finden des schnellsten Wegs:

dynamisches System	$\hat{=}$	Wegenetz
Stellgröße u_k	$\hat{=}$	Wahl der Bewegungsrichtung ausgehend vom bereits erreichten Punkt
N -stufiger Entscheidungsprozess	$\hat{=}$	Bestimmung des schnellsten Weges ausgehend vom Startpunkt
Zustand x_k	$\hat{=}$	erreichter Wegepunkt
Kostenwert $h_k(x_k, u_k)$ zur Stellgröße u_k und dem Zustand x_k	$\hat{=}$	Zeit zum nächsten Wegepunkt in Richtung u_k
Kostenfunktion $J_0 = \sum_{k=0}^{N-1} h_k(x_k, u_k) + J_N(x_N)$ für N -stufigen Entscheidungsprozess	$\hat{=}$	Zeit zum Durchschreiten des vollständigen Weges

3.1.2 Anwendung auf dynamische Systeme

Werden dynamische Systeme nach Gl.(3.7), mit a-priori bekannten Störungen \mathbf{w}_k , betrachtet, ist es sinnvoll die Lösung des Optimierungsproblems mittels dynamischer Programmierung nach dem Optimalitätsprinzip von Bellman wie folgt zu formulieren:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \quad k=0,1,\dots,N-1 \quad (3.7)$$

$$J_0(\mathbf{x}_0) = \sum_{k=0}^{N-1} h_k(\mathbf{x}_k, \mathbf{u}_k) + J_N(\mathbf{x}_N) \quad (3.8)$$

$$\begin{aligned} \mathbf{x}_k &\in \mathbf{X}_k^n \\ \mathbf{u}_k &\in \mathbf{U}_k^m \\ \mathbf{w}_k &\in \mathbf{D}_k^d. \end{aligned} \quad (3.9)$$

Soll nun eine optimale Steuerfolge \mathbf{u}_k^o zur Minimierung der Kostenfunktion nach Gl.(3.8) ausgehend vom Anfangszustand \mathbf{x}_0 gefunden werden, müssen die finalen Kosten $J_N(\mathbf{x}_N)$ vor dem Start für jeden möglichen Endzustand festgelegt werden. Bei freiem Endzustand gilt $J_N(\mathbf{x}_N) = 0$.

Die Vorgehensweise rückwärts in der Zeit zur Lösung bei dynamischen Systemen lautet somit für $k = N-1, N-2, \dots, 0$:

- (1) Lösen der Systemgleichung \mathbf{x}_{k+1} für jede mögliche Kombination aus \mathbf{x}_k und \mathbf{u}_k
Bestimmung der Kostenfunktion $J_k(\mathbf{x}_k)$ mittels Minimierung von:

$$J_k(\mathbf{x}_k) = \min_{\mathbf{u}_k} \{ h_k(\mathbf{x}_k, \mathbf{u}_k) + J_{k+1}(\mathbf{x}_{k+1}) \} \quad (3.10)$$

Das heißt, für jeden Zustand \mathbf{x}_k kann eine Stellgröße \mathbf{u}_k^o bestimmt und gespeichert werden, welche die Kostenfunktion nach Gl.(3.10) minimiert.

- (2) $k = k-1$ falls $k > 0$ und anschließend erneute Ausführung von Schritt (1) und (2) bis $k = 0$

- (3) Zum Zeitpunkt $k=0$, kann nun für jeden beliebigen Anfangszustand \mathbf{x}_0 eine optimale Steuerfolge \mathbf{u}_k^o für $k=0,1,\dots,N-1$ bestimmt werden.

3.1.3 Abschätzung des Rechenaufwands

Wie zuvor erwähnt, ist bei der dynamischen Programmierung für jeden Punkt im Zustandsraum und für jeden Zeitpunkt eine Minimierung der Kostenfunktion nach Gl.(3.10) nötig. Aus diesem Grund können nur diskrete Werte des Zustandsraums und der Stellgrößen betrachtet werden. Daraus lässt sich klar erkennen, dass die Anzahl der nötigen Auswertungen stark von der Wahl der Diskretisierung des Zustandsraums und der Stellgrößen abhängt. Für ein Problem, welches für N diskrete Zeitschritte gelöst werden soll, sei die Rechenzeit τ für eine Auswertung der Kostenfunktion nach Gl.(3.10) bekannt. Geht man von einem Problem mit n Zustandsvariablen mit jeweils Z Stützstellen und m Stellgrößen mit jeweils M Stützstellen aus, ergibt sich folgender Zusammenhang für die Rechenzeit:

$$T_R = N \cdot Z^n \cdot M^m \cdot \tau \quad (3.11)$$

Aus Gl.(3.11) lässt sich klar der exponentielle Zuwachs an Rechenzeit mit der Anzahl an Zustandsgrößen und Stellgrößen und ein lineares Wachstum mit der Problemlänge erkennen. Aus diesem Grund sind die folgenden Überlegungen zur Reduktion der nötigen Rechenzeit naheliegend:

- **Reduktion der Modellordnung:**

Eine Reduktion der Modellordnung wirkt sich erheblich (exponentiell) auf die Rechenzeit der dynamischen Programmierung aus. Aus diesem Grund sollten, falls möglich, nur für das Systemverhalten relevante Zustandsgrößen dynamisch modelliert werden. Im Allgemeinen gilt die Betrachtung von Systemen 2. oder 3. Ordnung noch als effizient lösbar.

- **Begrenzung des Zustandsraums:**

Die Betrachtung eines begrenzten Zustandsraums und eine gröbere Diskretisierung des Zustandsraums verkürzen ebenfalls die Rechenzeit, können jedoch zu einer schlechteren Güte durch ungenauere Interpolation führen. Die Begrenzung des betrachteten Zustandsraums lässt jedoch die Option offen, nach der Lösung in einem begrenzten Gebiet zu suchen und bildet daher einen weiteren Vorteil der dynamischen Programmierung.

- **Schrittweite:**

Eine größere Schrittweite, welche direkt die Problemlänge beeinflusst, hat einen linearen Einfluss auf die Berechnungsdauer. Eine zu große Schrittweite kann jedoch zu einem ungenauen oder sogar instabilen (siehe Punkt 2.3) Modellverhalten führen und reduziert die Optimierungsgüte.

- **Diskretisierung der Stellgrößen:**

Die Rechenzeit steigt exponentiell mit der Zahl an Stellgrößen, jedoch ist es in den meisten Fällen schwierig, diese Zahl zu minimieren, weshalb nur eine gröbere Diskretisierung der Stellgrößen zu einer kürzeren Rechenzeit führt. Eine zu grobe Diskretisierung führt jedoch, ähnlich wie beim Zustandsraum, zu einer geringeren Güte der Lösung.

3.2 Beschreibung der MATLAB®-Funktion *dpm.m*

Bei der MATLAB®-Funktion *dpm.m* vom Institut für Mess- und Regeltechnik der ETH Zürich [9] handelt es sich um eine Funktion zur Lösung von Optimierungsproblemen nach dem Optimalitätsprinzip von Bellman mit Hilfe der dynamischen Programmierung. Die MATLAB®-Funktion¹⁷ wurde 2009 von Sundström und Guzzella in [9] vorgestellt. Die Funktion unterstützt nicht-lineare, zeitvariante MIMO-Systeme bis zu 5.Ordnung.

¹⁷ Download unter <http://www.idsc.ethz.ch/Downloads/dpm> Oktober 2012

Allgemein kann die MATLAB®-Funktion `dpm.m` für die folgende Klasse von Optimierungsproblemen zur Bestimmung einer optimalen Steuerfolge verwendet werden:

$$\begin{aligned}
 & \min_{\mathbf{u}(t)} J(\mathbf{u}(t)) \\
 & \text{unter} \\
 & \dot{\mathbf{x}}(t) = F(\mathbf{x}(t), \mathbf{u}(t), t) \\
 & \mathbf{x}(0) = \mathbf{x}_0 \\
 & \mathbf{x}(t_f) \in [\mathbf{x}_{f,\min}, \mathbf{x}_{f,\max}] \\
 & \mathbf{x}(t) \in \mathbf{X}(t) \subset \mathfrak{R}^n \\
 & \mathbf{u}(t) \in \mathbf{U}(t) \subset \mathfrak{R}^m
 \end{aligned} \tag{3.12}$$

$$J(\mathbf{u}(t)) = G(\mathbf{x}(t_f)) + \int_0^{t_f} H(\mathbf{x}(t), \mathbf{u}(t), t) dt \tag{3.13}$$

Um das in Gleichung (3.12) und (3.13) angegebene allgemeine Optimierungsproblem nun mit Hilfe der MATLAB®-Funktion `dpm.m` lösen zu können, müssen das Modell und die Kostenfunktion noch zeitdiskretisiert werden. Es wird der Einfachheit halber nachfolgend von einem nichtlinearen, zeitinvarianten System 1. Ordnung mit einer Eingangsgröße ausgegangen.

Angenommen, für das zeitdiskrete dynamische System der Form

$$\mathbf{x}_{k+1} = F_k(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, 1, \dots, N-1 \tag{3.14}$$

existiert für den Anfangszustand $\mathbf{x}(0) = \mathbf{x}_0$, unter der Einhaltung der Zustands- und Stellgrößenbeschränkungen

$$\begin{aligned}
 \mathbf{x}_k & \in X_k \subset \mathfrak{R} \\
 \mathbf{u}_k & \in U_k \subset \mathfrak{R}^m
 \end{aligned} \tag{3.15}$$

eine begrenzte Anzahl an zulässigen Steuerfolgen $\mathbf{u}_k^\pi \in U_k^\pi \subseteq U_k$, mit den Kosten beschrieben durch die Kostenfunktion

$$J^\pi(\mathbf{x}_0) = g_N(\mathbf{x}_N) + \phi_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} (h_k(\mathbf{x}_k, \mathbf{u}_k^\pi) + \phi_k(\mathbf{x}_k)) \tag{3.16}$$

Mit dem Term $g_N(x_N) + \phi_N(x_N)$ der Kostenfunktion wird der Endzustand bewertet. Der erste Term $g_N(x_N)$ repräsentiert dabei die finalen Kosten im Endzustand. Der zweite Term bildet eine zusätzliche Möglichkeit zur „Bestrafung“, um Begrenzungen im Endzustand sicher zu stellen. Die Funktion $h_k(x_k, u_k^\pi)$ bildet die Kosten für die Anwendung der jeweiligen Stellgröße u_k^π im Zustand x_k . Mit dem Term $\phi_k(x_k)$, als Teil der Summe können wiederum Verletzungen von Beschränkungen im Zustandsraum für jeden Zeitpunkt $k=0,1,\dots,N-1$ „bestraft“ werden. Gesucht sei nun die optimale Steuerfolge u_k^o , aus der Menge an zulässigen Steuerfolgen, mit den minimalen Kosten beschrieben durch:

$$J^o(x_0) = \min_{u_k^o \in \tilde{U}_k} \{J^\pi(x_0)\}. \quad (3.17)$$

Die MATLAB®-Funktion benötigt zur Berechnung der optimalen Steuerfolge u_k^o eine MATLAB®-Funktion, z.B.: *model.m* der Form:

```
function [X C I out] = model(inp,par)
```

welche mittels Matrix-Operationen die Systemgleichungen des dynamischen Modells wie in Gl.(3.14) beschreiben und die Kosten $h_k(x_k, u_k)$ und $\phi_k(x_k)$ liefert. Die detaillierten Zusammenhänge für die Ein- und Ausgabevariablen können aus [9] entnommen werden. Des Weiteren ist die Übergabe der in Tabelle 3.1 beschriebenen Parameter nötig.

Parameter	Beschreibung
prb.N	Problemlänge
prb.NO	Index für Vorwärts-Rechnung
prb.Ts	Zeitintervall
prb.W $\{ \cdot \}$	Hilfsvektor für zeitvariable Daten des Modells
prb.Nx $\{ i \}$	Anzahl der Stützstellen für die Zustandsgröße x_i
prb.Xn $\{ i \}$.lo	Untere Grenze der Zustandsgröße x_i
prb.Xn $\{ i \}$.hi	Obere Grenze der Zustandsgröße x_i
prb.XN $\{ i \}$.lo	Untere Grenze des Endzustands der Zustandsgröße x_i
prb.XN $\{ i \}$.hi	Obere Grenze des Endzustands der Zustandsgröße x_i
prb.X0 $\{ i \}$	Anfangswert der Zustandsgröße x_i
prb.Nu $\{ j \}$	Anzahl der Stützstellen der Stellgröße u_j
prb.Un $\{ j \}$.lo	Untere Grenze der Stellgröße u_j
prb.Un $\{ j \}$.hi	Obere Grenze der Stellgröße u_j
options.InfCost	Wert zur Bewertung von unzulässigen Zustandsübergängen
options.InputType	Kontinuierliche oder diskrete Stellgröße

Tabelle 3.1: Eingabeparameter der MATLAB®-Funktion dpm.m

Die in Tabelle 3.1 mit $\{ i \}$ und $\{ j \}$ gekennzeichneten Argumente bilden den Index der jeweiligen Zustandsgröße x_i oder Stellgröße u_j .

Unter der Annahme eines Systems 1. Ordnung mit nur einer Stellgröße kann die Auswertung des Optimierungsproblems mittels der MATLAB®-Funktion dpm.m wie folgt, nach der in [3] beschriebenen „Set Implementation“ zusammengefasst werden. Handelt es sich bei dem zu lösenden Optimierungsproblem um ein Problem der Länge N , wird zunächst die Modell-spezifische Funktion N -mal im Zuge der Rückwärts-Rechnung aufgerufen. Unter der Annahme einer Zustandsgröße $inp.X\{1\} \in [0, 3]$ mit $Z=4$

Stützstellen und einer Stellgröße $inp.U\{1\} \in [-1, 1]$ mit $M=5$ Stützstellen werden zu jedem Zeitpunkt k die folgenden Matrizen bestimmt:

- $X\{1\} = x_{k+1}$ zur Auswertung der Systemgleichungen nach (3.14) für den nächsten Zeitschritt $k+1$,
- $C\{1\} = h_k(x_k, u_k)$ für die Kostenwerte der Kombinationen aus Zustands- und Stellgröße und
- $I = \phi_k(x_k)$ zur Bewertung von unzulässigen Kombinationen aus Zustands- und Stellgröße.

Für den vorliegenden Fall ergeben sich dabei stets $(Z \times M)$ Matrizen, die jede mögliche Kombination aus Zustands- und Stellgröße abdecken. Daraus ergeben sich für die Zustands- und Stellgröße als Eingänge für die Funktion folgende Matrizen:

$$inp.X\{1\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix} \quad (3.18)$$

$$inp.U\{1\} = \begin{bmatrix} -1 & -0,5 & 0 & 0,5 & 1 \\ -1 & -0,5 & 0 & 0,5 & 1 \\ -1 & -0,5 & 0 & 0,5 & 1 \\ -1 & -0,5 & 0 & 0,5 & 1 \end{bmatrix} \quad (3.19)$$

Werden die Matrizen (3.18) und (3.19) elementweise in die Systemgleichung und die Kostenfunktion eingesetzt, können mit dieser Art der Implementierung, der gesamte Zustands- und Stellgrößenraum bzw. die jeweils möglichen Kombinationen mit nur einem Funktionsaufruf abgedeckt werden. Die nötige Auswertung und, falls nötig lineare Interpolation der Kostenfunktion zur Bestimmung der optimalen Stellgröße für den jeweiligen Zustand, wird von der MATLAB®-Funktion *dpm.m* automatisch durchgeführt. Im Zuge der Vorwärts-Rechnung kann mit der Vorgabe eines Anfangszustands die optimale Steuerfolge u_k^o bestimmt werden.

3.3 Anwendung auf das Fahrzeugmodell

Um die in Abschnitt 3.2 beschriebene MATLAB®-Funktion für die Implementierung einer verbrauchsoptimalen Steuerung für das Fahrzeugmodell aus Abschnitt 2.3 verwenden zu können, müssen eine geeignete MATLAB®-Funktion erstellt und die in Tabelle 3.1 beschriebenen Parameter geeignet vorgegeben werden.

3.3.1 Anpassung des Fahrzeugmodells

Da im Zuge dieser Arbeit eine verbrauchsoptimale Regelung des Fahrzeugs erstellt werden soll, welche die drei Stellgrößen Gaspedalstellung, Bremspedalkraft und Wahl des Gangs selbstständig wählt, liegt es nahe, diese drei Größen als Stellgrößen für die Optimierung zu wählen. Da jedoch die gleichzeitige Benutzung des Gas- und Bremspedals unzulässig ist, muss diese Einschränkung beim Entwurf mit berücksichtigt werden. Der einfachste Weg dies zu realisieren, ist das Zusammenfassen der Gaspedalstellung und Bremspedalkraft in einer Stellgröße. Dabei gilt die folgende Relation:

$$\begin{aligned} \alpha_{GP,k} &= \begin{cases} u_{1,k} & \forall u_{1,k} \geq 0 \\ 0 & \text{sonst} \end{cases} \\ F_{BP,k} &= \begin{cases} -u_{1,k} & \forall u_{1,k} < 0 \\ 0 & \text{sonst} \end{cases} \end{aligned} \quad (3.20)$$

Für den positiven Bereich der Stellgröße $u_{1,k}$ gilt dabei die gleiche Beschränkung wie für die Gaspedalstellung (Gl.(2.1)).

Um die Lösung des Optimierungsverfahrens schneller und effizienter durchführen zu können, ist es sinnvoll den Suchbereich für die Stellgröße $u_{1,k}$ vor dem Start der Optimierung abzuschätzen und somit zu jedem Zeitpunkt auf einen sinnvollen Suchbereich zu begrenzen. Für diese Abschätzung kann das a-priori bekannte, geplante Fahrprofil für die Fahrzeuggeschwindigkeit v_k , Beschleunigung \dot{v}_k und das Höhenprofil γ_k herangezogen werden. Mit Hilfe dieser Daten und fahrzeugspezifischer Parameter kann das zur Bewältigung der vorgegebenen Strecke nötige Drehmoment $M_{Rad_soll,k}$ vorweg

berechnet werden und somit ein Bereich für die Stellgröße $u_{1,k}$ abgeschätzt werden. Das für die Abschätzung nötige Drehmoment kann mit Hilfe von Gl.(2.18) berechnet werden, wobei sich folgender Zusammenhang ergibt:

$$M_{Rad_soll,k} = r_{Rad} \left[\dot{v}_k \cdot \frac{\Theta_{Rad} + m \cdot r_{Rad}^2}{r_{Rad}} + F_{Luft,k} + F_{Roll,k} + F_{Steigung,k} \right]. \quad (3.21)$$

Für das verwendete Fahrzeugmodell kann mit Hilfe des Faktors k_{u_1} die folgende obere und untere Grenze des Stellgrößensuchbereichs für jeden Zeitpunkt k spezifisch festgelegt werden:

$$\begin{aligned} u_{1_obG,k} &= \frac{M_{Rad_soll,k}}{k_{u_1}} + 40 \\ u_{1_unG,k} &= \frac{M_{Rad_soll,k}}{k_{u_1}} - 40 \end{aligned} \quad (3.22)$$

Die obere Grenze für die Stellgröße $u_{1,k}$ wird in Gl.(3.22) mit $u_{1_obG,k}$ bezeichnet, die untere Grenze mit $u_{1_unG,k}$. Sie dienen als Vorgabe für die in Abschnitt 3.2 vorgestellte MATLAB®-Funktion für den zeitvariablen Definitionsbereich der Stellgröße $u_{1,k}$.

Die Wahl des richtigen Gangs durch eine Stellgröße zu implementieren, führt mangels implementierbarer Anstiegsbeschränkungen in den Stellgrößen zu einem nicht praktikablen Verhalten, da ein unkontrolliertes Schalten von etwa dem 4. Gang in den 1. Gang innerhalb eines Zeitschritts möglich wird. Um den Fahrkomfort nicht negativ zu beeinflussen ist es sinnvoll, den aktuellen Gang in Form einer zusätzlichen Zustandsgröße zu berücksichtigen. Durch die Einführung einer neuen Stellgröße

$$u_{2,k} = \begin{cases} +1 & \hat{=} & \text{hinaufschalten} \\ 0 & \hat{=} & \text{nicht schalten} \\ -1 & \hat{=} & \text{herunterschalten} \end{cases} \quad (3.23)$$

kann das unkontrollierte Schalten zwischen den Gängen verhindert werden. Daraus ergibt sich die folgende zusätzliche Differenzgleichung für die Zustandsgröße des aktuellen Gangs:

$$g_{G,k+1} = g_{G,k} + u_{2,k} \cdot \quad (3.24)$$

Ein weiteres wichtiges Problem besteht nun darin, dass es durch diese Art der Implementierung weiterhin möglich ist, zu jedem beliebigen Zeitpunkt von einem Gang in den nächsten zu schalten. Bei einem Zeitintervall von $T_D = 0,5s$ ist es somit möglich alle $0,5s$ den Gang zu wechseln. Dieses Verhalten wirkt sich wiederum negativ auf den Fahrkomfort aus und sollte beim Entwurf berücksichtigt werden.

Eine Möglichkeit das Verhalten des zu schnellen Schaltens zu unterbinden, ist die Freigabe eines Schaltvorgangs nur zu bestimmten Zeitpunkten. Das heißt, es kann nur zu regelmäßigen (oder unregelmäßigen) Zeitpunkten der Gang gewechselt werden. Die Implementierung einer Schaltfreigabe kann mit Hilfe der in Abschnitt 3.2 vorgestellten MATLAB®-Funktion relativ einfach erfolgen, da externe zeitvariante Parameter für das Modell vorgegeben werden können. Durch die Erweiterung der Differenzgleichung des Gangs um eine Variable für die Schaltfreigabe $s_{f,k}$, welche nur die Werte $s_{f,k} \in \{0,1\}$ annehmen kann,

$$g_{G,k+1} = g_{G,k} + u_{2,k} \cdot s_{f,k} \quad (3.25)$$

kann nur noch zu den Zeitpunkten, an denen $s_{f,k} = 1$ ist, geschaltet werden. Für alle anderen Zeitpunkte, an denen $s_{f,k} = 0$ gilt, wirkt sich eine Änderung in der Eingangsgröße $u_{2,k}$ nicht aus und es ist daher kein Wechsel des Gangs möglich. Bei der Vorgabe von periodischen Schaltzeitpunkten mit einer Periode T_{schalt} muss jedoch beachtet werden, dass sich eine zu große Wahl des Zeitfensters, in dem nicht geschaltet werden darf, auf die Optimierungsgüte auswirken kann, da potentielle Gangwechsel zwischen den vorgegebenen Schaltpunkten bei der Optimierung ignoriert werden. Der Vorteil dieser Lösung ist, dass kein zusätzlicher Zähler, wie in [10] vorgestellt, welcher die Schaltvorgänge limitiert, in der Optimierung als zusätzliche Zustandsgröße implementiert

werden muss. Somit erhöht sich die Zeit zur Lösung des Optimierungsproblems nicht, ein zu schnelles Schalten wird jedoch verhindert. Der Nachteil liegt im Ignorieren einer potentiellen Kraftstoffersparnis durch andere Schaltzeitpunkte. Simulationsstudien ergaben jedoch, dass sich die Kraftstoffersparnis durch die Implementierung eines zusätzlichen Zählers wie in [10] minimal verbessert, jedoch in keiner Relation zum zusätzlichen Rechenaufwand, welcher für den Einsatz für die modellprädiktive Regelung entscheidend ist, steht. Da in Folge die einfachere und weniger rechenintensive Implementierung verwendet wird, wird hier nicht im Detail auf eine Realisierung mit dem zusätzlichen Zähler eingegangen.

Es ergibt sich in Folge das neue nichtlineare, zeitdiskrete Fahrzeugmodell der Form:

$$\omega_{VM,k+1} = \omega_{VM,k} + \frac{T_D}{\Theta_{VM} + \Theta_P} \cdot \left[M_{VM,k} - \frac{1}{\eta_G \cdot i(g_{G,k}) \cdot \mu_k} \left[\dot{v}_k \frac{\Theta_{Rad} + m \cdot r_{Rad}^2}{r_{Rad}} + r_{Rad} \cdot F_{Verlust,k} + M_{BR,k} \right] \right] \quad (3.26)$$

$$g_{G,k+1} = g_{G,k} + u_{2,k} \cdot s_{f,k}$$

Da im Zuge der Arbeit eine Geschwindigkeits-Regelung des Fahrzeugs realisiert werden soll, welche dem Fahrverlauf unter minimalem Kraftstoffeinsatz folgen kann, liegt es nahe die beiden Größen momentaner Verbrauch V_k und die Abweichung der Momentangeschwindigkeit des Fahrzeugs vom Sollverlauf in die Kostenfunktion zur Bewertung eines aktuellen Zustands einzubinden. Durch den Einsatz des quasistatischen Ansatzes zur Vereinfachung des dynamischen Modells geht die Fahrzeuggeschwindigkeit als dynamische Größe verloren und kann nicht in der Kostenfunktion zur Bewertung herangezogen werden. Auch wenn die Fahrzeuggeschwindigkeit keine dynamische Größe im Modell darstellt, kann die rechte Seite der Differentialgleichung für die Fahrzeuggeschwindigkeit (2.27) zu jedem Zeitpunkt und somit auch die Fahrzeugbeschleunigung zu jedem Zeitpunkt bestimmt werden. Ersetzt man die Abweichung der Geschwindigkeit im Kostenfunktional durch die Abweichung der Beschleunigung, kann eine ähnliche Form der Kostenfunktion berechnet werden. Um den numerischen Aufwand zur Bestimmung der Kostenfunktion weiter zu reduzieren, kann

statt der Differenz der Beschleunigungen, die Differenz der Pumpenmomente des Drehmomentwandlers verwendet werden. Diese Differenz setzt sich zum einen aus dem mittels quasistationären Ansatz berechneten Pumpendrehmoment

$$M_{P_1,k} = \frac{1}{\eta \cdot i(g_{G,k}) \cdot \mu_k(\omega_{VM,k}, g_{G,k}, v_k)} \left[\dot{v}_k \frac{\Theta_{Rad} + m \cdot r_{Rad}^2}{r_{Rad}} + r_{Rad} \cdot F_{Verlust,k}(v_k, \gamma_k) + M_{BR,k}(u_{1,k}) \right] \quad (3.27)$$

und zum anderen aus dem mittels der aus Abschnitt 2.1.2 definierten Gleichung für das Pumpendrehmoment:

$$M_{P_2,k} = \lambda_k(\omega_{VM,k}, g_{G,k}, v_k) \cdot D^5 \cdot \rho \cdot \omega_{VM,k}^2 \quad (3.28)$$

Zusammen. Aus dieser Überlegung ergibt sich, dass in der Kostenfunktion jenes Pumpenmoment $M_{P_1,k}$, vorgegeben durch die Fahrzeuggeschwindigkeit, die Streckenbeschaffenheit und das Bremsmoment, welches jenem tatsächlich vom Drehmomentwandler aufgenommenen Moment $M_{P_2,k}$ am besten entspricht, mit den geringsten Kosten bewertet wird. Es ergibt sich daher folgender Kostenwert zum Zeitpunkt k für die Kostenfunktion:

$$h_k(\mathbf{x}_k, \mathbf{u}_k) = h(\omega_{VM,k}, g_{G,k}, u_{1,k}) = k_1 \cdot V_{VM,k} + k_2 \cdot (M_{P_1,k} - M_{P_2,k})^2 \quad (3.29)$$

Da der Verbrauch $V_{VM,k}$ nur positive Werte annehmen kann und im Falle $u_{1,k} < 0$ die Beziehung $V_{VM,k} = 0$ gilt, ist der Kostenwert in Gl.(3.29) für jeden Zeitpunkt positiv. Die Konstanten k_1 und k_2 bilden Gewichtungsfaktoren und können variiert werden, müssen jedoch stets positive Werte annehmen.

3.3.2 Einstellungen für die MATLAB®-Funktion *dpm.m*

Wie in Abschnitt 3.2 bereits angeführt muss das Fahrzeugmodell, beschrieben durch Gl.(3.26) und die Funktion für den jeweiligen Kostenwert Gl.(3.29), in einer geeigneten MATLAB®-Funktion implementiert werden. Um diese Funktion korrekt aufrufen und ausführen zu können, müssen noch die in Tabelle 3.2 aufgelisteten Parameter vorgegeben werden. Die Parameter $v_k, \dot{v}_k, M_{Rad_soll,k}$ können direkt vom gegebenen Fahrverlauf berechnet werden und definieren gleichzeitig die Problemlänge $prb.N$. Dieser Parameter entscheidet maßgeblich über den Rechenaufwand und wird in weiterer Folge als konstant angenommen, da sich für größere Zeitschritte eine Instabilität im Fahrzeugmodell beobachten lässt bzw. die Optimierungsgüte stark abnimmt. Die Folge $s_{f,k}$ ist durch die Vorgabe einer Schaltperiode T_{schalt} ebenfalls bereits vor der Optimierung bekannt und kann als zeitvariante „Störgröße“ in der Optimierung berücksichtigt werden. Die Zustandsgröße $x_1 = \omega_{VM,k}$ und die Stellgröße $u_{1,k}$ bilden die beiden Größen, welche durch Variation der Diskretisierung Einfluss auf den Rechenaufwand haben. Die Grenzen von x_1 und $u_{1,k}$ sind im Vorfeld bereits bekannt bzw. können mittels Gl.(3.22) berechnet werden. Die Wertemenge der Zustandsgröße $x_2 = g_{G,k}$ und der Stellgröße $u_{2,k}$ ist fix vorgegeben und kann nicht variiert werden. Aus diesem Grund ergibt sich durch diese Größen kein Einfluss auf die Rechendauer.

Parameter	
prb.NO = 1	prb.Ts = 0.5
prb.W{1} = v_k	prb.W{2} = \dot{v}_k
prb.W{3} = $M_{Rad_soll,k}$	prb.W{4} = $s_{f,k}$
prb.Nx{1} = 21	prb.Nx{2} = 4
prb.Xn{1}.lo = $800 \cdot \pi/30$	prb.Xn{2}.lo = 0
prb.Xn{1}.hi = $4800 \cdot \pi/30$	prb.Xn{2}.hi = 3
prb.XN{1}.lo = $800 \cdot \pi/30$	prb.XN{2}.lo = 0
prb.XN{1}.hi = $4800 \cdot \pi/30$	prb.XN{2}.hi = 3
prb.Nu{1} = 17	prb.Nu{2} = 3
prb.Un{1}.lo = $u_{1_unG,k}$	prb.Un{2}.lo = -1
prb.Un{1}.hi = $u_{1_obG,k}$	prb.Un{2}.hi = 1
options.InfCost = 1000	options.InputType = 'cd'

Tabelle 3.2: Vorgegebene Parameter zur Lösung des Optimierungsproblems

Nach dem Lösen des Optimierungsproblems können die optimalen Steuerfolgen für die Gaspedalstellung und Bremspedalkraft aus Gl.(3.20) und die automatische Wahl des Gangs aus $g_{G,k}$ bzw. x_2 bestimmt werden. Diese bilden die Eingangsgrößen für das detaillierte Simulink-Fahrzeugmodell ab, womit das longitudinale Fahrzeugverhalten bestimmt werden kann.

3.3.3 Simulationsergebnisse

Die in Tabelle 3.3 dargestellten Simulationsergebnisse wurden mit verschiedenen Vorgaben für die Anzahl der Stützstellen $N_x\{1\}$ der Zustandsgröße $x_1 = \omega_{VM,k}$ und $N_u\{1\}$ der Stellgröße $u_{1,k}$ durchgeführt. Um die Auswirkungen verschiedener Schaltperioden T_{schalt} zu untersuchen, wurde jede Simulation mit drei verschiedenen, zulässigen Schaltperioden durchgeführt. Als Bewertungskriterium für die Güte der Lösung wurden die benötigte Rechenzeit zu Lösung des Optimierungsproblems, der durch das Befahren der Strecke entstandene Kraftstoffverbrauch, die Abweichung Δs der gefahrenen Strecke von der vorgegebenen Streckenlänge und die maximal auftretende Geschwindigkeitsabweichung $|e_v|_{\max}$ des Fahrzeugs vom vorgegeben Geschwindigkeitsprofil gewählt. Alle Simulationen aus Tabelle 3.3 wurden für die Strecke Graz-Wolfsberg (Abbildung 2.13) für die ersten 10min durchgeführt.

3.3 Anwendung auf das Fahrzeugmodell

Fahrzeit	T_D	T_{schalt}	$Nx\{1\}$	$Nu\{1\}$	Rechenzeit	Verbrauch	Δs	$ e_v _{max}$
min	s	s	-	-	s	l	m	km/h
10	0,5	1	11	9	9,3	0,3906	303,5	5,55
10	0,5	3	11	9	9,7	0,3895	307,9	5,58
10	0,5	5	11	9	9,3	0,3867	302,5	5,25
10	0,5	1	11	17	11,6	0,3847	169,8	3,43
10	0,5	3	11	17	11,5	0,3847	147,2	2,99
10	0,5	5	11	17	11,5	0,3842	152,1	2,89
10	0,5	1	11	33	15,8	0,377	191,1	3,61
10	0,5	3	11	33	15,6	0,3807	107,8	2,72
10	0,5	5	11	33	15,7	0,3756	111,5	2,8
10	0,5	1	21	9	11,6	0,388	321,8	4,72
10	0,5	3	21	9	11,5	0,3904	232,8	3,98
10	0,5	5	21	9	11,5	0,3875	230,4	3,93
10	0,5	1	21	17	15,5	0,3765	127,3	2,88
10	0,5	3	21	17	15,6	0,3799	125,8	3,04
10	0,5	5	21	17	15,4	0,3783	102,4	2,73
10	0,5	1	21	33	22,5	0,3601	156,8	3,21
10	0,5	3	21	33	22,6	0,3691	102,1	2,79
10	0,5	5	21	33	22,6	0,3681	102,9	2,77
10	0,5	1	41	17	22,9	0,3741	160,3	3,52
10	0,5	3	41	17	22,5	0,3787	91,6	2,84
10	0,5	5	41	17	22,6	0,3753	108,9	2,67
10	0,5	1	41	33	32,6	0,3545	164,2	3,02
10	0,5	3	41	33	32,7	0,3597	109,1	2,96
10	0,5	5	41	33	32,3	0,3593	101,5	2,88

Tabelle 3.3: Simulationsergebnisse für verschiedene Einstellungen der dynamischen Programmierung

Der in Tabelle 3.3 grau hinterlegte Bereich kennzeichnet die für den weiteren Einsatz gewählten Parameter, da diese eine gute Optimierungsgüte, bei akzeptablem Rechenaufwand aufweisen und somit einen guten Kompromiss, welcher unweigerlich eingegangen werden muss, darstellen.

Abbildung 3.2 zeigt den vorgegebene Geschwindigkeitsverlauf v_{soll} für die ersten 10min der Strecke Graz – Wolfsberg 1 und die mittels dynamischer Programmierung berechneten Stellgrößen, sowie die erreichte Geschwindigkeit v_{DP} über der Zeit. Die mittels dynamischer Programmierung berechneten optimalen Steuerfolgen sind in Abbildung 3.3 dargestellt und wurden mit den in Tabelle 3.3 grau hinterlegten Parametern mit $T_{schalt} = 3s$ berechnet.

Abbildung 3.4 zeigt die Simulationsergebnisse für die Strecke Budapest 3 mit denselben Einstellungen wie zuvor. Bei genauer Betrachtung der Verläufe von Gaspedalstellung und Bremskraft in Abbildung 3.3 und Abbildung 3.5 ist klar zu erkennen, dass es, wie gefordert, zu keinem Zeitpunkt zu einer gleichzeitigen Betätigung der beiden Pedale kommt.

Die in Abbildung 3.2 und Abbildung 3.4 auftretenden Abweichungen der Fahrzeuggeschwindigkeit vom vorgegebenen Sollverlauf entstehen durch die Berechnung einer Steuerfolge anhand eines vereinfachten Fahrzeugmodells und treten in Folge dieser Abweichungen vom detaillierten Simulink-Fahrzeugmodell auf. Um diese Abweichungen zu umgehen ist es sinnvoll, von einer optimalen Steuerung auf eine in Kapitel 4 beschriebene optimale Regelung über zu gehen.

3.3 Anwendung auf das Fahrzeugmodell

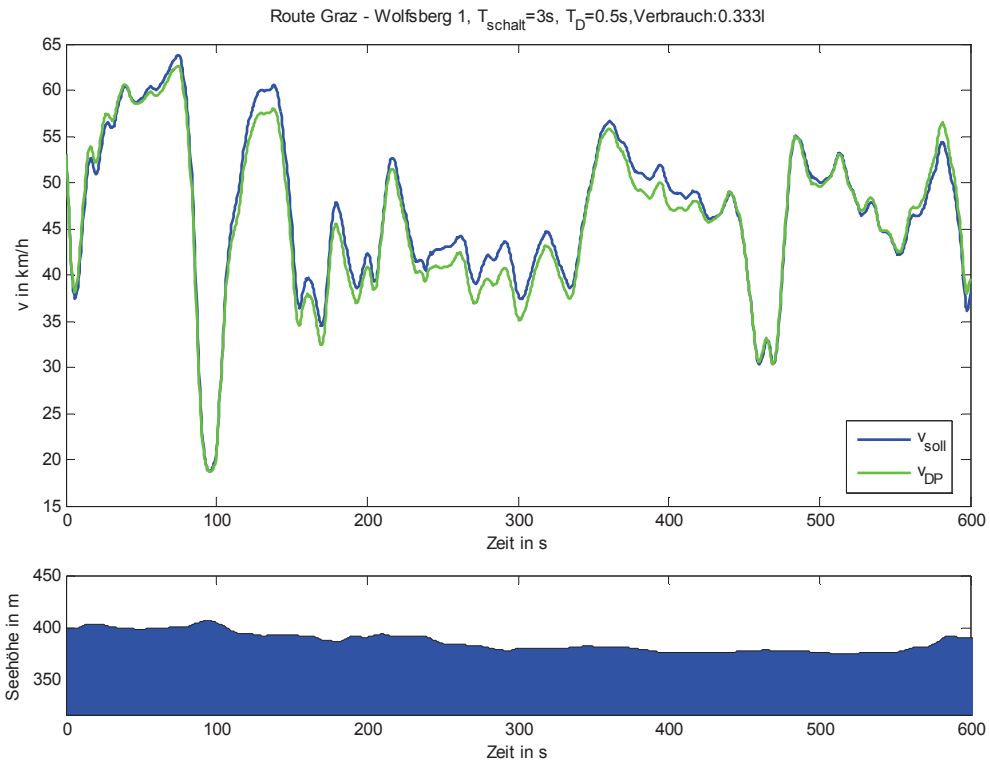


Abbildung 3.2: Geschwindigkeitsverlauf für optimale Steuerung der Route Graz–Wolfsberg 1

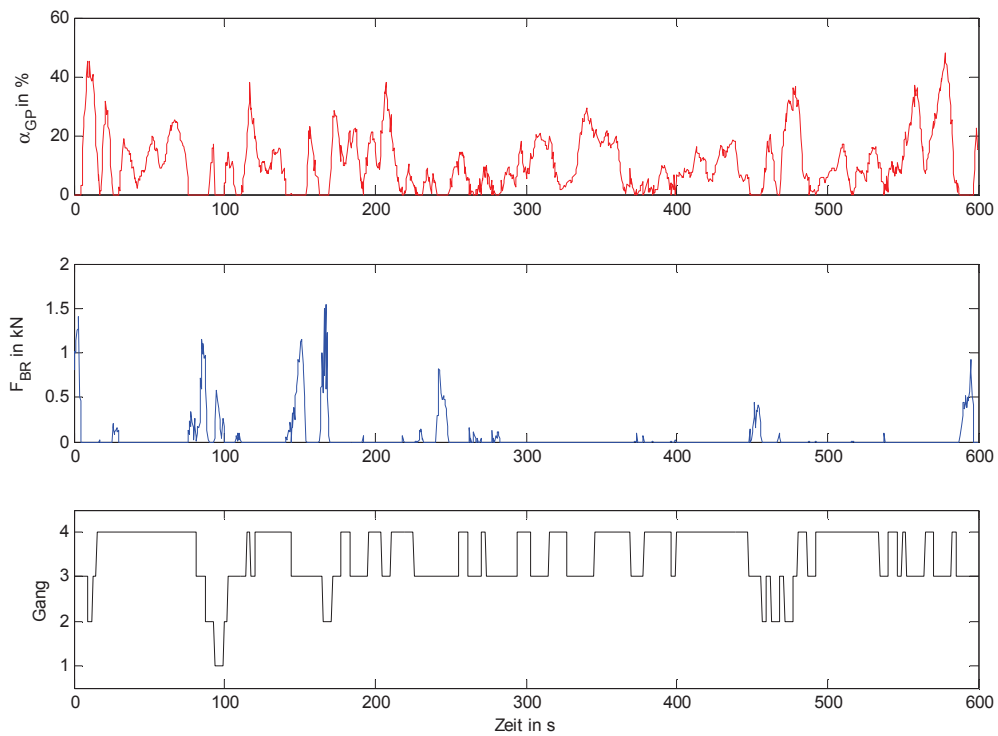


Abbildung 3.3: Berechnete optimale Steuerfolgen der 3 Stellgrößen, Route Graz–Wolfsberg 1

3 Dynamische Programmierung

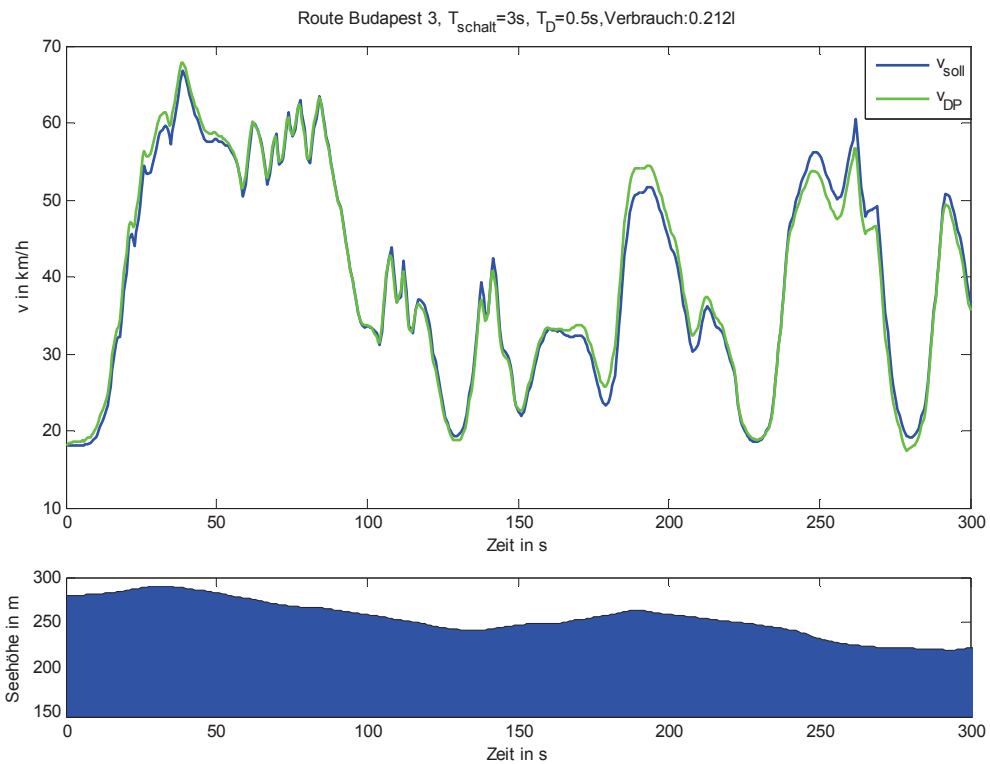


Abbildung 3.4: Geschwindigkeitsverlauf für optimale Steuerung der Route Budapest 3

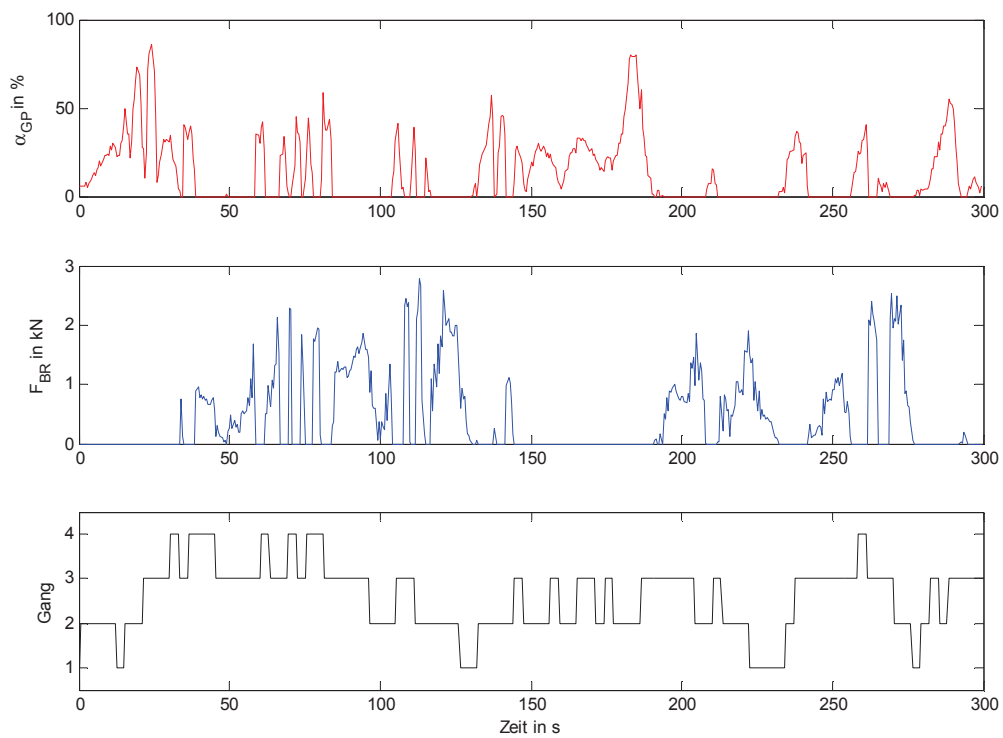


Abbildung 3.5: Berechnete optimale Steuerfolgen der 3 Stellgrößen, Route Budapest 3

4 Modellprädiktive Regelung

Wie in Abschnitt 3.3 bereits kurz erwähnt wurde, liefert die dynamische Programmierung nur eine optimale Steuerfolge. Daher ist es sinnvoll, eine Regelung, welche die auftretenden Störungen und Modellabweichungen ausgleichen kann, zu realisieren. Des Weiteren ist eine Steuerung nicht fähig, auftretende Änderungen im Geschwindigkeitsprofil durch Änderungen in der Verkehrssituation oder durch Umwelteinflüsse auszugleichen bzw. darauf zu reagieren. Auch Änderungen der zu fahrenden Route durch den Fahrer können von einer optimalen Steuerung nicht berücksichtigt werden. Daraus folgt, dass je länger das Fahrzeug lediglich mit einer Steuerung betrieben wird, desto größere Abweichungen vom vorgegebenen bzw. geplanten Fahrverlauf können entstehen. Diese können zu einem nicht gewollten oder sogar gefährlichen Verhalten des Fahrzeugs führen. Aus diesem Grund ist es unbedingt nötig, eine Regelung auf zu bauen, welche auf jegliche Änderungen im geplanten Fahrverlauf und äußere Einflüsse am Fahrzeug reagieren und diese ausgleichen kann. In Abschnitt 4.1 wird das Konzept der modellprädiktiven Regelung vorgestellt und im Abschnitt 4.2 wird dieses Konzept auf das vorliegende Fahrzeugmodell angewendet. Abschließend soll die modellprädiktive Regelung anhand des detaillierten Fahrzeugmodells für reale Strecken in Abschnitt 4.3 evaluiert werden.

4.1 Grundlagen der modellprädiktiven Regelung

Beim Konzept der modellprädiktiven Regelung wird zu jedem Zeitschritt die Optimierung mit dem aktuellen Fahrzeugzustand ($\hat{\Delta}$ Anfangszustand für Optimierung) neu berechnet und somit für jeden Zeitpunkt eine neue optimale Steuerung für den Betrieb des Fahrzeugs berechnet. Da es zu jedem Zeitpunkt zu einer Neuberechnung der Steuerfolge kommt, wird nur der erste berechnete Wert der optimalen Steuerfolge für den Betrieb

4.1 Grundlagen der modellprädiktiven Regelung

des Fahrzeugs verwendet. Durch diese Art der Regelung kann zu jedem Zeitpunkt auf Änderungen im vorgegebenen Fahrverlauf und auftretende Abweichungen durch das vereinfachte Fahrzeugmodell eingegangen werden. Auf die Idee der modellprädiktiven Regelung wird in [11] detailliert eingegangen, wobei das Konzept der nichtlinearen MPC-Regelung mit nichtlinearem Prozessmodell zur Anwendung kommt. Die wesentlichen Aspekte dabei sind:

- Anwendung des Prinzips des gleitenden Prädiktionshorizonts T_H
- Ermittlung einer optimalen Stellgrößenfolge für den Prädiktionshorizont, unter Berücksichtigung von Stellgrößen- und Zustandsraumbegrenzungen
- Stetige Korrektur (Neuberechnung) der Prädiktion durch neue bzw. geschätzte Zustandsgrößen als Anfangswerte der Optimierung

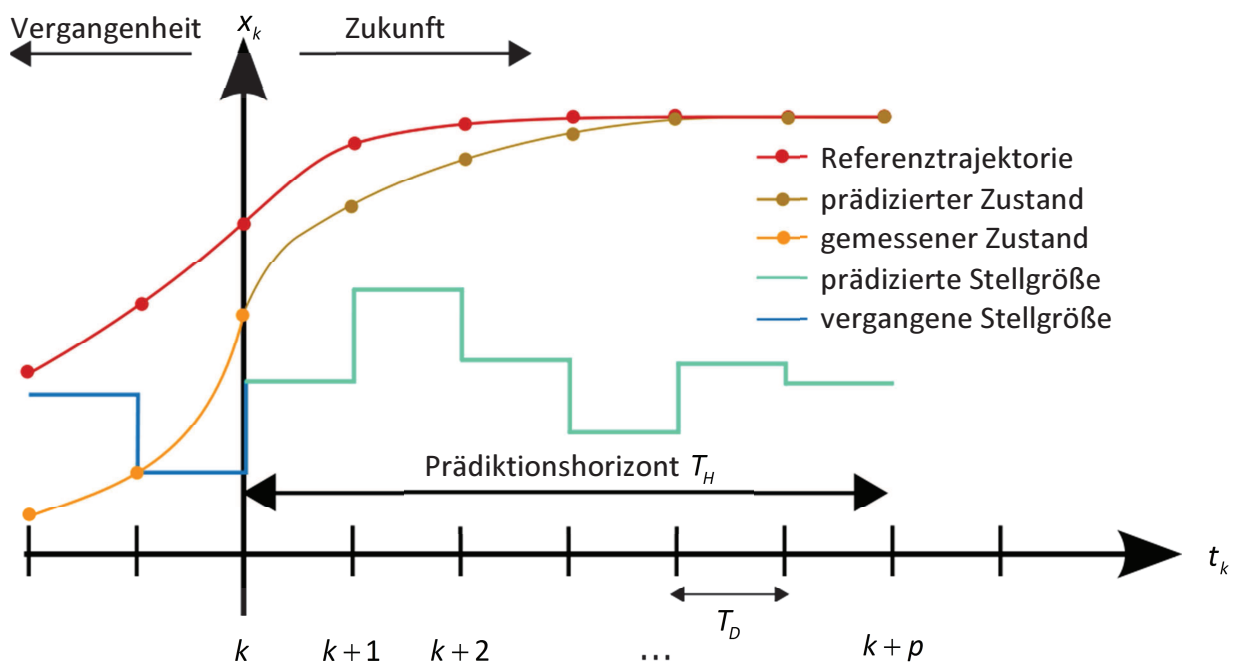


Abbildung 4.1: Prinzipielle Arbeitsweise der MPC - Regelung¹⁸

Abbildung 4.1 zeigt den zeitlichen Verlauf von Zustands- und Stellgröße eines zeitdiskreten Systems 1. Ordnung mit einer Stellgröße beispielhaft für den Zeitpunkt k

¹⁸ Überarbeitete Abbildung aus http://commons.wikimedia.org/wiki/File:MPC_scheme_basic.svg März 2013

dar. Es wird dabei eine Referenztrajektorie vorgegeben, welcher der Zustand x unter minimalen Kosten folgen soll. Im Zeitpunkt k wird dabei die aktuelle Zustandsgröße x_k gemessen oder mittels eines Beobachters bzw. Zustandsschätzers bestimmt und als Anfangszustand für die Optimierung mit der dem Prädiktionshorizont T_H vorgegeben. Durch die Optimierung erhält man eine optimale Steuerfolge

$$u_i^o \in U_i^\pi, \quad i = k, k+1, \dots, k+p \quad (4.1)$$

für den gesamten Prädiktionshorizont T_H . Aus dieser optimalen Steuerfolge u_i^o wird nun lediglich das erste Element u_k^o auf das System auf geschaltet und die Optimierung zum Zeitpunkt $k+1$ erneut durchgeführt. Durch die zu jedem Zeitpunkt erneute Optimierung mit dem tatsächlichen Anfangszustand kann so auf auftretende Störungen und Modellabweichungen reagiert werden. Diese Vorgehensweise wird als Prinzip des gleitenden Horizonts¹⁹ bezeichnet und beschreibt die typische Vorgehensweise bei MPC-Regelungen.

Bei diesem Konzept der MPC-Regelung wird jedoch idealer Weise davon ausgegangen, dass die optimale Stellgrößenfolge u_i^o bereits zum Zeitpunkt k , also gleich nach der Bestimmung des aktuellen Zustands x_k bekannt und somit berechnet ist. Da die Optimierung jedoch in der Praxis immer eine gewisse Rechenzeit T_{opt} in Anspruch nimmt, muss immer von einer Totzeit der Länge T_{opt} zwischen Bestimmung des aktuellen Zustands x_k und dem Aufschalten der entsprechenden Stellgröße u_k^o für den Zeitpunkt k , ausgegangen werden. Aus dieser Überlegung lässt sich auch die folgende Bedingung für den realistischen bzw. echtzeitfähigen Einsatz einer MPC-Regelung ableiten:

$$T_{opt} < T_D \quad (4.2)$$

Das heißt, die Lösung des Optimierungsproblems muss weniger Zeit in Anspruch nehmen als neue Messwerte bzw. Anfangszustände für die erneute Optimierung bereitgestellt werden und diskrete Werte für die Stellgröße vorgegeben werden.

¹⁹ engl. „receding horizon principle“

Je nach Anwendung kann es durch die auftretende Totzeit zu Instabilitäten oder einem ungewollten Verhalten des Systems kommen, weshalb bei dieser Art der Regelung $T_{opt} \ll T_D$ gelten sollte.

Im vorliegenden Falle des Fahrzeugmodells ist diese Bedingung nicht erfüllt. Es lässt sich jedoch dieses Problem mit den folgenden Schritten umgehen:

- (1) Einlesen des aktuellen Zustands x_k
- (2) Prädiktion bzw. Lösen der Systemgleichung $x_{k+1} = F_k(x_k, u_k^o)$ für einen Zeitschritt zur Abschätzung des Folgezustands x_{k+1}
- (3) Bestimmung der optimalen Steuerfolge

$$u_i^o, \quad i = k+1, k+2, \dots, k+p+1 \quad (4.3)$$

mit dem prädizierten Anfangszustand x_{k+1}

- (4) Speichern des ersten Werts der Steuerfolge u_{k+1}^o und zum Zeitpunkt $k+1$ bereitstellen

Das heißt, die optimale Steuerfolge wird stets im Vorhinein berechnet und zum benötigten Zeitpunkt ausgegeben. Mit dieser Art der Implementierung kann die durch die Optimierung entstehende Totzeit zwischen dem Messen des aktuellen Zustands und der Ausgabe der entsprechenden optimalen Stellgröße umgangen werden. Die Qualität dieser Lösung hängt jedoch sehr stark von der Qualität der Prädiktion des Folgezustands x_{k+1} ab, da die Optimierung eine dem Anfangszustand entsprechende Stellgrößenfolge u_i^o berechnet. Aus diesem Grund sollte für die Prädiktion des Folgezustands x_{k+1} ein möglichst genaues Lösungsverfahren gewählt werden. Die Prädiktion des Folgezustands x_{k+1} und die Rückwärts-Rechnung der dynamischen Programmierung bieten bei dieser Art der MPC-Regelung eine Möglichkeit der Parallelverarbeitung, das heißt, um Rechenzeit zu sparen können die beiden Berechnungen, unabhängig von einander parallel erfolgen.

Die Länge des Prädiktionshorizonts entspricht bei der MPC-Regelung der Länge des Optimierungsproblems und ist daher entscheidend für die Lösungsdauer der Optimierung. Wählt man den Prädiktionshorizont zu kurz, kann dies die Qualität der Regelung negativ beeinflussen und ein vorhandenes Optimierungspotential nicht vollständig ausnutzen. Wird jedoch der Prädiktionshorizont zu lange gewählt gefährdet dies die Bedingung (4.2), wobei der Nutzen weiter in der Zukunft prädizierte Zustände, durch die wachsenden Unsicherheiten, ohnehin begrenzt ist.

Die Untersuchung der Stabilität der modellprädiktiven Regelung ist im Allgemeinen mathematisch aufwändig, wobei in der Literatur verschiedene Ansätze vorgestellt wurden. In [12] und [13] werden einige dieser Ansätze beschrieben, wie zum Beispiel die Vorgabe von finalen Kosten $J_N(\cdot)$ um das System in einen vorgegeben, stabilen Bereich (z.B. eine Ruhelage) des Zustandsraums am Ende des Optimierungsintervalls zu zwingen. Bei dem vorliegenden Problem ist diese Methode jedoch nicht zweckmäßig da für die Motordrehzahl oder den zu wählenden Gang kein eingeschränkter Bereich (außer die ohnehin geltende Zustandsraumbeschränkung) am Ende des Optimierungsintervalls vorgegeben werden kann. Ein alternativer Ansatz ist die Annahme der Kostenfunktion als Lyapunov-Funktion und Nachweis der Stabilität. Besonders oft wird dabei die Methode der quadratischen Stabilität [14] verwendet, wobei eine quadratische Lyapunov-Funktion in Form der Kostenfunktion gefunden werden kann. In der Praxis wird die Stabilität des Weiteren oftmals mit Hilfe von Simulationen gezeigt und somit ein mathematischer Beweis umgangen. Im Zuge dieser Arbeit soll die Funktionalität und Stabilität der MPC-Regelung anhand von realen Streckendaten untersucht werden.

4.2 Anwendung der modellprädiktiven Regelung auf das Fahrzeugmodell

Die in Abschnitt 4.1 vorgestellte Abfolge zur Berechnung des MPC-Algorithmus muss durch den Einsatz des quasistatischen Ansatzes in der Prädiktion des Fahrzeugmodells wie folgt adaptiert werden:

- (1) Einlesen der aktuellen Motordrehzahl $\omega_{VM,k}$ und der aktuellen Fahrzeuggeschwindigkeit v_k
- (2) Prädiktion von $\omega_{VM,k+1}$ und v_{k+1} mittels der Systemgleichung (2.32) zur Abschätzung des Folgezustands unter Verwendung der zuvor berechneten Stellgrößen $\alpha_{GP,k}, F_{BR,k}$ und $g_{G,k}$.
- (3) Bestimmung der optimalen Steuerfolge für $i = k+1, k+2, \dots, k+p+1$ mit den Anfangszuständen für die Optimierung $\omega_{VM,k+1}$ und $g_{G,k+1} = g_{G,k}$ mit Berücksichtigung von v_{k+1} beim quasistatischen Ansatz als ersten Wert des vorgegebenen Geschwindigkeitsverlaufs.
- (4) Speichern der Gaspedalstellung $\alpha_{GP,k+1}$, der Bremspedalkraft $F_{BP,k+1}$ und des zu wählenden Gangs $g_{G,k+1}$ für den nächsten Zeitschritt $k+1$.

Bei der Anwendung der modellprädiktiven Regelung beschrieben in Abschnitt 4.1 auf das Fahrzeugmodell können die folgenden Größen, welche das Regelverhalten beeinflussen, definiert werden:

- (1) T_H als Länge des gleitenden Prädiktionshorizonts, welcher die Länge des zu lösenden Optimierungsproblems festlegt
- (2) T_{schalt} als Vorgabe für das Zeitfenster zwischen 2 Schaltvorgängen
- (3) T_D als Diskretisierungszeit für die Modellprädiktion und Neuberechnung der optimalen Steuerfolgen

Hinzu kommen weitere Parameter, welche zum Beispiel die Optimierungsgüte beeinflussen, wie die Anzahl der Stützstellen $N_x\{1\}$ und $N_u\{1\}$ des Zustands- und Stellgrößenraums aus Tabelle 3.3, oder Veränderungen in der Kostenfunktion durch Variation der Gewichtungsfaktoren k_1 und k_2 aus Gl.(3.29).

Durch diese Vielzahl an variablen Parametern ergibt sich eine schier unbegrenzte Anzahl an Realisierungsmöglichkeiten, welche ein sehr ähnliches Gesamtverhalten im Online-Betrieb zeigen. Nachfolgend wird lediglich auf eine Variation des Prädiktionshorizonts T_H und der Schaltzeit T_{schalt} näher eingegangen. Die Diskretisierungszeit wird in Folge mit $T_D = 0,5s$ angenommen.

Die Wahl des Prädiktionshorizonts T_H stellt jene Größe dar, welche den Rechenaufwand maßgeblich beeinflusst. Da der Prädiktionshorizont zur Laufzeit variabel gewählt werden kann, wäre es möglich wie in [15] oder [10] beschrieben, einen konstanten Prädiktionshorizont in Form einer Weglänge zu realisieren. Wird zum Beispiel ein konstanter Prädiktionshorizont mit $S_H = 1000m$ Länge gewählt, würde dies, unter der Annahme einer konstanten Fahrzeuggeschwindigkeit v_{konst} für die nächsten 1000m, zu folgenden Prädiktionshorizonten T_H führen:

$$\begin{aligned} v_{konst} = 15 \frac{km}{h} &\quad \rightarrow \quad T_H = \frac{S_H}{v_{konst}} = \frac{1000m}{\frac{15 m}{3,6 s}} = 240s \\ v_{konst} = 130 \frac{km}{h} &\quad \rightarrow \quad T_H = \frac{S_H}{v_{konst}} = \frac{1000m}{\frac{130 m}{3,6 s}} = 27,7s \end{aligned} \tag{4.4}$$

Eine Möglichkeit dem, wie aus Gl.(4.4) ersichtlich, sehr langen und damit rechentechnisch aufwändigen Prädiktionshorizont bei langsamen Fahrgeschwindigkeiten entgegen zu wirken, ist die Umstellung der Modellprädiktion von einem zeitdiskreten Fahrzeugmodell auf ein wegdiskretes Fahrzeugmodell. Aus Simulationsstudien in Kapitel 2.4 ist eine obere Schranke für den Zeitschritt T_D bekannt, für welchen ein stabiles Modellverhalten des nichtlinearen, vereinfachten Fahrzeugmodells mit Drehmomentwandler gewährleistet

werden kann. Aus dieser Überlegung lassen sich wiederum die bei gegebener konstanter Fahrzeuggeschwindigkeit v_{konst} resultierenden wegdiskreten Schritte ableiten:

$$\begin{aligned} v_{konst} = 15 \frac{km}{h} &\rightarrow S_D = v_{konst} \cdot T_{D,max} = \frac{15 \text{ m}}{3,6 \text{ s}} \cdot 0,5s = 2m \\ v_{konst} = 130 \frac{km}{h} &\rightarrow S_D = v_{konst} \cdot T_{D,max} = \frac{130 \text{ m}}{3,6 \text{ s}} \cdot 0,5s = 18m \end{aligned} \quad (4.5)$$

Aus Gl.(4.5) ist klar erkenntlich, dass eine Wegdiskretisierung von $S_D = 2m$ zu wählen ist, um ein stabiles Fahrzeugmodell zu gewährleisten. Diese Wahl würde jedoch zu einem erheblichen Rechenaufwand, pro Optimierungsschritt führen und eine echtzeitfähige Implementierung für einen Prädiktionshorizont der Weglänge $S_H = 1000m$ nicht ermöglichen.

Es ist ohnehin fraglich, ob vor allem bei sehr niedrigen Fahrzeuggeschwindigkeiten ein Prädiktionshorizont von $S_H = 1000m$ oder ähnlicher Länge in der Praxis sinnvoll ist, da dies wie aus Gl.(4.4) zu entnehmen einem Blick von 4min in die Zukunft entspricht. Die wahrscheinliche Menge an Fahrsituationen bei geringer Fahrzeuggeschwindigkeit, welche sich auf das Verhalten des Fahrers 4min in der Zukunft auswirken, kann als sehr begrenzt angenommen werden. Wählt man deshalb einen fixen zeitlichen Prädiktionshorizont von zum Beispiel $T_H = 30s$ ergibt sich ein für einen, Geschwindigkeitsbereich $15 \frac{km}{h} \leq v \leq 130 \frac{km}{h}$ der linear steigende Prädiktionshorizont in Metern:

$$S_H = v \cdot T_H, \quad (4.6)$$

wobei dieser Zusammenhang nur für eine konstante Fahrzeuggeschwindigkeit im jeweiligen Zeitfenster T_H gilt. Bei der Wahl von $T_H = 30s$ ergibt sich daraus der folgende Bereich des entsprechenden Prädiktionshorizont S_H in Metern

$$\begin{aligned} v_{konst} = 15 \frac{km}{h} &\rightarrow S_H = v_{konst} \cdot T_H = \frac{15 \text{ m}}{3,6 \text{ s}} \cdot 30s = 125m \\ v_{konst} = 130 \frac{km}{h} &\rightarrow S_H = v_{konst} \cdot T_H = \frac{130 \text{ m}}{3,6 \text{ s}} \cdot 30s = 1083m \end{aligned} \quad (4.7)$$

Aus Gl.(4.7) ergibt sich in Folge, dass abhängig vom jeweiligen Geschwindigkeitsverlauf im betrachteten Zeitintervall bei $T_H = 30s$ das für die Optimierung berücksichtigte zukünftige Wegstück zwischen $125m$ und $1083m$ liegt.

4.3 Evaluierung der Modell Prädiktiven Regelung

Aus den 3 Variationen für die Strecke Graz-Wolfsberg 1 aus Tabelle 4.1 bestätigt sich die Vermutung, dass eine kleinere Schaltzeit T_{schalt} bei gleichem Prädiktionshorizont zu einem geringeren Verbrauch führt. Es wurden 3 verschiedene Schaltzeiten zwischen $1s$ und $5s$ gewählt und die MPC-Regelung für die gesamte Strecke Graz – Wolfsberg 1 jeweils erneut mit verschiedenen Prädiktionshorizonten berechnet. Die kürzeste Schaltzeit $T_{schalt} = 1s$ führt zum erwarteten, geringsten Verbrauch, da diese Realisierung dem Optimierungsverfahren die meisten Freiheiten in Form von Schaltvorgängen erlaubt und somit die beste Optimierungsgüte, in Form von minimalem Verbrauch, liefert. Da in der Praxis, zwar Schaltvorgänge zu jeder Sekunde möglich wären, sich jedoch negativ auf den Fahrkomfort auswirken, wurde in weiterer Folge eine Schaltzeit von $T_{schalt} = 3s$ gewählt. Mit dieser gewählten Schaltzeit wurden die Auswirkungen der Länge des Prädiktionshorizonts T_H auf den Treibstoffverbrauch untersucht.

4.3 Evaluierung der Modell Prädiktiven Regelung

Route	Fahrzeit	Fahrstrecke	T_H	T_{schalt}	Rechenzeit	Verbrauch	Reduktion
-	min	km	s	s	min	l	%
Graz – Wolfsberg 1	29,8	32,7	10	1	19,7	1,3671	0
			10	3	19,8	1,3702	-0,23
			10	5	19,5	1,3746	-0,54
Graz – Wolfsberg 1	29,8	32,7	30	1	55,2	1,3645	0
			30	3	50,3	1,3697	-0,38
			30	5	52,1	1,3746	-0,74
Graz – Wolfsberg 1	29,8	32,7	60	1	96,4	1,3645	0
			60	3	100,2	1,3697	-0,38
			60	5	98,2	1,3746	-0,74
Graz – Wolfsberg 2	25,35	27,5	10	3	17,9	1,4255	-0,14
			30	3	41,4	1,4235	0
			60	3	82,1	1,4233	0,01
Budapest 1	34,9	18,1	10	3	23,7	1,6516	-0,03
			30	3	59,5	1,651	0
			60	3	109,8	1,651	0
Budapest 2	29,1	20,8	10	3	19,2	1,0767	-0,074
			30	3	51,7	1,0759	0
			60	3	94,1	1,0759	0
Budapest 3	36,9	17,0	10	3	25,3	1,1781	-0,17
			30	3	44,7	1,176	0
			60	3	116,2	1,176	0

Tabelle 4.1: Vergleich der Eingabeparameter für verschiedene Routen

Wird der Prädiktionshorizont $T_H > 30s$ gewählt, ergibt sich wie aus Tabelle 4.1 ersichtlich bei keiner der simulierten Strecken eine weitere Verbrauchsreduktion durch einen längeren Prädiktionshorizont. Generell lässt sich feststellen, dass Veränderungen im Prädiktionshorizont beim vorliegenden Problem nur geringe (<0,2%)

Verbrauchseinsparungen zur Folge haben. Ein zu kurzer Prädiktionshorizont kann jedoch zu Stabilitätsproblemen [11] führen und sollte daher mit Bedacht gewählt werden. Aus diesem Grund wird in weiterer Folge ein Prädiktionshorizont von $T_H = 30s$ gewählt.

Abbildung 4.3 und Abbildung 4.5 zeigen jeweils die Geschwindigkeits- und Höhenverläufe der Strecke Graz – Wolfsberg 2 und Budapest 3. Bei beiden Strecken zeigt sich ein sehr gutes Regelverhalten mit maximalen Abweichungen im Geschwindigkeitsverlauf vom vorgegeben Verlauf von 2-3km/h. Die von der modellprädiktiven Regelung vorgegebenen optimalen Stellgrößen für die gesamten Streckenverläufe sind in Abbildung 4.4 und Abbildung 4.6 dargestellt.

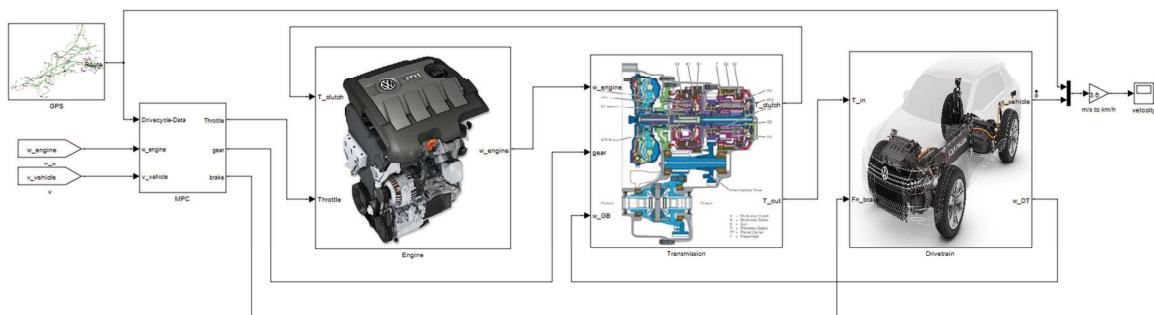


Abbildung 4.2: MPC-Aufbau in Simulink mit dem detaillierten Fahrzeugmodell

Abbildung 4.2 zeigt den prinzipiellen Aufbau der MPC-Regelung in Simulink mit dem detaillierten Fahrzeugmodell. Um eine möglichst effiziente Anbindung an Simulink zu ermöglichen wurde der MPC-Algorithmus in Form einer MATLAB® s-Funktion²⁰ implementiert. Näheres kann aus dem Anhang entnommen werden.

²⁰ <http://www.mathworks.de/de/help/simulink/slref/sfunction.html> März 2013

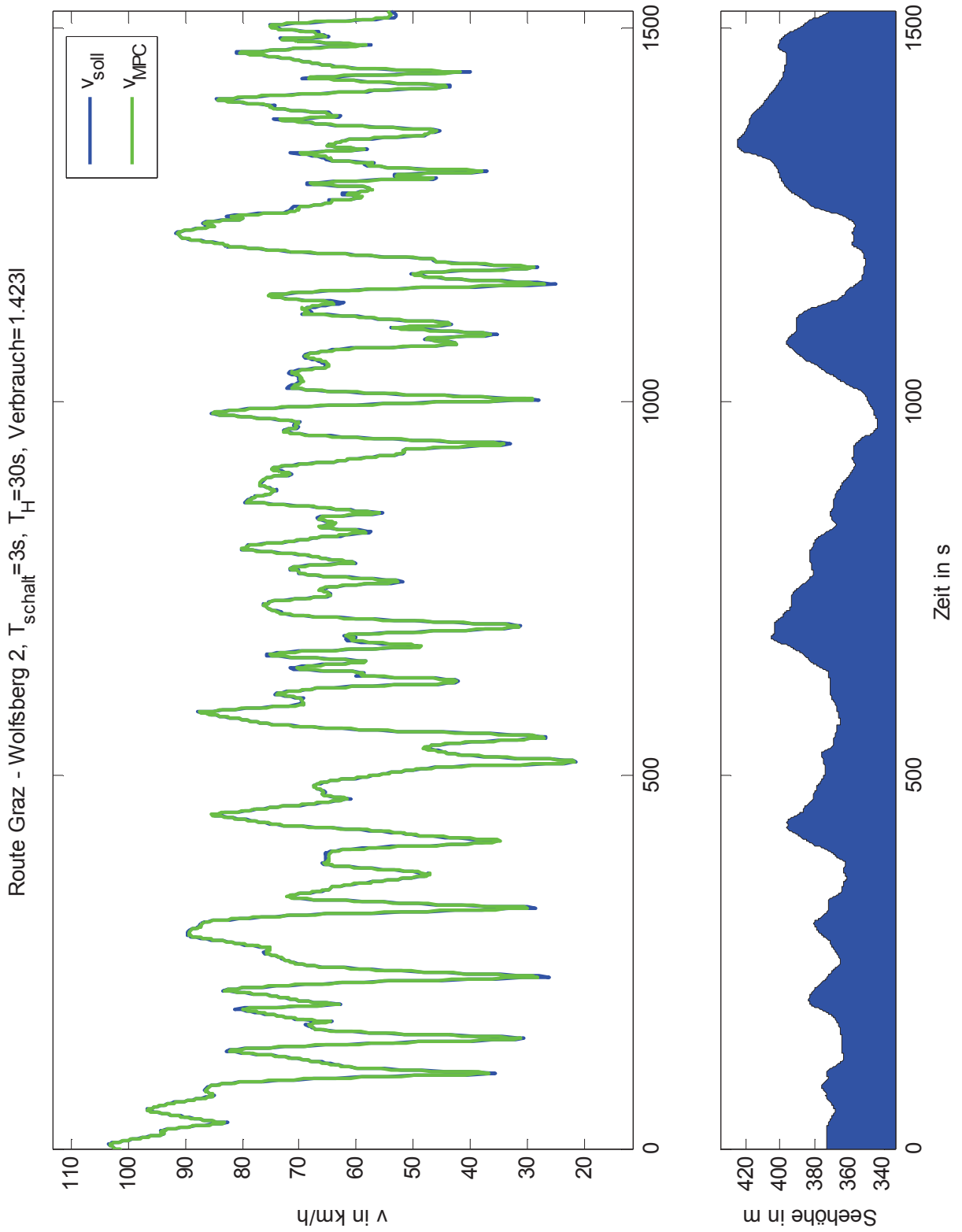


Abbildung 4.3: Geschwindigkeitsverlauf mit MPC-Regelung, Route Graz – Wolfsberg 2

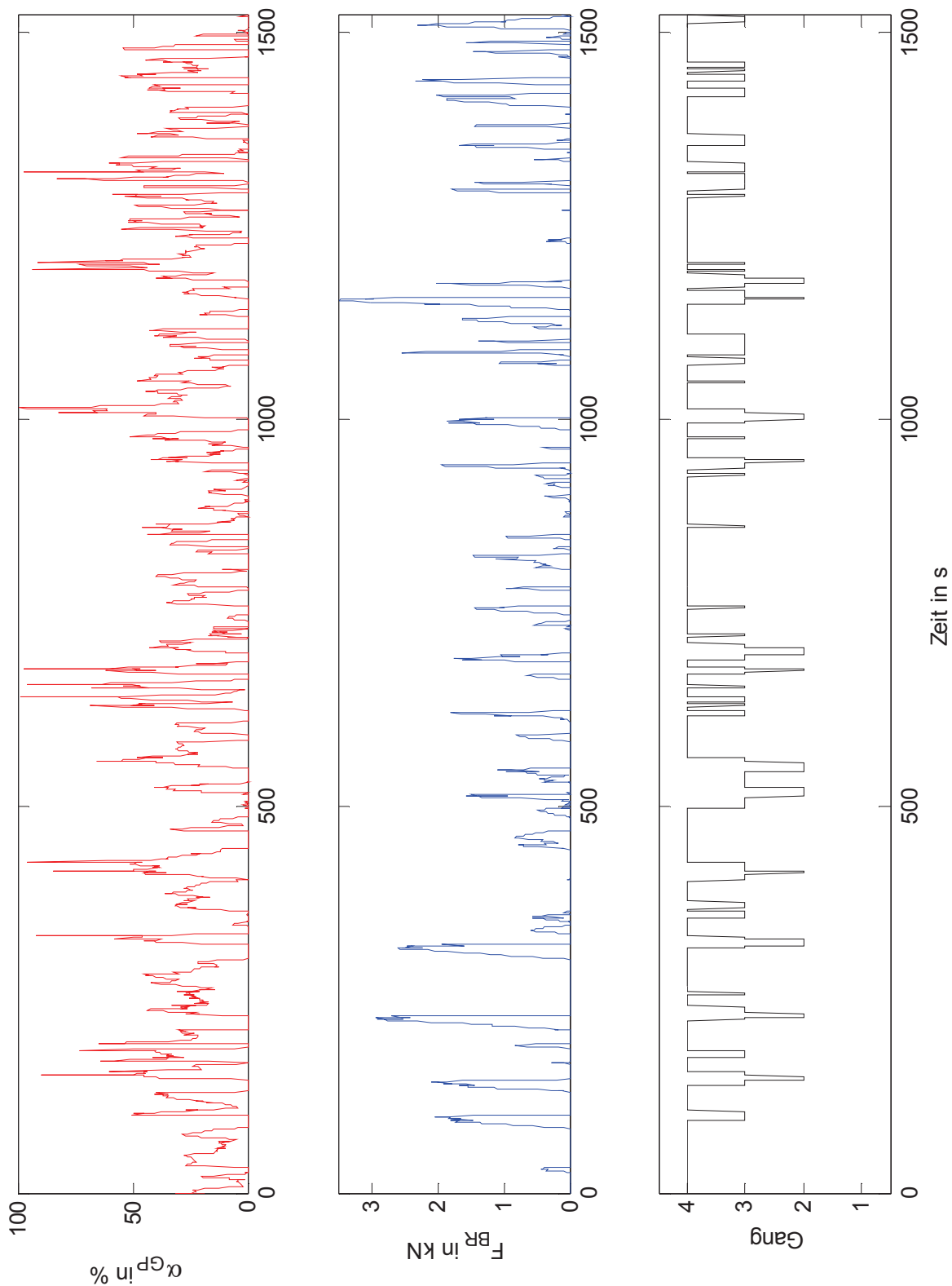


Abbildung 4.4: Berechnete optimale Stellgrößen der MPC-Regelung, Route Graz – Wolfsberg 2

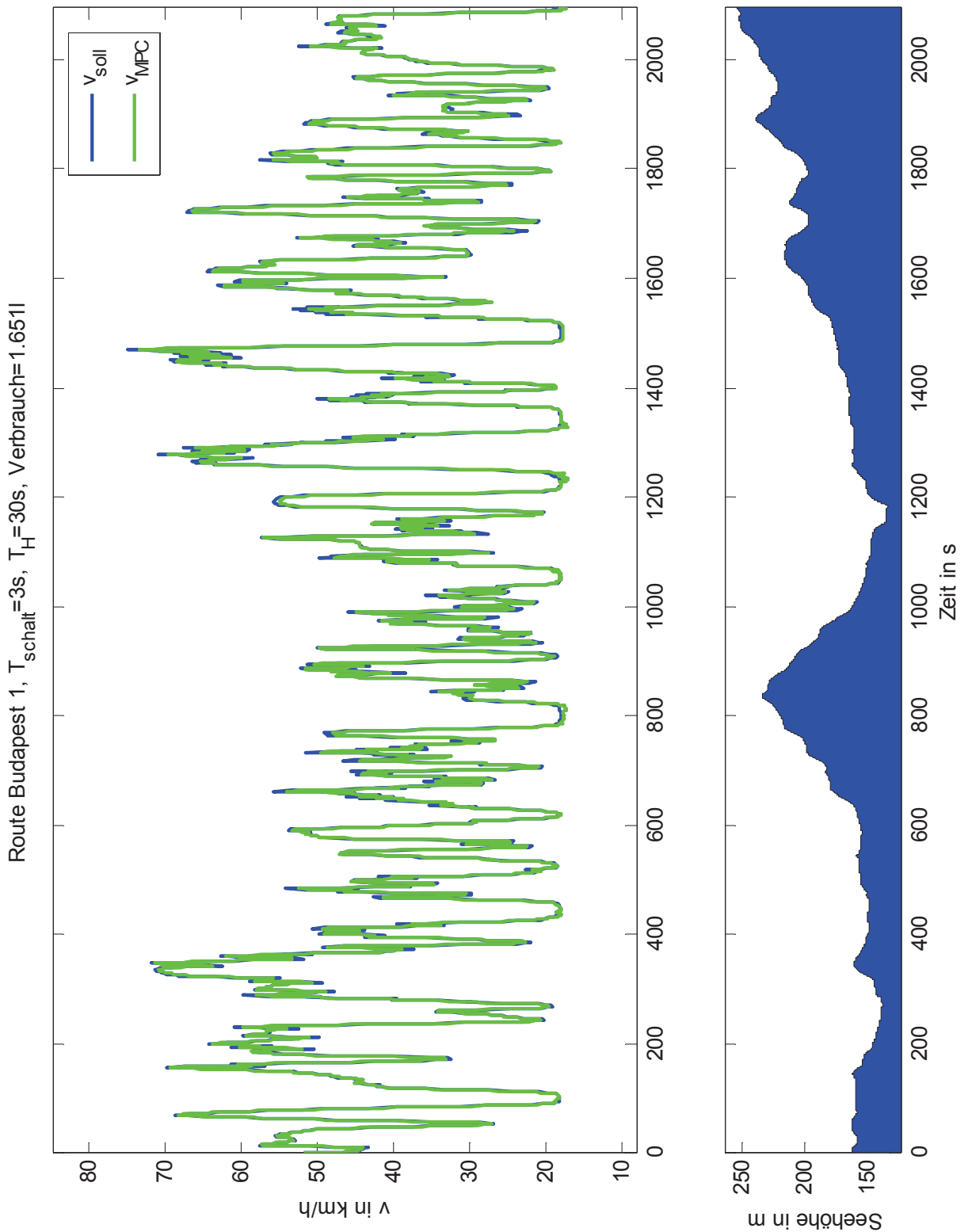


Abbildung 4.5: Geschwindigkeitsverlauf mit MPC-Regelung, Route Budapest 1

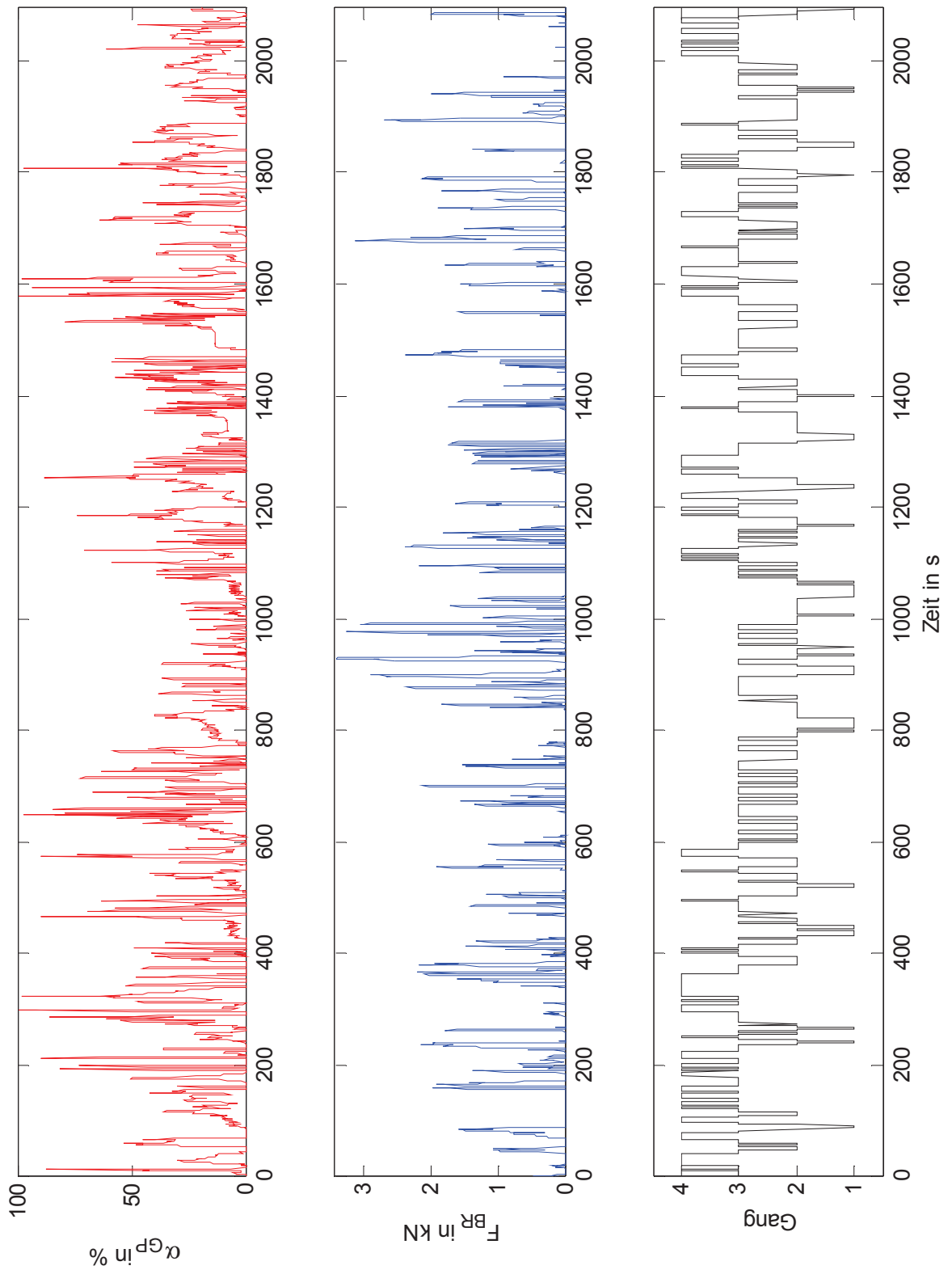


Abbildung 4.6: Berechnete optimale Stellgrößen der MPC-Regelung, Route Budapest 1

5 Vergleich modellprädiktive Regelung - PI-Regelung

Um die Qualität der in Kapitel 4 beschriebenen modellprädiktiven Regelung bewerten zu können, ist es sinnvoll, die Simulation mit realen Messdaten oder anderen Regelungsstrategien zu vergleichen. Da für den in Kapitel 2 beschriebenen Fahrzeugtyp und die aufgenommenen Teststrecken keine realen Vergleichsdaten existieren, wird eine alternative Regelungsstrategie als Vergleich herangezogen. Für den in Kapitel 2 beschriebenen Fahrzeugtyp existiert bereits eine auf den Konzepten der klassischen Regelungstechnik beruhende PI-Regelung nach [16] für die Gaspedalstellung und die Bremspedalkraft. Der Wechsel eines Gangs führt in der Regelung zu einem Wechsel der Reglerparameter des Gaspedalreglers. Der PI-Regler für die Bremspedalkraft bleibt im gesamten Betriebsbereich gleich und darf nur aktiv werden, wenn der Gaspedalregler keine Betätigung des Gaspedals fordert. Es gelten dabei die gleichen Begrenzungen wie in Gl.(2.1) für die Gaspedalstellung. Die Wahl der Reglerparameter der eingesetzten PI-Regler erfolgt durch die Linearisierung des Fahrzeugmodells in bestimmten Betriebspunkten, wobei der Drehmomentwandler aus Kapitel 2.1.2 mit einer linearen Übersetzung von $i_{DW} = 1$ angenommen wird, wie in [16] näher beschrieben.

Das automatische Wechseln des Ganges erfolgt ebenfalls unter der Vernachlässigung des Drehmomentwandlers. Durch Gl.(2.3) bzw. das entsprechende Kennfeld ist der Verbrauch des Motors bei gegebener Motordrehzahl ω_{VM} und gegebenem Motormoment M_{VM} eindeutig bestimmt. Änderungen in der Übersetzung $i_G(g)$ durch den Wechsel eines Gangs wirken sich auf beide Größen aus und resultieren daher in einem geänderten Kraftstoffverbrauch. Angepasst an die Fahrzeuggeschwindigkeit ergibt sich daher eine begrenzte Zahl an zulässigen Gängen. Der mit dem geringsten Verbrauch bildet dabei zu jedem Zeitpunkt den zu favorisierenden Gang. Es erfolgen bei dieser Implementierung jedoch keine Abschätzungen über den zukünftigen Fahrverlauf oder die Sinnhaftigkeit des

zu wählenden Gangs in den folgenden Zeitschritten. Unterschreitet die Motordrehzahl einen vorgegebenen Wert, so wird auf die Wahl des Gangs mit dem geringsten Verbrauch verzichtet und in einen niedrigeren Gang geschaltet, um ein Absterben des Motors zu verhindern.

Des Weiteren wurde eine Schaltlogik implementiert, welche die Schaltfolge zeitlich begrenzt und ein kontrolliertes Wechseln von einem Gang in den Nächsten sicherstellt. Ähnlich wie bei der in Kapitel 4 beschriebenen Regelung kann ein Zeitfenster T_{schalt} , in dem nicht geschaltet werden darf, vorgegeben werden. Zum Vergleich der beiden Regelungen soll dieses Zeitfenster stets gleich groß gewählt werden, um beiden Regelungen die gleichen Möglichkeiten in Form von Änderungen im Schaltverhalten zu geben.

Die beiden Regelungskonzepte der PI-Regelung und der modellprädiktiven Regelung sollen Anhand derselben Routen verglichen werden und vor allem auf die Kraftstoffersparnis ΔV durch die Modellprädiktion eingegangen werden. Bezeichnet man den Kraftstoffverbrauch durch die PI-Regelung als V_{PI} und den Kraftstoffverbrauch durch die MPC-Regelung als V_{MPC} , ergibt sich die folgende Kraftstoffersparnis ΔV in %:

$$\Delta V = \frac{V_{PI} - V_{MPC}}{V_{MPC}} \cdot 100\% \quad (5.1)$$

Route	Fahrzeit	Fahrstrecke	V_{MPC}	V_{PI}	ΔV
-	min	km	l	l	%
Graz – Wolfsberg 1	29,8	32,7	1,3697	1,3987	2,11
Graz – Wolfsberg 2	25,35	27,5	1,4235	1,4935	4,92
Budapest 1	34,9	18,1	1,6510	1,7524	6,14
Budapest 2	29,1	20,8	1,0759	1,1449	6,41
Budapest 3	36,9	17,0	1,1760	1,2212	3,84

Tabelle 5.1: Vergleich der Regelungskonzepte in Bezug auf den Kraftstoffverbrauch

Die in Tabelle 5.1 gezeigten Simulationsergebnisse wurden mit den folgenden Einstellungen der MPC-Regelung durchgeführt:

- Zeitliche Schrittweite $T_D = 0,5s$
- Prädiktionshorizont $T_H = 30s$
- Zeitliches Schaltfenster $T_{schalt} = 3s$

Für beide Regelungen gelten die gleichen Anfangszustände für die Motordrehzahl $\omega_{VM,0}$, den eingestellten Gang g_0 und die Fahrzeuggeschwindigkeit v_0 .

Bei allen durchgeführten Simulationen zeigt sich eine Kraftstoffersparnis ΔV durch die Modellprädiktion im Bereich von 2,1% bis 6,4%. Vor allem im Bereich von Steigungsänderungen im Straßenverlauf lässt sich durch die modellprädiktive Regelung ein erhöhtes Einsparungspotential feststellen. Abbildung 5.1 zeigt einen Teilausschnitt der Strecke Graz – Wolfsberg 1 mit einer Steigung von 3,2% für 1,8km und einem anschließenden Gefälle von 2,9% für 1,9km. In Abbildung 5.2 sind die von beiden Regelungen ausgegebenen Stellgrößen für die Teilstrecke dargestellt. Beide

Regelstrategien erreichen ein gleichwertiges Verhalten im betrachteten Teilstück mit einer maximalen Abweichung vom vorgegebenen Geschwindigkeitsverlauf von 2km/h. Der PI-Regler für die Gaspedalstellung (dargestellt in Abbildung 5.2) erreicht 3 mal im Zuge des Fahrmanövers die maximale Gaspedalstellung von 100%, die MPC-Regelung hingegen bleibt zu jedem Zeitpunkt innerhalb der vorgegebenen Stellgrößenbeschränkungen für die Gaspedalstellung. Des Weiteren zeigt sich bei der Wahl des verwendeten Gangs bei beiden Regelungen in mehreren Bereichen ein unterschiedliches Verhalten. Durch die Modellprädiktion und die Berücksichtigung des zukünftigen Fahrverlaufs wechselt bzw. verbleibt die modellprädiktive Regelung stets in einen niedrigeren Gang, um zu den benötigten Zeitpunkten durch eine niedrigere Übersetzung das benötigte Moment bei einer geringeren Gaspedalstellung zu liefern. Dadurch ergibt sich stets ein möglichst effizienter Betrieb der Verbrennungskraftmaschine und somit eine Verbrauchsreduktion gegenüber der PI-Regelung im betrachteten Teilstück von 10,9%. Beim Verlauf der Bremskraft zeigt sich ebenfalls ein stark unterschiedliches Verhalten. Die MPC-Regelung erzeugt stets eine kontinuierliche Bremskraft, die PI-Regelung hingegen erzeugt ein eher gepulstes Bremsverhalten, welches vor allem am großen P-Anteil des Bremskraftreglers liegen dürfte. Ein gepulstes Bremsen verschlechtert den Fahrkomfort erheblich und sollte verhindert werden.

In Abbildung 5.3 ist der Geschwindigkeits- und Höhenverlauf eines Teilstücks der Strecke Budapest 2 dargestellt. Erneut erreichen beide Regelungen ein annähernd gleichwertiges Regelverhalten und folgen dem vorgegebenen Geschwindigkeitsprofil. Im betrachteten Teilstück wird ein Gefälle von 2,6% befahren, wobei sich eine Verbrauchsreduktion von 17% gegenüber der PI-Regelung ergibt. Es ergeben sich wiederum starke Unterschiede im Schaltverhalten der beiden Regelungen, wobei vor allem die PI-Regelung sehr häufig Schaltvorgänge vornimmt.

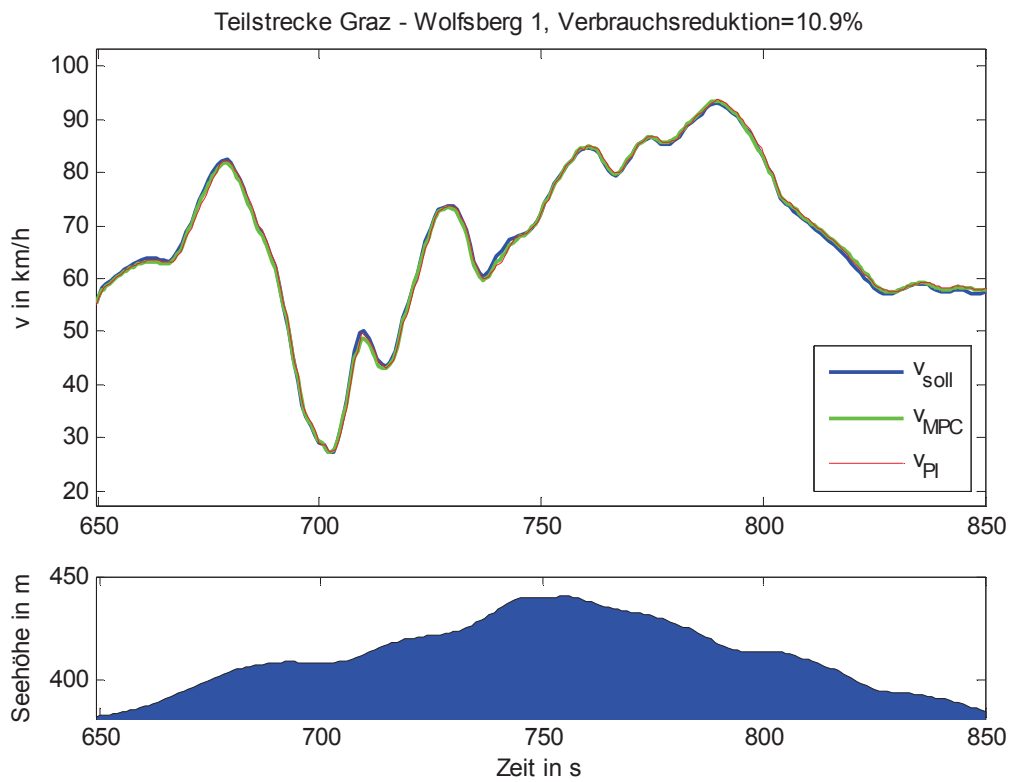


Abbildung 5.1: Vergleich der Regelungskonzepte anhand einer Teilstrecke Graz – Wolfsberg 1

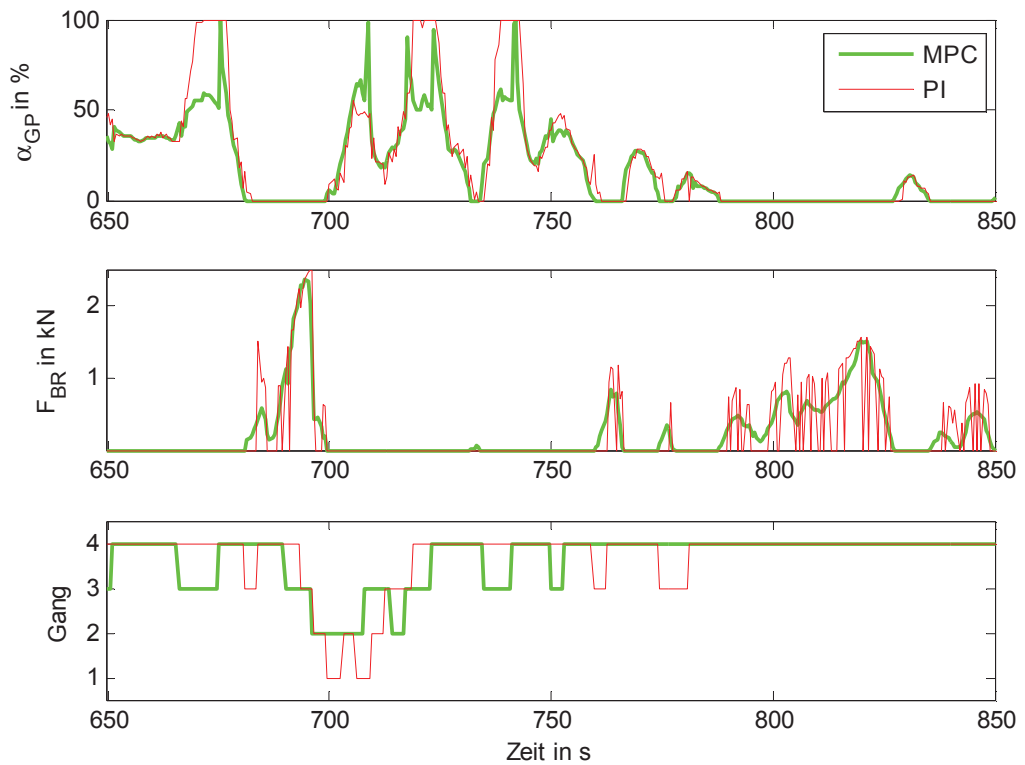


Abbildung 5.2: Stellgrößen der beiden Regelungen für das betrachtete Teilstück

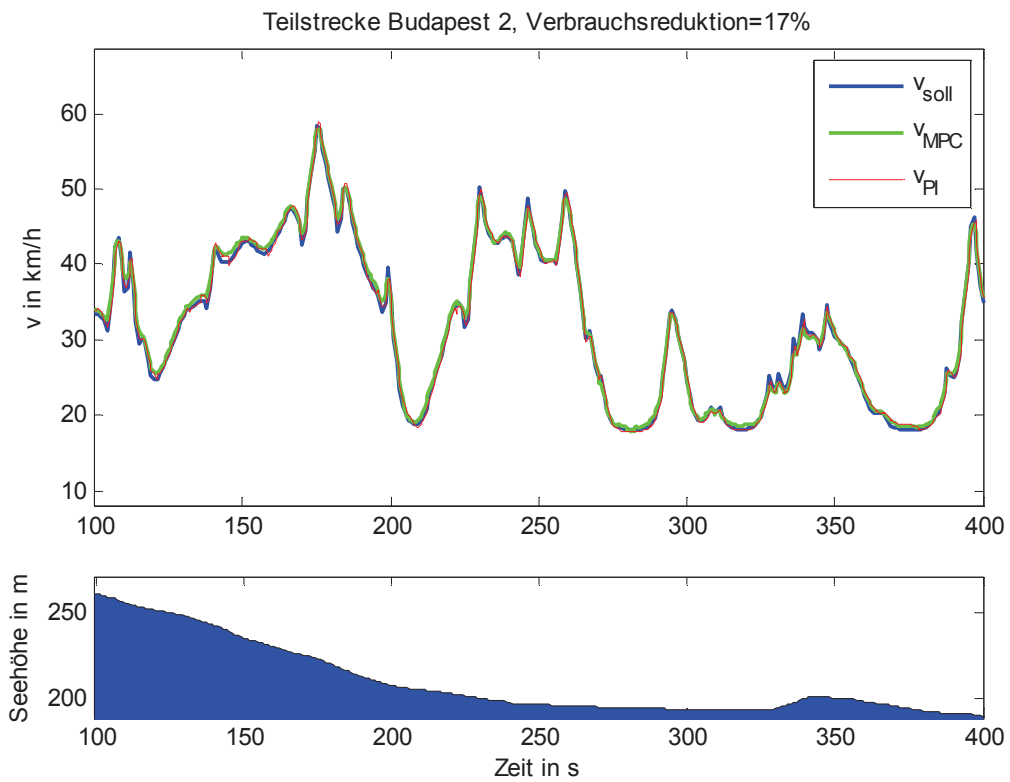


Abbildung 5.3: Vergleich der Regelungskonzepte anhand einer Teilstrecke der Route Budapest 2

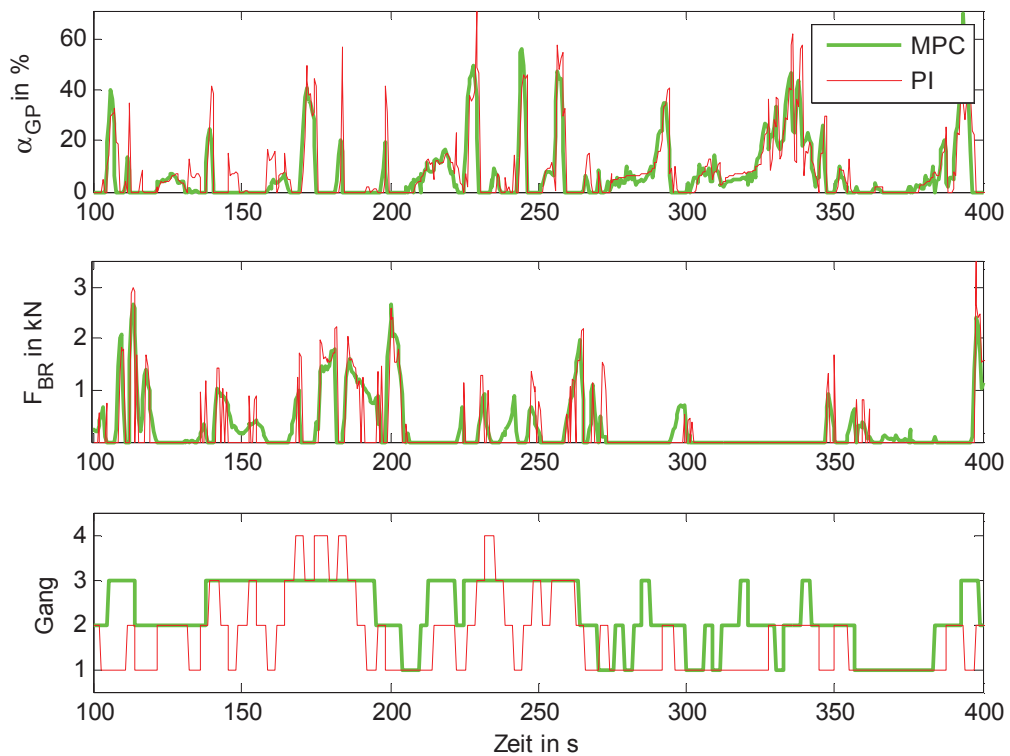


Abbildung 5.4: Stellgrößen der beiden Regelungen für das betrachtete Teilstück

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde mit Hilfe der modellprädiktiven Regelung, eine verbrauchsoptimale Regelung zum Betrieb eines PKWs realisiert. Diese Beruht auf der Methode der dynamischen Programmierung als Optimierungsverfahren und erlaubt eine einfache Implementierung und robuste Prädiktion.

Eine Besonderheit des behandelten Fahrzeugtyps liegt im Drehmomentwandler als Teil der Getriebeeinheit, wodurch die Vernachlässigungen aus [10] nicht zulässig sind und sich der Rechenaufwand der Modellprädiktion, wie in [3] beschrieben, erheblich erhöht. Aus diesem Grund ist ein Einsatz im realen Fahrbetrieb, also in einem Echtzeitsystem, wahrscheinlich nur mit einer effizienten Implementierung und ausreichender Rechenleistung möglich. Unter der Voraussetzung des Vorhandenseins von entsprechendem Kartenmaterial und der Einbindung von lokalen Verkehrsdaten in die Streckenplanung kann somit mit Hilfe der vorgestellten modellprädiktiven Regelung das Fahrzeug selbstständig verbrauchsoptimal Beschleunigungs- und Verzögerungsvorgänge einleiten. Zur Evaluierung der Robustheit der Regelung wurde ein detailliertes Fahrzeugmodell, welches auf realen Messdaten eines VW Golf 6 basiert, verwendet. Dabei wurde stets ein stabiles Verhalten bei Verwendung des MPC-Reglers beobachtet.

Die Verbrauchseinsparungen die sich im Vergleich zu einem klassischen Regelungskonzept in Form einer PI-Regelung ergeben, liegen bei ca. 2-6%.

In Zukunft könnte, um den Verbrauch weiter zu reduzieren die Effizienz der Implementierung weiter gesteigert werden und die Regelung um den Leerlauf im Getriebe erweitert werden. Alternativ könnte der Algorithmus selbstständig Änderungen am Fahrverlauf vornehmen. Damit erhöht sich jedoch der nötige Rechenbedarf durch die zusätzliche Geschwindigkeitsgröße nochmals erheblich. Dies Bedarf noch weiteren Untersuchungen und Analysen.

Tabellenverzeichnis

Tabelle 3.1: Eingabeparameter der MATLAB®-Funktion dpm.m	41
Tabelle 3.2: Vorgegebene Parameter zur Lösung des Optimierungsproblems	49
Tabelle 3.3: Simulationsergebnisse für verschiedene Einstellungen der dynamischen Programmierung.....	51
Tabelle 4.1: Vergleich der Eingabeparameter für verschiedene Routen	64
Tabelle 5.1: Vergleich der Regelungskonzepte in Bezug auf den Kraftstoffverbrauch.....	72

Abbildungsverzeichnis

Abbildung 1.1: Prinzipieller Aufbau der geforderten Regelung	2
Abbildung 2.1: Aufbau eines Verbrennungskraftfahrzeugs	5
Abbildung 2.2: Schematische Darstellung des Motorblocks	5
Abbildung 2.3: Kennfeld für das Motormoment der verwendeten VKM	6
Abbildung 2.4: Kraftstoffverbrauch für gesamten Betriebsbereich	8
Abbildung 2.5: Schematische Darstellung des hydrodynamischen Wandlers	8
Abbildung 2.6: Hauptkomponenten eines hydrodynamischen Wandlers	9
Abbildung 2.7: Kennlinien des hydrodynamischen Wandlers	11
Abbildung 2.8: Schematische Darstellung des Schaltgetriebes	12
Abbildung 2.9: Umfangskraftdiagramm des verwendeten Reifens	14
Abbildung 2.10: Anwendung des quasistatischen Ansatzes auf das Fahrzeugmodell	20
Abbildung 2.11: Modellvalidierung anhand des EUDC-Fahrzyklus	23
Abbildung 2.12: Modellvalidierung anhand des FTP-Fahrzyklus	24
Abbildung 2.13: Route Graz- Wolfsberg 1 in Google Maps	26
Abbildung 2.14: Geschwindigkeits- und Höhenprofil der Route Graz – Wolfsberg 1	27
Abbildung 2.15: Route Budapest 1 in Google Maps	28
Abbildung 2.16: Geschwindigkeits- und Höhenprofil der Route Budapest 1	29
Abbildung 3.1: Rückwärts-Rechnung bei freiem Endpunkt.	32
Abbildung 3.2: Geschwindigkeitsverlauf für optimale Steuerung der Route Graz– Wolfsberg 1	53
Abbildung 3.3: Berechnete optimale Steuerfolgen der 3 Stellgrößen, Route Graz– Wolfsberg 1	53
Abbildung 3.4: Geschwindigkeitsverlauf für optimale Steuerung der Route Budapest 3 ..	54
Abbildung 3.5: Berechnete optimale Steuerfolgen der 3 Stellgrößen, Route Budapest 3 .	54
Abbildung 4.1: Prinzipielle Arbeitsweise der MPC - Regelung	56
Abbildung 4.2: MPC-Aufbau in Simulink mit dem detaillierten Fahrzeugmodell	65

Abbildung 4.3: Geschwindigkeitsverlauf mit MPC-Regelung, Route Graz – Wolfsberg 2..	66
Abbildung 4.4: Berechnete optimale Stellgrößen der MPC-Regelung, Route Graz – Wolfsberg 2	67
Abbildung 4.5: Geschwindigkeitsverlauf mit MPC-Regelung, Route Budapest 1	68
Abbildung 4.6: Berechnete optimale Stellgrößen der MPC-Regelung, Route Budapest 1.	69
Abbildung 5.1: Vergleich der Regelungskonzepte anhand einer Teilstrecke Graz – Wolfsberg 1	74
Abbildung 5.2: Stellgrößen der beiden Regelungen für das betrachtete Teilstück	74
Abbildung 5.3: Vergleich der Regelungskonzepte anhand einer Teilstrecke der Route Budapest 2	75
Abbildung 5.4: Stellgrößen der beiden Regelungen für das betrachtete Teilstück	75

Anhang

A.MATLAB® MPC – S-Funktion

```

function [sys,x0,str,ts,simStateCompliance] =
sfunMPC_t(t,x,u,flag,DC_t,DC_v,DC_incl,Ts,Th,T_shift,x0_MPC)
% [sys,x0,str,ts,simStateCompliance] =
%     sfunMPC_t(t,x,u,flag,DC_t,DC_v,DC_incl,Ts,Th,T_shift,x0_MPC)
%sfunMPC_t S-function for a user-defined MPC - algorithm
%
% Inputs:      t           time
%             x           s_function state
%             [t;w_eng;v;gear;alpha;F_brake]' input values [w_eng;v]
%             u           input values [w_eng;v]
%             [CD_t,DC_v,DC_incl] DC data for receding horizon
%             Ts          sample time
%             Th          prediction horizon
%             T_shift     shift time
%             x0_MPC      initial state of MPC
%
% for further information on S-functions:
% http://www.mathworks.de/de/help/simulink/slref/sfunction.html

switch flag,

    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes(Ts,x0_MPC);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Update %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    case 2,
        sys = mdlUpdate(t,x,u,DC_t,DC_v,DC_incl,Ts,Th,T_shift);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Output %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    case 3,
        sys = mdlOutputs(t,x,u);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%
    case 9,
        sys = [];

```

```

    otherwise
        DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
    end

end

%
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-
function.
%=====
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes(Ts,x0_MPC)

sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 6;
sizes.NumOutputs = 4;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

x0 = x0_MPC;
str = [];
ts = [Ts 0]; % Sample period

% specify that the simState for this s-function is same as the default
simStateCompliance = 'DefaultSimState';

end

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
requirements.
%=====
%
function sys = mdlUpdate(t0,x,u,DC_t,DC_v,DC_incl,Ts,Th,T_shift)
% tic;
% read inputs - update state vector
% disp('updated state:');
x(1) = t0;
% t0
% disp('next timestep');
% prediction error
e = [t0, (x(2)-u(1))*30/pi, (x(3)-u(2)).*3.6];

x(2) = u(1);
x(3) = u(2);

% state prediction for next time step
% disp('predicted state:');

```

```

[tz,z] =
ode23(@(t,z)f_vehicle(t,z,(x(4)+1),x(5),x(6),DC_t,DC_incl),[0,Ts],[x(2);x
(3)]);

% state update with predicted states
x(1) = x(1) + Ts;
x(2) = z(end,1);
x(3) = z(end,2);

%-----
% DP - settings:
%-----
model = 'MPC_vehicle';
par = [];
res.a0 = 146.488132;
res.a1 = 0.255831;
res.a2 = 0.029651;
wheel.r = 0.3;
wheel.J = 4.3;
DT.J = 0.27;
vehicle.m = 1500;
par.Nmin = 600*pi/30;
par.Nmax = 4800*pi/30;

% Create desired speed vector
Tstart = x(1);
Tend = x(1)+Th;
t = Tstart:Ts:Tend;

Drivecycle_v = [x(3),interp1(DC_t,DC_v,t+Ts,'linear','extrap')];
Drivecycle_dv = diff(Drivecycle_v)./Ts;
Drivecycle_v = Drivecycle_v(1:end-1);
Drivecycle_incl = interp1(DC_t,DC_incl,t,'linear','extrap');
shift_enable = (mod(t,T_shift)==0);
% initial values for DP - algorithm
grd.X0{1} = x(2);
grd.X0{2} = x(4);

% Calculation of required torque:
% Torque by resistances(roll, air,...)
T_res = wheel.r.*(res.a0 + res.a1.*Drivecycle_v +
res.a2.*Drivecycle_v.^2);
% Torque by road inclination
T_incl = wheel.r.*vehicle.m.*9.81.*sin(Drivecycle_incl);
% Torque by vehicle inertia
T_inert = (wheel.J + DT.J +
vehicle.m.*wheel.r.^2)./wheel.r.*Drivecycle_dv;

T_req = T_res + T_incl + T_inert;

% boundaries of u1
u1_ub = T_req./8+40;ind = find(u1_ub>100);
u1_lb = T_req./8-40;
u1_lb(ind) = u1_lb(ind) - (u1_ub(ind) - 100);
u1_ub(ind) = 100;

% DP - problem and options structure:
prb.Ts = Ts;

```

```

prb.N = length(Drivecycle_v)-1;
prb.N0 = 1;
prb.W{1} = Drivecycle_v;
prb.W{2} = Drivecycle_dv;
prb.W{3} = T_req;
prb.W{4} = shift_enable;

options = dpm();

% DP - state and input grid:
% X{1} grid structure for engine speed
grd.Xn{1}.lo = (800*pi/30)-200*pi/30.*(x(2)>600*pi/30).*(x(2)<800*pi/30);
grd.Xn{1}.hi = grd.Xn{1}.lo + (4000*pi/30);
grd.XN{1}.lo = 800*pi/30;
grd.XN{1}.hi = 4800*pi/30;
grd.Nx{1} = 21;

% X{2} grid structure for gears
grd.Xn{2}.lo = 0;
grd.Xn{2}.hi = 3;
grd.XN{2}.lo = 0;
grd.XN{2}.hi = 3;
grd.Nx{2} = 4;

% U{1} grid structure for u1
grd.Un{1}.hi = u1_ub(1:end-1);
grd.Un{1}.lo = u1_lb(1:end-1);
grd.Nu{1} = 17;

% U{2} grid structure for u2
grd.Un{2}.hi = 1;
grd.Un{2}.lo = -1;
grd.Nu{2} = 3;

options.MyInf = 10000;
options.InputType = 'cd';
options.BoundaryMethod = 'none';
options.Waitbar = 'off';
options.Verbose = 'off';

%-----
% BASIC DP:
%-----
global X2 U1 Fn_brake Tsim
[out dyn] = dpm(model,par,grd,prb,options);

% disp('new state');

% update state for new optimal output
if isnan(out.X2(1))
    % optimization not successfull
    x(4) = X2(2);
    x(5) = U1(2).*(U1(2)>=0);
    x(6) = -Fn_brake(2);
    X2 = X2(2:end);
    U1 = U1(2:end);
    Fn_brake = Fn_brake(2:end);

```

```
fprintf('%s','MPC running:  %%');

% disp('Optimization not successful, using stored output from previous
optimization');
else
    % optimization successfull
    x(4) = out.X2(1);
    x(5) = out.U{1}(1) .* (out.U{1}(1) >= 0);
    x(6) = -out.Fn_brake(1);
    X2 = out.X2;
    U1 = out.U{1};
    Fn_brake = out.Fn_brake;
end
% update progress in statusbar
fprintf('%s%2d  %%',ones(1,4)*8,round(100*t0/Tsim));
sys = x;
end

%
%=====
% mdlOutputs
% Return the output vector for the S-function
%=====
%
function sys = mdlOutputs(t,x,u)
sys = [x(5); (x(4)+1);x(6);x(1)];

end
```

Literaturverzeichnis

- [1] G. Lechner und H. Naunheimer, Fahrzeuggetriebe - Grundlagen, Auswahl, Auslegung und Konstruktion, Stuttgart: Springer-Verlag Berlin Heidelberg New York, 1994.
- [2] M. Mitschke und H. Wallentowitz, Dynamik der Kraftfahrzeuge, Braunschweig, Deutschland: Springer-Verlag Berlin Heidelberg New York, 2004.
- [3] L. Guzzella und A. Sciarretta, Vehicle Propulsion Systems - Introduction to Modeling and Optimization, ETH Zürich - Inst. Mess- und Regeltechnik: Springer Berlin Heidelberg New York, 2007.
- [4] A. Hofer, Skriptum - Computerunterstützte Modellbildung und Simulation, Graz: IRT - Technische Universität Graz, 2004.
- [5] Office of Transportation and Air Quality (OTAQ), „Dynamometer Drive Schedules,“ United States Environmental Protection Agency, 6. Februar 2013. [Online]. Available: <http://www.epa.gov/nvfe/testing/dynamometer.htm#vehcycles>. [Zugriff am 13. März 2013].
- [6] D. Foster, „GPX 1.1 Schema Documentation,“ 19. Juli 2007. [Online]. Available: <http://www.topografix.com/GPX/1/1/>. [Zugriff am 05. Jänner 2013].
- [7] R. Bellman, „The Theory of Dynamic Programming,“ *American Mathematical Society*, 30 July 1954.
- [8] G. Scheider und H. Mikolcic, Einführung in die Methode der dynamischen Programmierung, Wien: Oldenburg Verlag München, 1972.

- [9] O. Sundström und L. Guzzella, „A Generic Dynamic Programming Matlab Function,“
In Proceedings of the 18th IEEE International Conference on Control Applications, pp.
1625-1630, Saint Petersburg, Russia, 2009.
- [10] A. Fröberg, *Efficient Simulation and Optimal Control for Vehicle Propulsion*,
Linköping, Sweden: LiU-Tryck, 2008.
- [11] B. Pfeiffer und R. Dittmar, *Modellbasierte prädiktive Regelung*, München: Oldenburg
Wissenschaftsverlag GmbH, 2004.
- [12] L. Grüne und J. Pannek, *Nonlinear Model Predictive Control - Theory and Algorithms*,
Bayreuth: Springer London Dordrecht Heidelberg New York, 2011.
- [13] D. Q. Mayne, J. B. Rawlings, C. V. Rao und P. O. M. Scokaert, „Constrained model
predictive control: Stability and optimality,“ *Automatica*, Bd. 36, pp. 789-814, 2000.
- [14] T. Zheng, *Model Predictive Control*, Rijeka, Croatia: Sciyo, 2010.
- [15] M. Back, *Prädiktive Antriebsregelung zum energieoptimalen Betrieb von
Hybridfahrzeugen*, Karlsruhe, Deutschland: Universitätsverlag Karlsruhe, 2005.
- [16] M. Adler, „Vehicle driveline control for predictive green driving,“ *Diplomarbeit*,
Universität Maribor, 2012.