

DIPLOMARBEIT

Optimierung von Parametern einer virtuellen Motorsteuerung basierend auf einem nichtlinearen dynamischen Motormodell

Zur Erlangung des akademischen Grades eines Diplom-Ingenieurs

Christian Ellmer

Technische Universität Graz
Fakultät für Elektrotechnik und Informationstechnik
Studienrichtung: Elektrotechnik

Betreuer:

Ao. Univ. Prof. Dipl.- Ing. Dr. Magele Christian
Institut für Grundlagen und Theorie der Elektrotechnik

Externe Betreuer:

Dipl.-Ing. Birnstingl Johannes
AVL List GmbH

Dipl.-Ing. Stolz Michael
Das virtuelle Fahrzeug (ViF)

Graz, Mai 2012

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.

Graz, am

.....
(Unterschrift)

Kurzfassung

Moderne Diesel PKWs unterliegen den immer steigenden Anforderungen von Gesetzgebern in Bezug auf Emissionsgrenzwerte und von Kunden, die sich sparsamere Autos bei gleichbleibender oder höherer Leistung wünschen. Um diese Anforderung zu erfüllen, besitzen aktuelle Dieselmotoren eine Vielzahl von mechanischen und elektronischen Komponenten. Bei einer optimalen Einstellung regeln diese Elemente den Verbrennungsprozess, sodass der Motor ruhiger läuft und die Emissionen verringert werden. Die Anzahl von solchen elektrischen und mechanischen Stellgrößen, die sich unter anderem auch gegenseitig beeinflussen, steigt deshalb von Jahr zu Jahr an. Heutige Motorkonzepte besitzen daher eine so hohe Komplexität, dass eine manuelle Einstellung dieser Regelsysteme meist nur von sehr erfahrenen Ingenieuren durchgeführt werden kann.

Durch immer neuere Modell-Architekturen und schnellere Computer ist es möglich, das nichtlineare dynamische Verhalten der Motorgrößen zu erfassen und nachzubilden. In dieser Forschungsarbeit wird unter Nutzung eines nichtlinearen dynamischen Motormodells, welches über den gesamten Betriebsbereich gültig ist, eine Optimierung des Ladedruckreglers für den Turbolader durchgeführt. Dazu wird der Teil der Motorsteuerung virtuell nachgebaut, in welchem sich der Ladedruckregler befindet. Die Parameter dieses Ladedruckreglers werden mit einem stochastischen Optimierungsalgorithmus eingestellt, um das experimentell ermittelte Motormodell möglichst optimal zu regeln. Die Einstellung der Parameter erfolgt anhand von verschiedenen Beurteilungskriterien, welche minimiert werden müssen. Mit Hilfe des Motormodells und eines multi-kriteriellen Optimierers lässt sich schließlich eine automatische Optimierung am Computer durchführen.

Am Ende der Forschungsarbeit ist zu erkennen, welches Verbesserungspotenzial die modellbasierte Regleroptimierung bringt und wie der Optimierungsvorgang des Ladedruckreglers funktioniert, indem das nichtlineare Motormodell in der Simulation geregelt wird. Durch die unterschiedlichen Beurteilungskriterien erhält der erfahrene Ingenieur am Schluss eine Menge von optimalen Lösungen, die jeweils ein anderes Kriterium bevorzugen. Seine Aufgabe wird es dann sein, einen geeigneten Parametersatz zu wählen und diesen am realen Fahrzeug zu testen.

Abstract

Modern diesel powered vehicles do not only have to compete with the increasing demands of legislators with regard to emission limits, but also have to meet the expectations of the customers who want fuel-efficient cars but no loss in performance. In order to meet these requirements and expectations, modern diesel engines feature lots of mechanical and electronic components. These components are meant to control the combustion process in an optimal way, which reduces emissions and makes the engine run considerably quieter. As the number of such electrical and mechanical components increases from year to year, the engines have reached such a high complexity, that manual calibration of these control systems can usually only be performed by highly experienced engineers.

Due to newer model architectures and faster computers, it is possible to capture and reproduce the nonlinear dynamic behavior of the engine outputs. In this thesis a nonlinear dynamic engine model for the whole operating area is used to optimize the boost pressure controller of the turbocharger. A part of the engine control unit (ECU), in which the boost pressure controller is located, was constructed virtually. The parameter of the boost pressure controller was adjusted by a stochastic optimization algorithm to control the experimentally determined engine model. As there are different evaluation criteria for the setting of the turbocharger controller, a multi-objective optimization algorithm was applied. With the help of the engine model an automatic optimization may be achieved merely by means of using a computer.

The thesis results show an improvement due to the model-based controller calibration, as well as an optimization of the control parameters of the turbocharger, which controls the nonlinear system. At the end of the optimization algorithm the experienced engineer receives a set of optimal solutions to minimize the different evaluation criteria. His task will be to choose a suitable set of parameters and to test this set on a real vehicle.

Inhaltsverzeichnis

Abbildungsverzeichnis.....	III
Tabellenverzeichnis	IV
Formelverzeichnis.....	IV
1 Einleitung.....	1
1.1 Aufgabenstellung	1
1.2 Beschreibung Status Quo	2
1.3 Zielsetzung.....	3
1.4 Stand der Technik.....	4
2 Grundlagen.....	7
2.1 Dieselmotor	7
2.1.1 Funktionsweise Dieselmotor.....	7
2.1.2 Luftpfad.....	9
2.1.3 Ladedruckregler.....	11
2.1.3.1 Wastegate Turbolader.....	11
2.1.3.2 VTG Turbolader	13
2.2 Grundlagen der Regelungstechnik	14
2.2.1 PID Regler	16
2.2.2 Sprungantwort.....	19
2.2.3 Nichtlinearitäten.....	20
2.2.4 Reglerkalibrierung	21
2.3 Testmotor.....	21
2.4 Motorsteuergerät ECU.....	21
2.4.1 Ladedruckregler.....	22
2.4.2 Zusätzliche ECU Funktionen	23
2.5 Global dynamisches Motormodell	23
2.5.1 Künstliche neuronale Netze	24
2.5.2 LOLIMOT (LOcal LInear MOdel Tree)	26
2.5.3 dINN (dynamic Intelligent Neural Network).....	28
2.5.4 Dynamische Identifikation	30
3 Modellbasierte Regleroptimierung.....	31
3.1 Fahrmanöver	31
3.2 Zielsetzung.....	33
3.3 Optimierungsalgorithmus.....	36

3.3.1	Differential Evolution (DE).....	36
3.3.2	Basiskonzept	37
3.3.2.1	Initialisierung.....	38
3.3.2.2	Mutation.....	39
3.3.2.3	Crossover.....	42
3.3.2.4	Selektion.....	44
3.3.2.5	Nebenbedingungen	45
3.3.3	Multikriterielle Optimierung.....	46
3.3.3.1	Pareto Front	47
3.3.3.2	Non-Dominated Sorting Differential Evolution (NSDE).....	48
4	Implementierung	51
4.1	Entwicklungsumgebung.....	51
4.2	Modellgenerierung	51
4.3	Implementierung in Simulink.....	52
4.3.1	Global Dynamisches Modell	53
4.3.2	Motorsteuerung ECU.....	53
4.3.3	Parameteradaption	55
4.4	Validierung.....	57
4.5	Modellbasierte Regleroptimierung	60
4.5.1	Programmablauf.....	60
4.5.2	Optimierungsalgorithmus	62
5	Experimente und Resultate	63
5.1	Auswahl der optimierten Parameter.....	63
5.1.1	Multikriterielle Funktionswerte.....	63
5.1.2	Auswahl der optimierten Parameter.....	66
5.2	Anwendung der optimierten Parameter	68
5.2.1	Optimierten Zyklus.....	68
5.2.2	Robustheitsuntersuchung.....	71
5.2.3	Validierung.....	71
6	Schlussfolgerung und Ausblick	77
	Quellenverzeichnis.....	80
	Literaturverzeichnis.....	80
	Verzeichnis der Internetquellen	82

Abbildungsverzeichnis

Abbildung 1.1:	Status Quo, traditionelle Optimierung.....	2
Abbildung 1.2:	Optimierung anhand eines Modells	3
Abbildung 2.1:	Arbeitsschritte Dieselmotor	9
Abbildung 2.2:	Luftpfad Dieselmotor.....	10
Abbildung 2.3:	Wastegate mit Stellglied.....	12
Abbildung 2.4:	a.) VTG b.) Prinzip VTG.....	13
Abbildung 2.5:	Regelung und Steuerung	14
Abbildung 2.6:	Schematische Darstellung eines Standard Regelkreises.....	15
Abbildung 2.7:	a.) Stellgrößenbeschränkung b.) Regelabweichung	18
Abbildung 2.8:	a.) Istwert b.) Anti Windup Maßnahme.....	19
Abbildung 2.9:	Sprungantwort.....	19
Abbildung 2.10:	a.) Neuron b.) dynamisches Neuron.....	24
Abbildung 2.11:	Prinzip von Neuronalen Netzen.....	25
Abbildung 2.12:	a.) lokale Teilmodelle b.) Gültigkeitsfunktion.....	25
Abbildung 2.13:	a.) Approximation der nichtlineare Funktion b.) Teilung.....	26
Abbildung 2.14:	Baumstruktur dINN	28
Abbildung 2.15:	APRBS Signal	30
Abbildung 3.1:	Fahrmanöver: Gaspedalstellung.....	31
Abbildung 3.2:	Fahrmanöver: Drehzahl/Drehmoment.....	32
Abbildung 3.3:	Fahrmanöver: Regelabweichung/Großsignalbereich	32
Abbildung 3.4:	Kleinsignal-/ Großsignalbereich.....	33
Abbildung 3.5:	Multikriterielle Optimierung.....	34
Abbildung 3.6:	Programmablauf klassischer DE	37
Abbildung 3.7:	a.) Zielfunktion b.) Initialisierung.....	39
Abbildung 3.8:	a.) Differenzvektor b.) Mutation.....	40
Abbildung 3.9:	Zielfunktion	41
Abbildung 3.10:	Differenzvektor Verteilung.....	41
Abbildung 3.11:	Crossover.....	43
Abbildung 3.12:	a.) Pareto Front b.) Pareto Dominanz	47
Abbildung 3.13:	Schema des NSGA-II.....	48
Abbildung 3.14:	a.) Pareto Rangordnung b.) Crowding Distanz.....	49
Abbildung 4.1:	global dynamisches Modell.....	51
Abbildung 4.2:	Simulink Aufbau des global dynamischen Motormodell	53
Abbildung 4.3:	Simulink Aufbau des Ladedruckregler ECU.....	54

Abbildung 4.4:	Simulink Aufbau der Parameteradaption.....	56
Abbildung 4.5:	Validierung: Drehzahl/Drehmoment	57
Abbildung 4.6:	Validierung offener Regelkreis: Motorsteuergerät	58
Abbildung 4.7:	Validierung geschlossener Regelkreis.....	59
Abbildung 4.8:	Regelstrecke	60
Abbildung 4.9:	Programmablaufplan	61
Abbildung 5.1:	a.) Pareto Front f_2 zu f_1 b.) Pareto Front f_3 zu f_1	64
Abbildung 5.2:	Pareto Front f_4 zu f_1	65
Abbildung 5.3:	Pareto Front f_3 zu f_2	65
Abbildung 5.4:	a.) K_{pPos} zu ISE b.) K_{iPos} zu ISE	66
Abbildung 5.5:	a.) K_{pNeg} zu ISE b.) K_{iNeg} zu ISE	67
Abbildung 5.6:	Optimierungszyklus	69
Abbildung 5.7:	Validierungszyklus 1	74
Abbildung 5.8:	Validierungszyklus 2	75
Abbildung 5.9:	NEDC.....	76

Tabellenverzeichnis

Tabelle 2.1:	Regelungstechnische Grundelemente	16
Tabelle 4.1:	Kennfeld.....	55
Tabelle 5.1:	Kleinsignalparameter	67
Tabelle 5.2:	Großsignalparameter	67

Formelverzeichnis

Formel (2.1).....	29
Formel (2.2).....	29
Formel (2.3).....	30
Formel (3.1).....	33
Formel (3.2).....	35
Formel (3.3).....	35
Formel (3.4).....	35
Formel (3.5).....	35
Formel (3.6).....	37
Formel (3.7).....	38
Formel (3.8).....	39
Formel (3.9).....	40

Formel (3.10).....	42
Formel (3.11).....	44
Formel (3.12).....	46
Formel (3.13).....	50
Formel (5.1).....	63

1 Einleitung

Die Einstellung von Parametern (im Folgenden auch „Kalibrierung“ genannt) von Motorsteuergeräten (engl. Engine Control Unit, kurz ECU) für neue Fahrzeuge ist entscheidend für viele Eigenschaften des Fahrzeugs und ist heute ein wesentlicher Bestandteil des Entwicklungsprozesses von neuen Motoren und Fahrzeugen. Der modellbasierte Ansatz eröffnet dabei ein signifikantes Einsparungspotenzial durch reduzierte Arbeitsaufwände im Fahrzeug, da die Kalibrierung einer Regelung in einer Simulationsumgebung durchgeführt werden kann.

1.1 Aufgabenstellung

In modernen Fahrzeugen werden heutzutage elektronische Motorsteuergeräte eingesetzt, um ein möglichst gutes Betriebsverhalten über den gesamten Betriebsbereich zu gewährleisten. Das setzt die Kalibrierung der entsprechenden Parameter innerhalb des Steuergeräts und deren Optimierung voraus. Die Reglerkalibrierung wird dabei zu einem großen Teil per Hand im Fahrzeug durchgeführt, was üblicherweise zeit- und kostenintensiv ist.

Durch neue Verfahren in der Modellbildung von realen Systemen und steigende verfügbare Rechenleistung von heutigen Computern ist es möglich, die Optimierung des Gesamtsystems bestehend aus ECU, Motormodell und Aktuatoren mit Hilfe von numerischen Algorithmen durchzuführen. Dabei wird die Dynamik des Motors erfasst, indem der reale Motor vermessen und durch neuronale Netze in einem dynamischen Motormodell nachgebildet wird. Dieses dynamische Modell und die im Steuergerät integrierte Reglerstruktur werden in eine Simulationsumgebung eingebaut und zur Optimierung mehrfach simuliert und ausgewertet. Die Struktur des Steuergeräts ist dabei durch den Hersteller bereits vorgegeben. Der wesentliche Vorteil einer modellbasierenden Kalibrierung besteht darin, die automatische Parametrierung offline in einer Simulation durchführen zu können, um damit den Arbeitsaufwand im Fahrzeug zu reduzieren.

Für die Optimierung von Reglerparametern zur Regelung von linearen Systemen gibt es bereits eine Vielzahl von Entwurfsmethoden, jedoch liegen für die Regleroptimierung in Verbindung mit nichtlinearen experimentell ermittelten Systemen noch keine geeigneten Methoden vor. Ziel dieser Arbeit ist es daher, anhand eines vorhandenen Modells von Motor und ECU über eine Optimierungsroutine die Parameter des Reglers automatisch zu optimieren. Der Algorithmus soll zur Optimierung der Reglerparameter des Ladedrucks eines Diesel-PKW's eingesetzt werden und anhand der Ergebnisse soll bewertet werden, ob sich diese Optimierungsstrategie bewährt.

Bei dieser automatischen Kalibrierung müssen verschiedene Beurteilungskriterien (wie z.B. Überschwingen, Anstiegszeit usw.) berücksichtigt werden, um die spezifischen Anforderung (z.B.: Fahrkomfort, Minimierung des Kraftstoffverbrauchs) des Kunden zu gewährleisten. Durch Anwendung des Optimierungsalgorithmus erhält der Applikateur verschiedene Lösungsvorschläge, aus denen er den bestmöglichen Kompromiss zwischen den widersprechenden Einzelzielen auswählen kann. Als Applikateur wird in dieser Arbeit ein Ingenieur bezeichnet, welcher versucht, die verschiedenen Regelsysteme wie den Turbolader Regler möglichst optimal abzustimmen.

1.2 Beschreibung Status Quo

Eine wesentliche Herausforderung für die Kalibrierung ist das Finden von Kompromissen zwischen widersprüchlichen Anforderungen wie z.B. Leistung vs. Verbrauch, Komfort vs. Dynamik, Stickoxide (NO_x) vs. Partikelemission. Die manuelle Kalibrierung solcher Regelsysteme wird meist nur von sehr erfahrenen Ingenieuren (siehe Abbildung 1.1) durchgeführt.

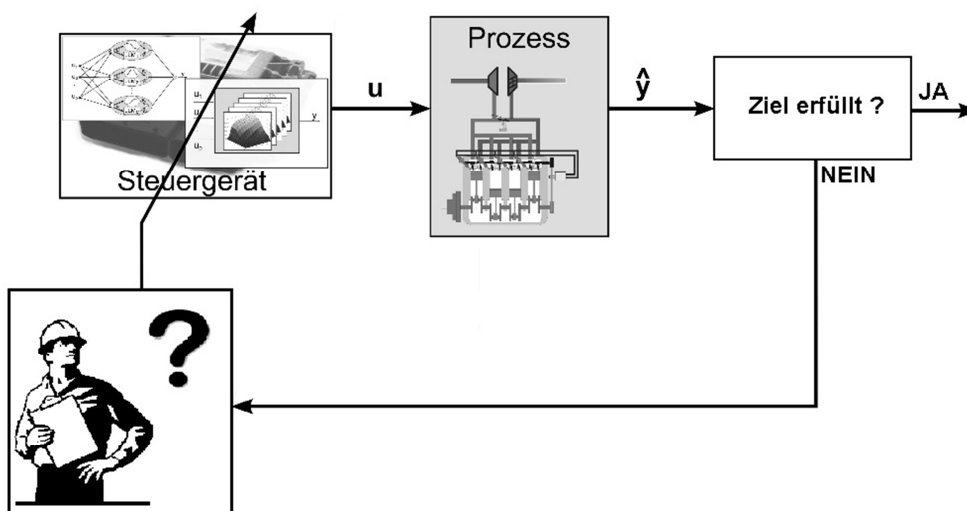


Abbildung 1.1: Status Quo, traditionelle Optimierung

Der Ingenieur analysiert dabei eine bestimmte Fahrsituation und beurteilt diese anhand von bestimmten Bewertungskriterien. Wenn ein bestimmtes Problem bei einer Fahrsituation (z.B. zuviel Überschwingen) auftritt, so versucht der Ingenieur die Parameter entsprechend zu ändern, um die Situation zu verbessern. Wenn er erfolgreich ist, wird das Verhalten der neu eingestellten Parameter in den übrigen Betriebspunkten überprüft. Ergeben sich aus der veränderten Kalibrierung keine neuen Probleme, so sind diese Parameter als annehmbar zu bewerten. Üblicherweise

sind die meisten Parameter voneinander abhängig, so dass die Optimierung eines Parameters zum Nachjustieren eines anderen führt und das neue Ergebnis daher auch unzureichend werden kann. Um diese Aufgabe mit vertretbarem Aufwand an Zeit und Kosten zu meistern, sind neue Methoden in der Kalibrierung erforderlich. Der Einsatz von automatischen Optimierungsroutinen innerhalb einer modellbasierten Kalibrierung ist deshalb unumgänglich.

1.3 Zielsetzung

Die Neuheit dieser Arbeit besteht darin, eine Optimierung der Parameter des Ladedruckreglers eines Diesel-PKW's durchzuführen, um so die nichtlineare Strecke optimal zu regeln. Der Optimierungsalgorithmus läuft dabei als Skript ab, mit welchem die Ladedruckregelung kalibriert wird. Bei dieser modellbasierten Kalibrierung wird der reale Motor auf einem Computer mit Hilfe eines Modells simuliert. Ein automatischer Algorithmus löst dabei die Kalibrierungsaufgabe, indem das Regelsystem mit dem Motormodell mehrfach simuliert wird (siehe Abbildung 1.2).

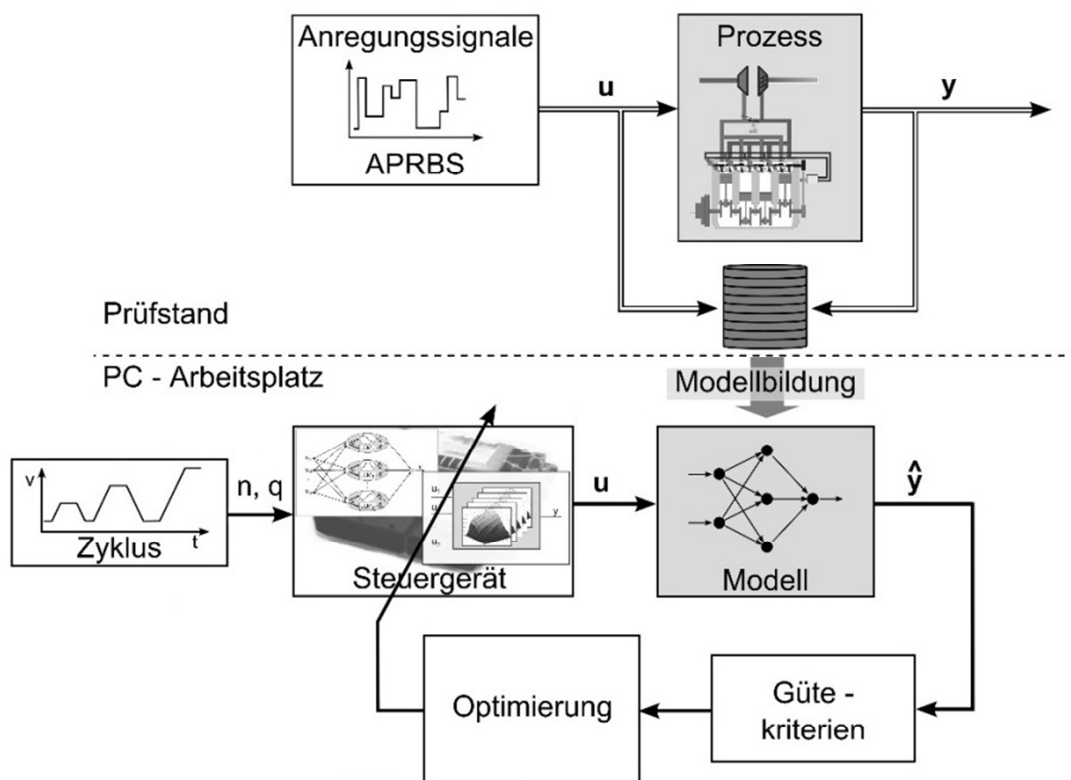


Abbildung 1.2: Optimierung anhand eines Modells

Ein wesentlicher Vorteil des modellbasierten Ansatzes ist die Möglichkeit, das globale Verhalten des Motors einschließlich der komplexen Einflüsse von Drehzahl und Last reproduzierbar zu beschreiben. Ein einziges Modell kann dabei den gesamten Betriebsbereich des Motors abdecken. Da es sich um ein experimentell ermitteltes Motormodell handelt, welches aus mehreren verschiedenen mathematischen Teilfunktionen zusammengesetzt ist, wird ein stochastischer Optimierungsalgorithmus verwendet. Als Optimierungskriterium werden Kenngrößen verwendet, welche durch einen kundenspezifischen Fahrzyklus oder einen für die Fahrzeugabnahmeprüfung vorgeschriebenen Fahrzyklus, ermittelt werden. Durch diesen Fahrzyklus und das simulierte Motorverhalten wird versucht, die Parameter des virtuellen Motorsteuergeräts mit Hilfe eines geeigneten Algorithmus zu optimieren.

Am Ende der Optimierung soll der Applikateur eine größere Auswahl von möglichen Parametern haben, welche verschiedene Kriterien minimieren. So ist es möglich, dass er bei der Validierung am Fahrzeug je nach Kundenwunsch (z.B. besseres Fahrverhalten) einen anderen Parametersatz verwenden kann.

1.4 Stand der Technik

Im Folgenden werden aktuelle Arbeiten kurz vorgestellt, die sich mit der Thematik der Parameteroptimierung des Motorsteuergeräts auseinandersetzen. Die Literaturrecherche wurde dabei in Richtung folgender Punkte betrieben:

- 1.) Modellprädiktive Regler zur Regelung von dynamischen Modellen
- 2.) Stochastische multikriterielle Optimierungsverfahren mit neuronalen Netzen
- 3.) Firmen, welche die Optimierung von Motorsteuergeräten durchführen
- 4.) Aktuelle stochastische Optimierungsverfahren

Zu 1.) Das Institut für Regelungstechnik der RWTH (Rheinisch-Westfälische Technische Hochschule Aachen) forscht intensiv an der Luftpfadregelung des aufgeladenen Dieselmotors (siehe RWTH Homepage [38]). Die wissenschaftliche Arbeit von Rückert und Abel beschäftigt sich in [32] mit der Regelung durch einen modellgestützten prädiktiven Regler (MPR). Das Motormodell besteht aus einem Mehrgrößensystem mit zwei gekoppelten Systemen (Turbolader und Abgasrückführung), welches ein nichtlineares Verhalten aufweist. Um das nichtlineare Verhalten der Strecke zu regeln, wird zwischen verschiedenen Parametern umgeschaltet. Da die Reg-

lerstruktur bereits durch den Hersteller des Motorsteuergeräts fix vorgegeben ist, wurde nicht weiter in diese Richtung recherchiert.

Zu 2.) In der wissenschaftlichen Arbeit „Multiobjective optimization of feedforward control maps in engine management systems towards low consumption and low emissions“ konnten Hafner und Isermann in [16] die verschiedenen Parameter in der ECU optimieren, um so einen geringeren Verbrauch und niedrigere Emissionen zu erreichen. Als Optimierungsalgorithmus verwendeten sie einen sequential quadratic programming (kurz: SQP) Algorithmus.

Vossoughi und Rezazadeh [34] optimierten ebenfalls das Motorsteuergerät, um den Verbrennungsmotor besser zu regeln. Als Regelstrecke bedienten sie sich eines neuronalen Netzes. Sie verglichen zwei multikriterielle Optimierer (DPGA ... Distance based Pareto Genetic Algorithm und den NSGA ... Non-Dominated-Sorting Genetic Algorithm), um den Verbrauch und die Emissionen zu minimieren.

Alonso *et al.* verwenden in [1] eine Kombination aus neuronalen Netzen und stochastischen Optimierungsverfahren zur Reduktion von Diesel-Emissionen. Zur Optimierung bedienten sie sich des genetischen Algorithmus (GA).

Lochau *et al.* veröffentlichten ihre Arbeit [21] über modellbasierte Parameter-Optimierung der ECU mithilfe eines genetischen Algorithmus. Hierzu setzten sie eine multikriterielle Optimierung mit Nebenbedingungen auf Grundlage des GA an. In dieser Arbeit haben sie die maximale Motor-Leistung unter Minimierung des Verbrauchs abhängig vom Betriebsbereich optimiert.

Zu 3.) Die Firma ESTECO setzt ihre Softwarelösung modeFRONTIER (siehe ESTECO Homepage [36]) ebenfalls zur Optimierung eines Drehzahlreglers in der ECU ein. Auch hier kommt ein multikriterieller genetischer Algorithmus in Kombination mit einem einfachen Optimierungsalgorithmus (Simplex-Algorithmus) zum Einsatz.

Eine bedeutende Veröffentlichung ist von der Firma ETAS (siehe ETAS Homepage [37]). Die ETAS bietet mehrere Anwendungen zur Entwicklung und Diagnose von Motorsteuergeräten (ECUs) an. Sie ist eine Tochterfirma der Bosch AG, welche ca. 70%-80% der heute eingesetzten Dieselmotorsteuergeräte für den Automobilbereich herstellt. Die ETAS bietet ihren Kunden eine neue Anwendung Namens ASCMO (Advanced Simulation for Calibration, Modeling and Optimization) an. Diese Software beinhaltet sowohl Modellbildung als auch eine Optimierung unter Verwendung eines bestimmten Fahrzyklus (z.B.: New European Driving Cycle, kurz: NEDC). Kruse *et al.* haben eine Arbeit [20] über diese Anwendung veröffentlicht, indem sie eine Reglerka-

librierung mit Hilfe des Non-Dominated-Sorting Genetic Algorithm (NSGA-II) von Deb *et al.* [10] durchführten.

Basierend auf diesen Arbeiten, in denen Motormodelle mit Hilfe von neuronalen Netzen generiert und die Optimierung mit stochastischen Optimierungsalgorithmen durchgeführt werden, fällt die Wahl der Optimierungsstrategie in der vorliegenden Arbeit auf einen stochastischen multikriteriellen Algorithmus auf Basis der biologischen Evolution.

Zu 4.) Der Differential Evolution (DE) von Price *et al.* [29] ist einer der aktuell Besten stochastischen Optimierungsverfahren auf Basis der biologischen Evolution (siehe Das und Suganthan [9]). Der NSDE (Non-Dominated-Sorting Differential Evolution), wie er von Antony und Xiaodong in [5] beschrieben wird, ist eine Erweiterung des DE und löst mit Hilfe des Non-Dominated-Sorting Algorithmus von Deb *et al.* [10] auch Probleme mit mehreren Zielfunktionen.

2 Grundlagen

In diesem Kapitel werden die wichtigsten Grundlagen über die in dieser Arbeit behandelten Themen wie Dieselmotor, Ladedruck, Modellbildung und Regelungstechnik erklärt.

2.1 Dieselmotor

Vor ca. 120 Jahren meldete Rudolf Diesel sein Konzept über den Dieselmotor zum Patent an. Die wohl wichtigsten Merkmale bei den derzeit aktuellen Diesel Verbrennungsmotoren sind (siehe Mollenhauer und Tschöke[23]):

- Selbstentzündung,
- innere Gemischbildung und
- Direkteinspritzung.

Beim Dieselpprozess wird eine geregelte Menge von Kraftstoff am Ende der Zylinderverdichtung in eine Vertiefung in den Kolbenboden eingespritzt. Die daraus resultierende Gemischbildung von Kraftstoff und Luft erfolgt im Brennraum des Zylinders und wird durch die hohen Temperaturen (infolge des hohen Drucks) selbstentzündet. Diese Verbrennung bewirkt eine Temperaturerhöhung, daraus entsteht Druck im Brennraum, der wiederum den Kolben bewegt. Der Dieselmotor weist einen höheren Wirkungsgrad als der Ottomotor auf und ist daher die bevorzugte Wahl bei den verschiedensten Fahrzeugtypen wie PKW, LKW und bei großen Schiffen (siehe Pischinger *et al.* [28]).

Dieselmotoren können wie folgt unterteilt werden:

- 2- oder 4-Takt-Motoren,
- Motoren mit direkter oder indirekter Einspritzung und
- Saugmotoren oder turboaufgeladene Motoren.

Im nächsten Kapitel wird das Prinzip eines Dieselmotors anhand eines 4-Takt-Motors erklärt.

2.1.1 Funktionsweise Dieselmotor

Derzeit neu produzierte PKW-Dieselmotoren werden vorzugsweise als 4-Takt-Motoren mit direkter Einspritzung und Turbolader hergestellt. Aus diesem Grund folgt an dieser Stelle eine kurze Erklärung der Funktionsweise. Bei einem 4-Takt Motor durchläuft der Dieselmotor 4 Arbeits-

schritte, um Arbeit zu verrichten. Dabei wird nur jede zweite Umdrehung dazu genutzt, den Kolben anzutreiben.

Abbildung 2.1 zeigt diese vier Arbeitsschritte eines Dieselmotors etwas genauer. (siehe Homepage Dieselmotor [39])

- 1.) **Ansaugen:** Im ersten Arbeitsschritt bewegt sich der Kolben vom oberen Totpunkt zum unteren Totpunkt. Bevor der Kolben den oberen Totpunkt erreicht, öffnet sich das Einlassventil. Bei der Abwärtsbewegung des Kolbens entsteht ein Unterdruck im Brennraum und durch das geöffnete Einlassventil wird Frischluft zugeführt. Das Auslassventil schließt bereits nach dem oberen Totpunkt, um nicht ausströmende Restgase anzusaugen. Das Einlassventil verriegelt nach dem unteren Totpunkt, um ein Maximum an Frischgasen im Brennraum zu erhalten.
- 2.) **Verdichten:** Während sich der Kolben vom unteren zum oberen Totpunkt bewegt, ist das Einlassventil geschlossen und die im vorherigen Arbeitsschritt angesaugte Luft wird verdichtet. Durch die Druckerhöhung erhöht sich auch die Lufttemperatur im Brennraum. Die Einspritzdüse spritzt kurz vor dem oberen Totpunkt den fein zerstäubten Kraftstoff ein, welcher sich mit der Luft vermischt und es daher zur inneren Gemischbildung kommt.
- 3.) **Arbeiten:** Die hohen Temperaturen im Verdichtungsraum entflammen diese Gemischbildung nach einer gewissen Zeitspanne (Zündverzug). Der eingespritzte Kraftstoff verbrennt nach dem Zündbeginn sofort. Der Verbrennungsdruck wirkt auf die Kolbenfläche und überträgt die Kolbenkraft über die Pleuelstange auf die Kurbelwelle, wodurch eine Drehbewegung entsteht.
- 4.) **Ausstoßen:** Im vierten und letzten Arbeitsschritt öffnet sich das Auslassventil vor dem unteren Totpunkt und durch die Aufwärtsbewegung des Kolbens werden die verbrannten Gase aus dem Verbrennungsraum gedrückt.

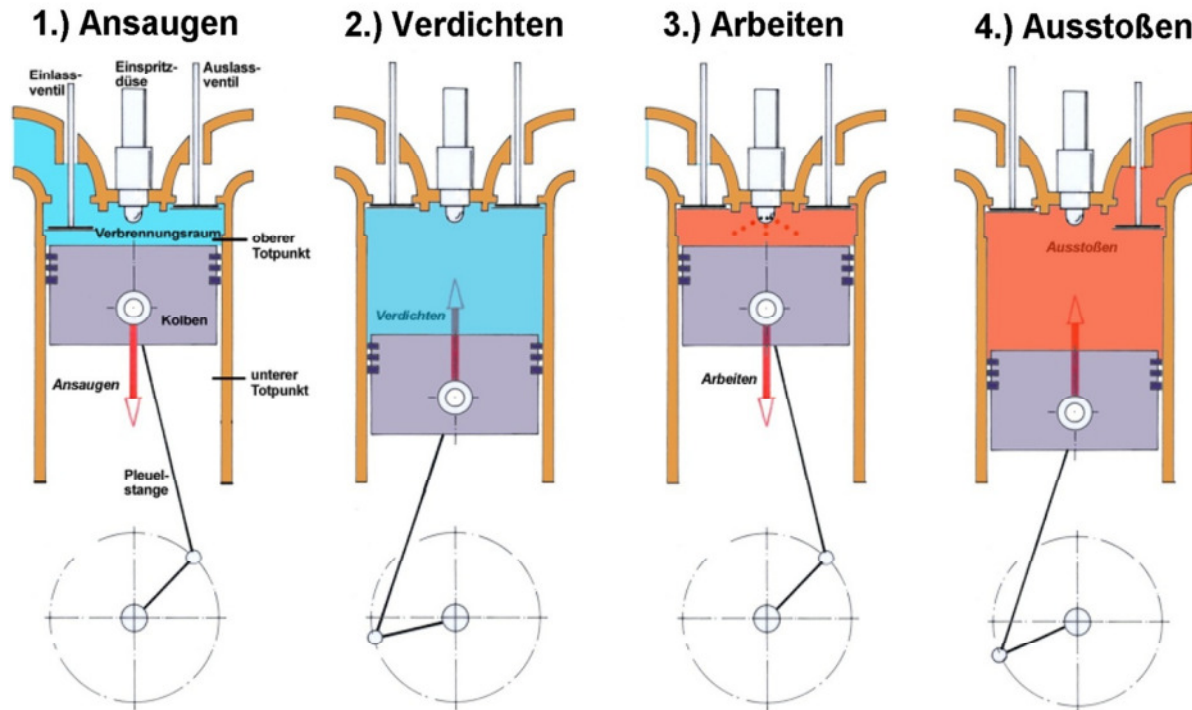


Abbildung 2.1: Arbeitsschritte Dieselmotor

Quelle: Dieselmotor [39]

Anhand des Funktionsprinzips sieht man, dass die zwei wichtigsten Medien einer Verbrennungskraftmaschine der Kraftstoff und die Luft sind. Da diese wissenschaftliche Arbeit die Regelung des Ladedrucks beinhaltet, wird im nächsten Kapitel der Luftpfad näher beschrieben.

2.1.2 Luftpfad

Der Luftpfad stellt die korrekte Menge an Sauerstoff für die Verbrennung im Zylinder zur Verfügung. Der Turbolader nutzt die Restenergie des Abgases, um eine Druckerhöhung im Einlasskanal zu ermöglichen. Dadurch wird der Füllgrad des Brennraumes verbessert und die Motorleistung steigt an.

Eine messbare Leistungssteigerung gegenüber den Saugmotoren ergibt sich durch die Druckerhöhung bei den turboaufgeladenen Motoren. Durch den höheren Druck wird bei gleichbleibendem Öffnungsintervall des Einlassventils mehr Sauerstoff in den Zylinder befördert. Dadurch kann eine größere Kraftstoffmenge in den Brennraum gespritzt und verbrannt werden, was zu mehr Arbeit pro Arbeitstakt führt. Infolge dessen erhöht sich das Drehmoment und es ergibt sich eine höhere Leistung an der Kurbelwelle (siehe Hack [15]).

Durch diese Leistungssteigerung können leistungsstärkere Motoren mit gleichen Abmessungen (im Vergleich mit den Saugmotoren) zum Einsatz kommen oder durch Downsizing des Motors die gleiche Leistung mit kleineren und leichteren Maschinen (weniger Verbrauch) erzielt werden (siehe Golloch [14]).

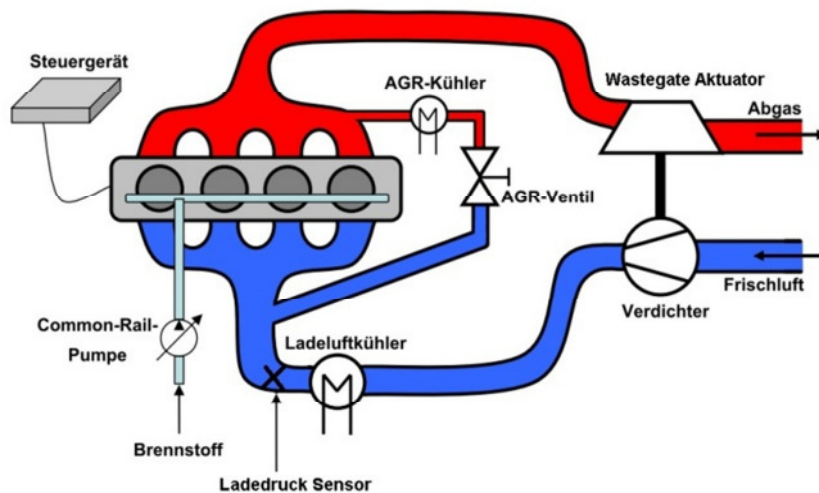


Abbildung 2.2: Luftpfad Dieselmotor

Quelle: Bosch [30]

Abbildung 2.2 zeigt das Prinzip der Turboaufladung. Die Abgase treiben die Turbine des Turboladers an, welche auf einer gemeinsamen Welle mit dem Verdichter (Kompressor) liegt. Durch eine Komprimierung der Frischluft kommt es zu einer Erhöhung des Drucks. Dies bewirkt einen höheren Luftmassenstrom, welcher mehr Sauerstoff in den Zylinder befördert. Daher kann eine größere Menge des Kraftstoffes eingespritzt werden, was eine höhere Leistung nach sich zieht.

Ein Nachteil der Aufladung des Motors ist jedoch, dass es durch die Komprimierung zu einer deutlichen Erwärmung der einströmenden Luft kommt und daher ein Ladeluftkühler benötigt wird, um die Dichte der Luft wieder zu reduzieren und dadurch die Ladungsmenge zu erhöhen. Trotz dieses Nachteils ist der Turbolader eine effektive und effiziente Maßnahme, um die Füllung des Zylinders durch die Energie der Abgase zu erhöhen. Sie ermöglichen selbst bei niedriger Drehzahl ein hohes Motordrehmoment (siehe Hack [15]).

2.1.3 Ladedruckregler

Wie bereits im vorherigen Kapitel erwähnt, ist der Turbolader abhängig von der Restenergie der Abgase. Wenn also die Motordrehzahl steigt, so gibt es mehr Abgasenergie und die Turbine des Turboladers dreht schneller. Dadurch komprimiert der Verdichter wieder mehr Luft, was wiederum zu einem Anstieg des Druckes führt und deswegen eine größere Menge an Sauerstoff im Zylinder zur Verfügung steht. Durch die Einspritzung einer entsprechend größeren Kraftstoffmenge und der Zündung wird die Drehbewegung erhöht. Dies hat wieder eine höhere Abgasrestenergie zur Folge und die Turbine wird noch schneller angetrieben. Die mechanischen und thermischen Grenzen des Motors können durch diese weiterführende Steigerung überschritten werden. Um die Aufladung des Motors zu begrenzen, haben sich deshalb 2 Varianten der Ladedruckregelung in der Automobilindustrie durchgesetzt (siehe Hack [15]):

- Wastegate (Bypassventil) und
- Variable Turbinen Geometrie (VTG).

Im Folgenden werden diese zwei Typen und deren Stellglieder kurz beschrieben.

2.1.3.1 Wastegate Turbolader

Die einfachste Art, eine Ladedruckregelung zu realisieren, ist mit Hilfe eines Bypassventils (Wastegate). Bei dieser abgasseitigen Regelung wird ein Teil der Abgase an der Turbine herumgeführt (Bypass) (siehe Abbildung 2.3). Sollte nun der Druck aufgrund des Verdichters ansteigen, so wird dieser zurück auf das Wastegate geleitet und drückt auf eine Membrane, welche eine Feder spannt. Das Wastegate öffnet sich und ein Teil der Abgasmenge wird an der Turbine vorbeigeführt (Bypass). Dadurch dreht sich die Turbine wieder langsamer und der Druck auf der Verdichterseite sinkt ab. Infolgedessen kommt weniger Druck zum Wastegate, welches sich daraufhin etwas schließt und wieder mehr Abgasenergie zur Turbine leitet. Es handelt sich also um einen mechanischen Regelkreis, der sich selbst einstellt. Das Wastegate ist hierbei das Regelorgan, welches entweder als Ventil oder Klappe ausgeführt werden kann (siehe Hack 15).

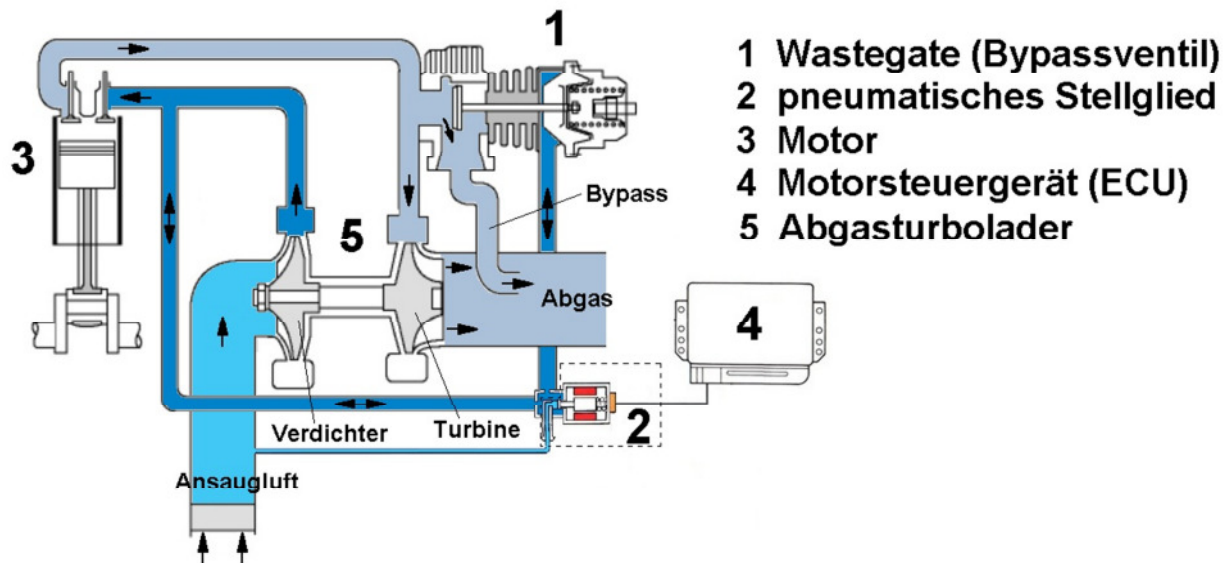


Abbildung 2.3: Wastegate mit Stellglied

Quelle: Bosch [30]

Ein großer Nachteil ist, dass sich so nur ein fixer Solldruck einstellen lässt, ab dem geöffnet wird. Diese Art der pneumatischen Regelung dient eigentlich mehr als Begrenzung bei Vollastbetrieb. Erst mithilfe eines Stellgliedes lässt sich das Wastegate über das Motorsteuergerät auf einen anderen Sollwert einstellen (siehe Abbildung 2.3). Ebenso nachteilig ist, dass sich das Bypassventil auf der heißen Abgasseite befindet, das heißt es wird thermisch dementsprechend hoch belastet.

Stellglied:

Zwei verschiedene Arten von Stellgliedern kommen derzeit am häufigsten in PKWs zum Einsatz.

- Vollelektronische Verstellung mittels Servomotor und
- pneumatische Druckdose mit elektrischem Luftventil.

Die Verstellung mittels Servomotor hat den Vorteil, dass sie präziser und weniger Abweichung hat, aber dafür teurer ist. Diese Art von Stellglied kommt bei dieser Arbeit nicht zum Einsatz und wird deshalb auch nicht weiter behandelt.

Die pneumatische Druckdose erhält ein pulsweitenmoduliertes (PWM) Signal vom Motorsteuergerät. Ein elektronisch pneumatischer Konverter wandelt die Signale in einen negativen Druck (Unterdruck) um. Dieser Unterdruck steuert ein System aus Membrane und Feder, welches den Druck aus der Verdichterseite (Frischlufte) variieren kann und zu dem Bypassventil zurückführt.

Ein wesentlicher Nachteil dieses Stellgliedes ist der nichtlineare Zusammenhang (Hysterese) zwischen elektrischem Signal und Öffnungswinkel der Wastegate Klappe. Wegen seiner hohen Robustheit und Temperaturunempfindlichkeit wird die pneumatische Druckdose dennoch oft als Stellglied eingesetzt (siehe Hack [15]).

2.1.3.2 VTG Turbolader

Eine weitere Art von Turbolader ist der VTG (Variable Turbinen Geometrie) Turbolader. Das Prinzip ist der Francis Turbine ähnlich. Am Eingang der Turbine befinden sich nicht rotierende Leitschaufeln, welche die Abgase zur Turbine führen (siehe Abbildung 2.4 b). Mit Hilfe eines Stellgliedes (Servomotor oder Druckdose) werden diese Leitschaufeln bewegt (siehe Abbildung 2.4 a). Durch Senkrechtmachen dieser Leitschaufeln wird der Strömungsquerschnitt reduziert und die Strömungsgeschwindigkeit, welche auf die Turbine trifft, erhöht sich und damit auch die Leistung des Verdichters. Stellt man die Leitschaufeln waagrecht, so ist der Strömungsquerschnitt größer und die Geschwindigkeit verringert sich. Dadurch sinkt die Leistung des Verdichters (siehe Hack [15]).

VTG (Variable Turbinen Geometrie)

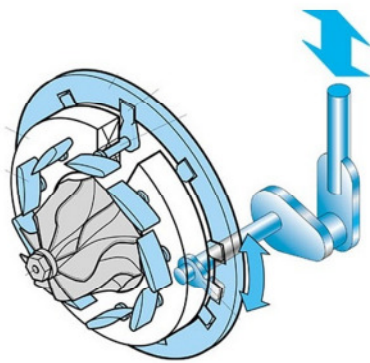
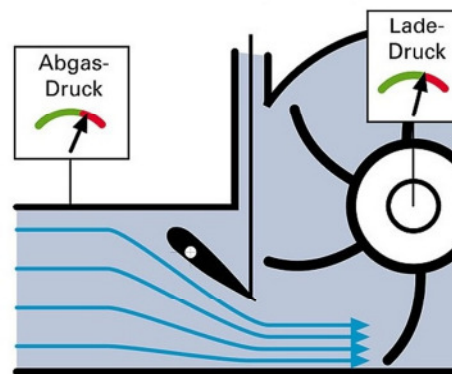


Abbildung 2.4: a.) VTG

Quelle: Bosch [30]



b.) Prinzip VTG

Der VTG Turbolader erlaubt die Regelung eines größeren Arbeitsbereiches des Motors. Die Vorteile bei dieser Art der Aufladung sind die bessere Dynamik und die höhere Komprimierung der Luft bei niedrigen Motordrehzahlen. Die höhere Komplexität dieses Systems schlägt sich aber auf die Kosten nieder.

2.2 Grundlagen der Regelungstechnik

Die Aufgabe eines Regelkreises ist, eine Regelgröße (Istwert) fortlaufend mit einer Führungsgröße (Sollwert) zu vergleichen und die Regeldifferenz durch einen geeigneten Stelleingriff zu verkleinern (siehe Föllinger [13]). Der Unterschied zwischen einer Regelung und einer Steuerung soll am Beispiel einer Heizung erklärt werden.

Bei einer Steuerung wird eine Raumtemperatur von z.B. 22°C eingestellt. Öffnet man ein Fenster, so sinkt die Raumtemperatur infolge der Störgröße (kalte Außenluft).

Bei einer Regelung wird die Innenraumtemperatur fortlaufend mitgemessen. Ändert sich nun der Istwert durch eine Störung (Öffnen des Fensters), so registriert der Regler diese Abweichung und versucht durch Ändern der Ventilstellung (Stellglied) den Sollwert (die vorgegebenen 22°C) wieder zu erreichen.

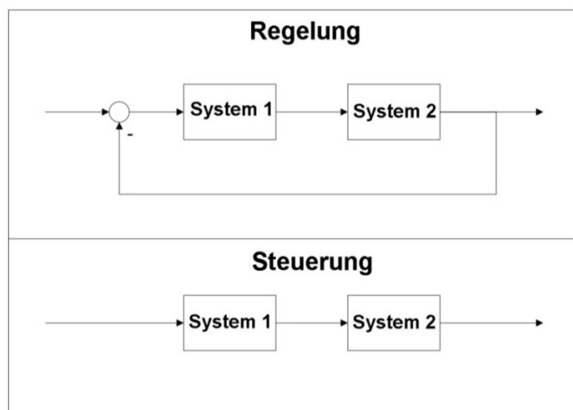


Abbildung 2.5: Regelung und Steuerung

Der wesentliche Unterschied zwischen Regelung und Steuerung liegt in der Rückführung des aktuellen Istwertes, welcher über ein Messglied (Sensor) erfasst wird. Abbildung 2.6 zeigt eine schematische Darstellung eines Standard Regelkreises mit den wichtigsten Systemen

- Regler,
- Stellglied (Aktuator),
- Regelstrecke und
- Messglied (Sensor).

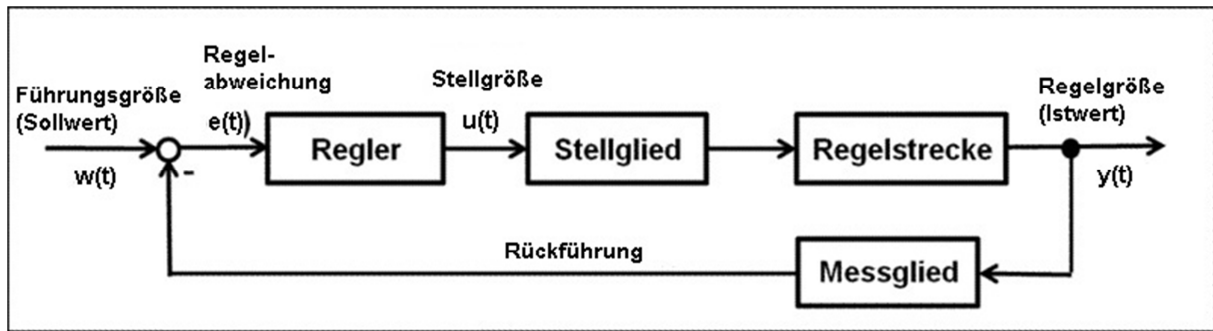


Abbildung 2.6: Schematische Darstellung eines Standard Regelkreises

Die einzelnen Systeme (Regler, Stellglied, Regelstrecke, Messglied) können sowohl lineares als auch nichtlineares Verhalten aufweisen. Bei einer Regelung von nichtlinearen Strecken kommen in der Automobilindustrie häufig nichtlineare Kennlinien zum Einsatz, um ein lineares Verhalten der Strecke nachzubilden. Die Regelstrecke wird meist über ein mathematisches Modell beschrieben. Die Identifikation des mathematischen Modells kann entweder theoretisch oder experimentell erfolgen. Da es sich bei dem Dieselmotor um ein komplexes nichtlineares System handelt, wird in Kapitel 2.5 näher auf die experimentelle Modellbildung eingegangen.

Ein Regelkreis kann durch folgende Kriterien beurteilt werden (siehe Föllinger [13]):

- **Dynamik:** Wie schnell der Regler auf eine Änderung des Soll-/Ist Vergleichs reagiert.
- **Stabilität:** Gibt an, ob das System zur Ruhe kommt oder nicht.
- **Genauigkeit:** Gibt die verbleibende Regelabweichung im stationären Zustand an.

Für die genaueren, mathematischen Definitionen bezüglich Stabilität usw. wird auf das Buch „Regelungstechnik“ von Horn und Dourdoumas [18] verwiesen. Bei falscher Wahl der Reglerparameter kann der Regelkreis langsam werden, es zu einer großen Regelabweichung kommen oder eine Schwingung (bis hin zur Instabilität) auftreten.

2.2.1 PID Regler

Der PID Regler ist ein Standardregler, welcher in der Industrie und auch im Motorsteuergerät häufig zum Einsatz kommt, weswegen er hier näher betrachtet wird.

Viele physikalische Gesetzmäßigkeiten in der Natur werden durch Differentialgleichungen ausgedrückt. Da der Umgang mit diesen sehr anspruchsvoll und umständlich ist, werden sie durch die Laplace Transformation in den Bildbereich (s-Bereich) übertragen. Im Bildbereich sind die Rechenoperationen einfacher und daher leichter handhabbar. Der PID Regler ist ein linearer Standardregler, welcher aus einem P-Glied, einem I-Glied und einem D-Glied besteht. Die nachfolgende Tabelle 2.1 beinhaltet eine kurze Übersicht der regelungstechnischen Grundelemente. Das Übertragungsverhalten von Ausgangssignal zu Eingangssignal bezeichnet man im s-Bildbereich als Übertragungsfunktion. Genauere Informationen zu dieser Thematik sind in dem Buch von Horn und Dourdoumas in [18] zu finden.

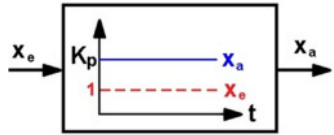
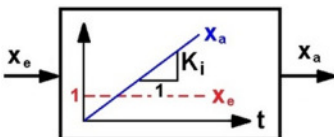
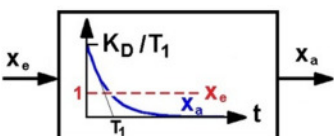
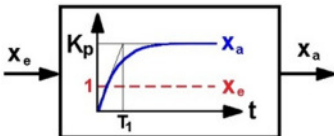
Element	Sprungantwort	s- Übertragungsfkt.	Parameter
P – Glied		$G(s) = \frac{x_a(s)}{x_e(s)} = K_P$	$x_{a(s)}$: Ausgangssignal $x_{e(s)}$: Eingangssignal K_p : Proportionalitätskonstante
I – Glied		$G(s) = \frac{1}{T_i \cdot s} = \frac{K_i}{s}$	K_i : Integrationskonstante T_i : Zeitkonstante
DT ₁ - Glied (Reales D-Glied)		$G(s) = \frac{K_D \cdot s}{T_1 \cdot s + 1}$	K_D : Differenziationskonstante T_1 : Zeitkonstante
PT ₁ - Glied		$G(s) = \frac{K_p}{T_1 \cdot s + 1}$	K_p : Proportionalitätskonstante T_1 : Zeitkonstante

Tabelle 2.1: Regelungstechnische Grundelemente

Proportional Glied (P- Glied): Das Ausgangssignal $u(t)$ ist proportional zum Eingangssignal $e(t)$ mit der Verstärkung K_p .

Wichtige Eigenschaften:

- Aufgrund des fehlenden Zeitverhaltens reagiert er sofort. (sehr gute Dynamik)
- Bleibende Regelabweichung. (schlechte Genauigkeit)

Integral Glied (I-Glied): Zeitliche Integration der Regelabweichung, bis keine mehr vorhanden ist. Das Integral Glied speichert den alten Regelfehler und zählt den aktuellen hinzu, welcher mit K_i gewichtet wird. Bei einem konstanten Eingangssignal ergibt sich daher eine lineare Steigung mit K_i .

Wichtige Eigenschaften:

- Langsame Reaktion
- Keine bleibende Regelabweichung (Verbesserung der stationären Eigenschaft)

Differential Glied (D-Glied): Ist ein Differenzierer, der nur auf die Änderungsgeschwindigkeit der Regelabweichung, nicht aber auf die Höhe reagiert. Durch einen Sollwert-Sprung würde sich durch die unendlich große Änderungsgeschwindigkeit ein Ausgangssignal mit unendlicher Amplitude ergeben. Durch die Verzögerungszeit der Hardware lässt sich ein D-Glied nicht ideal realisieren. Eine Realisierung kann daher über eine Approximation durch ein DT1-Glied erfolgen (Tiefpass gefiltertes D-Glied).

PT1-Glied: Das PT1 Glied wird meist als Filter (Tiefpassfilter) vorgeschaltet. Durch das PT1 Glied kann ein verrauschtes Eingangssignal geglättet (abgeflacht) werden, indem die hohen Änderungsgeschwindigkeiten unterdrückt werden.

Wind up Effekt:

Bei realen Systemen kommt es aufgrund von physikalischen Gegebenheiten meistens zu einer Begrenzung der Stellgröße z.B. kann die Kraftstoffmenge nicht beliebig groß werden oder der Ladedruck ist durch den Öffnungswinkel des Wastegate begrenzt. Durch diese Begrenzung kommt es bei Regelkreisen mit Integrieren (I-Anteil) vor, dass das Phänomen Wind up Effekt

auftritt. Aufgrund der Begrenzung der Stellgröße wird das dynamische Verhalten des Regelkreises verschlechtert (siehe Abbildung 2.7 a). Anhand von Abbildung 2.7 b) ist die Ursache für die Verschlechterung des Regelverhaltens zu erkennen. Der Regelfehler verringert sich ab dem Zeitpunkt ($t > 0,2$ s) und ändert nach ca. 2,5 Sekunden sein Vorzeichen (siehe obere Abbildung 2.7 b). Das heißt, der Istwert hat den Sollwert erreicht und überschreitet diesen jetzt. Der Regler sollte dem Überschwingen nun entgegenwirken. In Abbildung 2.7 b (unten) sieht man die Reglerausgangsgröße (Stellgröße) vor und nach der Stellgrößenbegrenzung ($\pm 0,3$). Die Stellgröße (in diesem Beispiel ein I-Anteil) besitzt nun vor der Begrenzung einen sehr großen Wert (bei $t = 2,5$). Durch die Begrenzung ist die Stellgröße auf 0,3 beschränkt und nimmt erst wieder ab, wenn die Begrenzung unterschritten wird, d.h. für $t > 4,5$ s. Der Regler integriert offensichtlich weiter, obwohl die Stellgröße den maximalen Wert ($\pm 0,3$) bereits erreicht hat. Das unnötige Anwachsen der Stellgröße vor der Begrenzung (Reglerausgangsgröße) wird Windup Effekt genannt („Aufwickeln des Integrierers“) (siehe Horn und Dourdoumas [18]).

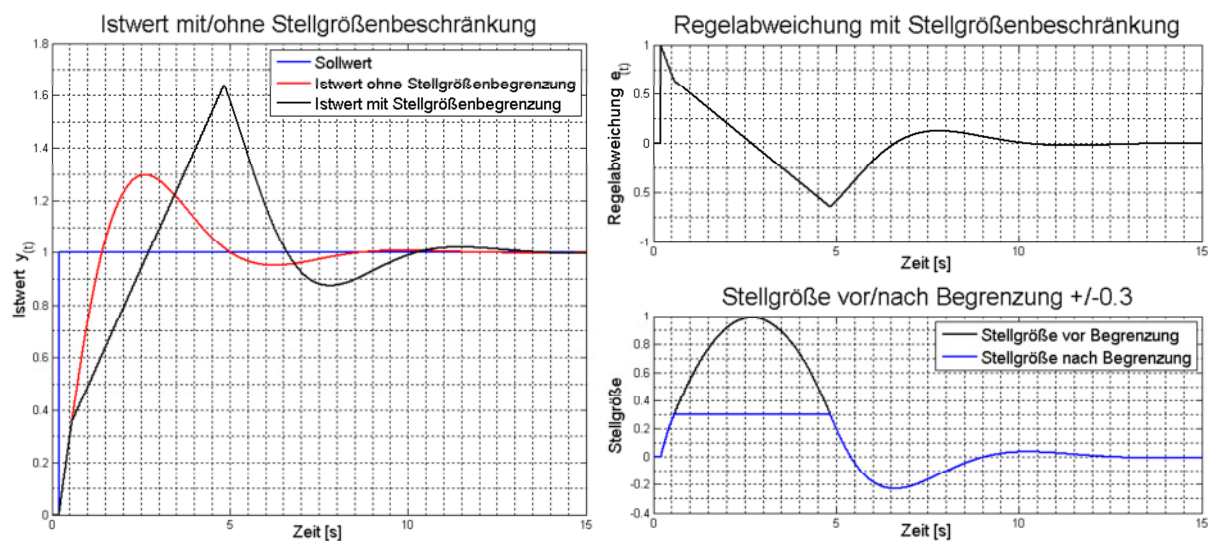


Abbildung 2.7: a.) Stellgrößenbeschränkung b.) Regelabweichung

Das Ziel der Anti Windup Maßnahme ist es, der Integration durch den Regler entgegenzuwirken, wenn die Reglerausgangsgröße die Stellgrößenbeschränkung erreicht. Eine einfache Anti Windup Maßnahme stellt das Einfrieren des I-Anteils dar (siehe untere Abbildung 2.8 b). Die blaue Linie der unteren Abbildung 2.8 b) zeigt die Reglerausgangsgröße (I-Anteil) und wie sie durch die Anti Windup Maßnahme bei der Stellgrößenbeschränkung ($\pm 0,3$) einfriert. Wenn die Regelabweichung positiv wird (Istwert schwingt über) bei $t = 2,5$ s, muss die Stellgröße nur mehr einen Wert von 0,3 abbauen (siehe Horn und Dourdoumas [18]).

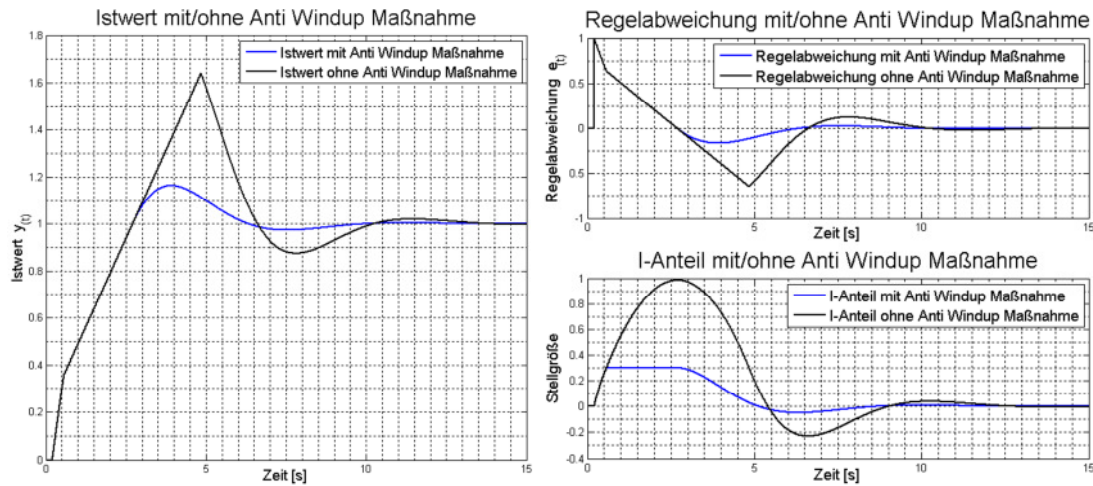


Abbildung 2.8: a.) Istwert

b.) Anti Windup Maßnahme

Durch die Anti Windup Maßnahme bekommt der Regelkreis ein deutlich besseres dynamisches Verhalten als ohne Maßnahme (Abbildung 2.8 a).

2.2.2 Sprungantwort

Die Sprungantwort beschreibt die Reaktion des Systems auf eine sprungförmige Änderung des Eingangssignals. Um einen Regelkreis zu bewerten, wird oft auf diese zurückgegriffen. Hierbei kann direkt auf das Verhalten des Systems geschlossen werden (z.B.: zu viel Überschwingen).

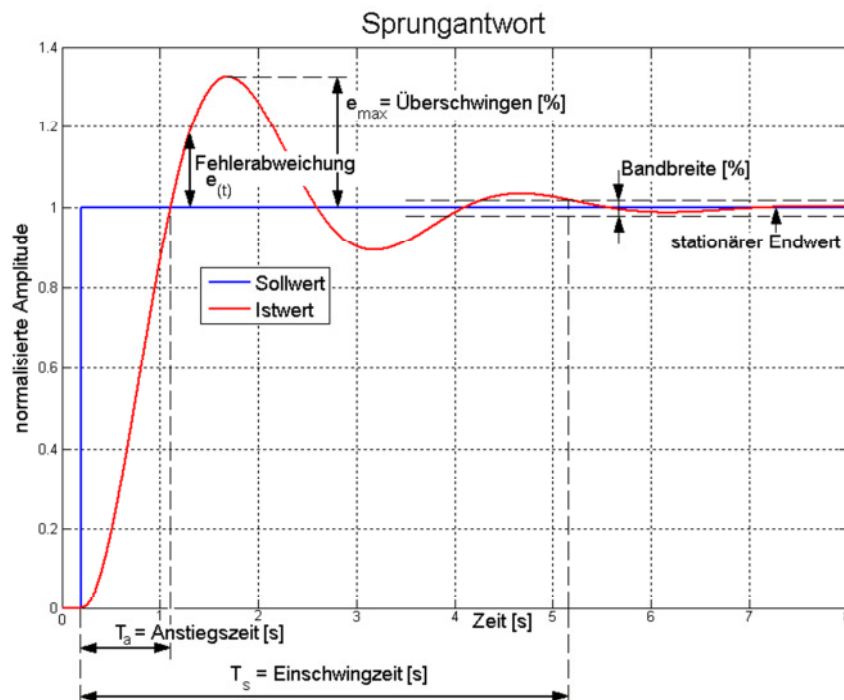


Abbildung 2.9: Sprungantwort

Folgende Kenngrößen werden dabei wie folgt definiert, um das zeitliche Einschwingverhalten zu beurteilen (siehe Föllinger [13]).

- 1.) Sollwert: blaue Linie
- 2.) Istwert: rote Linie
- 3.) Anstiegszeit T_a : Zeit, die vergeht, bis der Sollwert erstmals erreicht wurde. Die Anstiegszeit ist daher ein Maß für die Schnelligkeit.
- 4.) Einschwingzeit T_s : Zeit, bis der Istwert gleich dem Sollwert (+/- einer vorgegebenen Bandbreite) ist.
- 5.) Überschwingen e_{\max} : Ist die maximale Abweichung, nachdem der Sollwert bereits einmal erreicht wurde.
- 6.) Regelabweichung $e_{(t)}$: Abweichung des Istwerts vom Sollwert (Sollwert - Istwert)

Die Anstiegszeit T_a ist ein Indiz dafür, wie schnell ein Regler seinen Sollwert erreicht. Ein schneller linearer Regler hat aber meistens den Nachteil, dass er ein großes Überschwingen mit sich bringt und daher länger braucht, um einzuschwingen (große Einschwingzeit). Die bleibende Regelabweichung (Genauigkeit) beschreibt das statische Verhalten und ist die Abweichung des stationären Endwerts des Systems zum Sollwert. Um als Bewertungskriterium zu fungieren, wird die Regelabweichung je nach Anwendungsfall in eine andere Gütefunktion (z.B. Fehlerquadratsumme, usw.) eingesetzt (siehe Föllinger [13]).

2.2.3 Nichtlinearitäten

Im betrachteten Gesamtsystem treten mehrere Nichtlinearitäten an verschiedenen Stellen auf. Der Öffnungswinkel des Wastegate Turboladers stellt zum Beispiel eine Stellgrößenbegrenzung (Sättigung) dar, weil er mechanische Grenzen aufweist. Er lässt sich nicht mehr als 100% öffnen und auch nicht mehr als 0% schließen.

Das pneumatische Stellglied des Wastegate Turboladers ist ein nichtlinearer Aktuator, welcher nachgebildet werden muss, um den Einfluss der Nichtlinearität zu berücksichtigen. Das Motormodell (Strecke) selbst ist durch komplexe Verbrennungsprozesse und Abhängigkeiten zu den Umwelteinflüssen stark nichtlinear.

Die theoretische Stabilitätsuntersuchung von Regelkreisen kann aufgrund dieser Nichtlinearitäten sehr aufwendig werden. Deshalb können bei der Reglerkalibrierung immer nur bestimmte Be-

triebsbereiche des nichtlinearen Gesamtsystems betrachtet werden. Sollten die dynamischen Abweichungen jedoch über den Betriebsbereich hinausgehen, so muss die gesamte Nichtlinearität mit einbezogen werden.

2.2.4 Reglerkalibrierung

Die bisherigen Betrachtungen bauen auf der Theorie der linearen Regelungstechnik auf. In der Realität sind aber die meisten Vorgänge nichtlinear. Um die lineare Theorie dennoch anwenden zu können, wird das nichtlineare Verhalten in lineare Teilsysteme zerlegt. Für jedes dieser Teilsysteme gibt es eigene Parameter, die das lineare Teilsystem regeln. Bei der Optimierung von Reglerparametern einer Ladedruckregelung haben sich 3 Standardverfahren, welche sich in der Vorgehensweise unterscheiden, bewährt.

- Ziegler Nichols: Ist eine heuristische Methode zur Bestimmung von Reglerparametern (siehe Horn und Dourdoumas [18]).
- Auto Relay tuning: Als Grundlage hierfür dient ein Ein/Aus Schalter (Relay) (siehe Åström und Hägglund [7]).
- Linear quadratic Regulator (kurz LQR): Das Motormodell wird hierbei durch ein mathematisches Zustandsraummodell approximiert und mit Hilfe der Riccati Gleichung gelöst (siehe Lunze [22]).

2.3 Testmotor

Als Testmotor diene ein moderner 4 Takt Dieselmotor mit 4 Zylindern und 2 Liter Hubraum. Er hat ein maximales Drehmoment von 300Nm und eine maximale Leistung von 96kW (130PS). Laut neuen EU-Richtlinien muss dieser die Abgasnorm Diesel Euro 5+ erfüllen. Das Fahrzeug ist mit einem Bypassventil (Wastegate) Turbolader (siehe Kapitel 2.1.3.1) ausgestattet, welches über eine pneumatische Unterdruckdose als Stellglied angesteuert wird.

2.4 Motorsteuergerät ECU

Das Motorsteuergerät ist seit seiner Einführung in den 80iger Jahren aus keinem Automobil mehr wegzudenken. Die Funktionalität dieser Steuereinheit wurde dabei kontinuierlich gesteigert.

Diese Einheit greift in den Verbrennungsprozess des Motors ein, um die Emissionen besser zu regeln und die Grenzwerte einzuhalten. Ebenso läuft der Motor ruhiger und der Komfort des Fahrers erhöht sich. Das elektronische Motorsteuergerät (ECU: Engine Control Unit) misst dabei über die verschiedensten Sensoren den Zustand des Automobils. Die Sensoren übermitteln die elektronischen Signale an die ECU, welche diese dann auswertet, verarbeitet und gegebenenfalls neue Stellgrößen berechnet und an die jeweiligen Aktuatoren weiterleitet.

Die Steuereinheit beinhaltet eine Vielzahl von mathematischen Funktionen, mehrere tausend Kennfelder, Kurven und über 12.000 Konstanten, welche alle über eine logische Struktur miteinander verknüpft sind. Sie beinhaltet mehrere Regler u.a. für die Drehzahlregelung, Abgasrückführregelung (AGR) und Fahrgeschwindigkeitsregelung. Das Stellglied des Ladedruckreglers wird ebenfalls über kalibrierbare Parameter der ECU und einer logischen Datenverarbeitung eingestellt. Die Hauptaufgabe der ECU besteht darin, die Stellgrößen des Verbrennungsmotors über die entsprechenden Stellglieder zu steuern bzw. zu regeln (siehe Bosch [30]).

2.4.1 Ladedruckregler

Die Ladedruckregelung hat die Aufgabe, den Ladedruck im Hinblick auf eine saubere Verbrennung und einen günstigen Verbrauch einzustellen. Ebenso soll der Ladedruck zum Schutz des Motors und des Laders begrenzt werden. Die Ladedruckregelung ist in der Gesamtstruktur im Luftpfadsystem angesiedelt (siehe Abbildung 2.2).

Bei digitalen Algorithmen werden die aus Kapitel 2.2 kontinuierlichen Übertragungsfunktionen in diskrete z -Übertragungsfunktionen umgewandelt. In dem Buch „Regelungstechnik“ von Horn und Dourdoumas [18] werden diese Grundlagen genauer behandelt. Da es sich bei der ECU ebenfalls um eine digitale Einheit handelt, müssen die Signale auch dementsprechend digitalisiert werden. Dies erfolgt bei der Ladedruckregelung zeitsynchron mit einer Abtastrate von 20ms.

Die Ladedruckregelung besteht aus der Parallelschaltung von einem P-, I- und DT1-Glied (D-Glied) (siehe Kapitel 2.2.1). Je nachdem, in welchem Betriebsbereich sich das Fahrzeug befindet, werden die Parameter des Reglers umgeschaltet, um das nichtlineare Verhalten der Strecke regeln zu können.

Der Ladedrucksollwert (Führungsgröße) wird aus einem Drehzahl/Last (Betriebspunkt) abhängigem Basiskennfeld berechnet. Das Reglerausgangssignal wird in ein pulsweitenmoduliertes (PWM) Signal umgewandelt und über eine Leistungsendstufe dem Stellglied (pneumatische Druckdose) übermittelt. Dieses Signal steuert das Magnetventil des Stellgliedes und damit die

Abgaszufuhr für die Turbine des Verdichters. Der gemessene Druck hinter dem Ladeluftkühler (siehe Abbildung 2.2) dient als Istwert (Regelgröße) und wird zur ECU zurückgeführt (Vergleiche Abbildung 2.6: Standard Regelkreis).

2.4.2 Zusätzliche ECU Funktionen

Die ECU beinhaltet des Weiteren einige zusätzliche Funktionen, auf die kurz eingegangen wird.

Aktuator Linearisierung: Durch diese nichtlineare Kennlinie wird das Tastverhältnis (PWM Signal) in eine linearere Aktuator Sollposition umgewandelt.

Anti Wind up: Wie bereits in Kapitel 2.2.1 erwähnt, friert das Anti Wind up den I-Anteil des Reglers ein, um in Folge der Stellgrößenbegrenzung den Regelfehler zu beschränken.

Zusätzliche Kennfelder: Je nachdem, in welchem Modus der Motor betrieben wird (normal mode, eco mode, regeneration mode) wird auf andere Kennfelder umgeschaltet.

Groß- und Kleinsignal Umschaltung: Sollte die Regelabweichung einen gewissen Schwellwert über- bzw. unterschreiten, so werden andere Reglerparameter verwendet. Dadurch kann auf das dynamische Verhalten der Strecke Einfluss genommen werden.

Transienterkennung: Die Transienterkennung erfolgt anhand der Anstiegsgeschwindigkeit des Drehmoments. Bei einer zu hohen Anstiegsgeschwindigkeit schaltet die ECU auf andere Reglerparameter um.

Anpassungsfaktoren: Es gibt noch eine Vielzahl von Anpassungsfaktoren, die Umwelteinflüsse wie zum Beispiel Außendruck, Außentemperatur berücksichtigen, aber auch Faktoren, die von der Temperatur des Motors abhängig sind.

Gain Scheduling: Um die lineare Regelungstheorie anwenden zu können, wird das nichtlineare Verhalten in lineare Teilsysteme zerlegt. Die Umschaltung zwischen den verschiedenen Reglerparametern für die Teilsysteme erfolgt mithilfe des Gain Scheduling.

2.5 Global dynamisches Motormodell

Ziel der Motormodellbildung ist es, die dynamischen Effekte des Verbrennungsmotors in geeigneter Weise zu beschreiben. Vor allem die dynamischen Effekte durch den Gastransport und die chemischen Reaktionen beeinflussen die Ausgangsgrößen (Drehmoment, Kraftstoffverbrauch, Abgasbildung, usw.) in einer Komplexität, die sich nicht mehr analytisch genau genug beschrei-

ben lässt. Daher wird in dieser Arbeit ein empirisches Modellkonzept anstelle eines physikalischen Modells verwendet. Diese bringen aufgrund der kompakteren Struktur und Handhabbarkeit Vorteile in der Qualität und Rechengeschwindigkeit mit sich. Bei dieser Methode werden mehrere mathematische Funktionen verknüpft, um so das Verhalten der vermessenen Strecke (Motor) zu reproduzieren (siehe Nebel *et al.* [24]).

Für die empirischen Ansätze werden künstliche neuronale Netze (KNN) eingesetzt. In der Motormodellierung haben sich vor allem die Fast Neural Networks (FNN) und die Intelligent Neural Networks (INN) durchgesetzt, die auch in den folgenden Kapiteln näher beschrieben werden (siehe Nebel *et al.* [24]).

2.5.1 Künstliche neuronale Netze

Künstliche neuronale Netze (KNN) sind dem menschlichen Gehirn nachempfunden. Dort sind mehrere Neuronen zu einem Netz miteinander verknüpft. Ein künstliches Netz besteht aus mehreren Funktionen (Neuronen). Ziel eines KNN ist es, durch Zusammenschaltung und Gewichtung der einzelnen Neuronen ein gewünschtes Systemverhalten nachzubilden.

Abbildung 2.10 a) zeigt ein solches Neuron. Diese Funktion ist abhängig von den Eingängen x_1 bis x_n und deren Gewichten w_1 bis w_n . Um dynamische Vorgänge abbilden zu können, braucht man noch eine Rückführung des Ausgangs zum Eingang (siehe Abbildung 2.10 b) (siehe Rojas [31]).

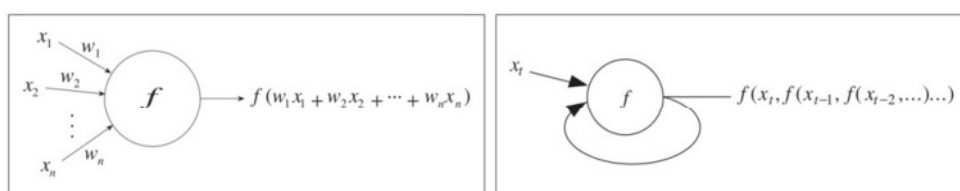


Abbildung 2.10: a.) Neuron b.) dynamisches Neuron

Quelle: Rojas [31]

Mehrere dieser Neuronen sind durch arithmetische Operationen miteinander verknüpft und bilden das neuronale Netz. Den prinzipiellen Verlauf, wie die Gewichtsparameter des neuronalen Netzes bestimmt werden, zeigt Abbildung 2.11.

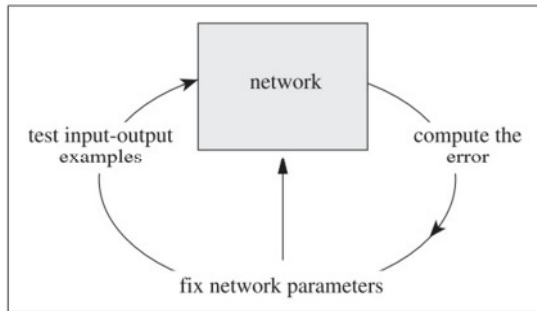


Abbildung 2.11: Prinzip von Neuronalen Netzen

Quelle: Rojas [31]

Um die benötigten Ausgangsmessdaten durch Sensoren zu erfassen, wird eine geeignete Testsequenz für die Eingangsgrößen generiert und an den realen Motor übertragen. Die Testsequenz dient zum Lernen des künstlichen neuronalen Netzes (KNN), indem die Ausgangsdaten mit den Messdaten verglichen werden. Anhand von Abweichungen berechnet ein Algorithmus neue Gewichtparameter für die einzelnen Neuronen in dem KNN. Mit den neu eingestellten Werten beginnt ein neuer Durchlauf, indem die Testsequenz wieder in das Netz geschickt und der Ausgang mit den Messdaten verglichen wird. Dies wird so lange fortgeführt, bis ein bestimmtes Abbruchkriterium (z.B. Abweichung klein genug) erfüllt ist (siehe Rojas [31]).

Die Unterteilung des gesamten Motorbetriebsbereichs bezüglich Drehzahl/Last in mehrere lokale Teilmodelle (siehe Abbildung 2.12 a) bringt gewisse Vorteile mit sich. Diese nichtlinearen Teilmodelle werden mit Hilfe einer Gültigkeitsfunktion (siehe Abbildung 2.12 b) zu einem einzigen globalen dynamischen Motormodell überlagert, welches den gesamten Betriebsbereich abdeckt. Dies hat den Vorteil, dass die Teilmodelle danach keine scharfen Übergänge mehr zueinander haben. Die Teilmodelle werden mit verschiedenen Testsequenzen, welche in den jeweiligen lokalen Bereichen gültig sind, trainiert (siehe Nebel *et al.* [24]).

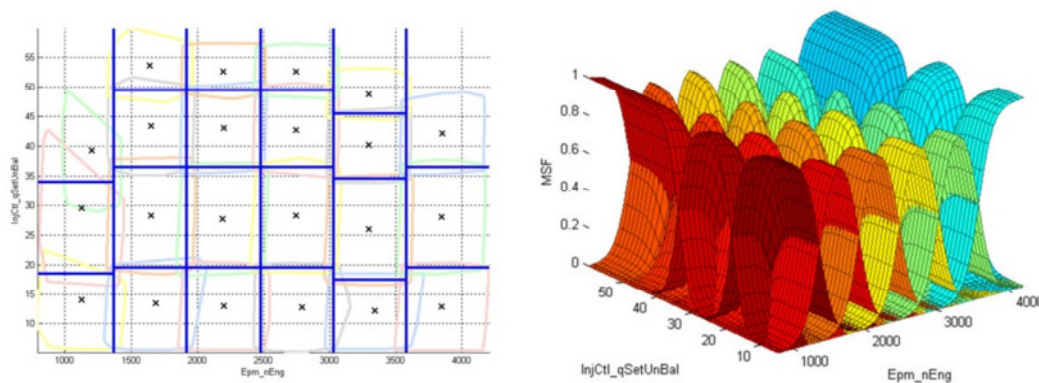


Abbildung 2.12: a.) lokale Teilmodelle b.) Gültigkeitsfunktion

Quelle: Nebel *et al.* [24]

Diese nichtlinearen lokalen Teilmodelle sind dINN (dynamic Intelligent Neural Networks). Die dINN gehören zu der Klasse der INN (Intelligent Neural Networks), welche eine Erweiterung der FNN (Fast Neural Networks) darstellen. Die im nächsten Kapitel behandelte LOLIMOT (Local Linear Model Tree) Methode gehört ebenfalls zu den FNN.

2.5.2 LOLIMOT (Local Linear Model Tree)

Wie schon vorher erwähnt, sind die nichtlinearen lokalen Teilmodelle vom Typ der dINN. Als Grundlage der dINN dient die Methode des LOLIMOT. Dabei wird durch mehrere stückweise lineare Teilmodelle eine nichtlineare Funktion approximiert (Siehe Abbildung 2.13 a) (vergleiche Nelles [26], [27]).

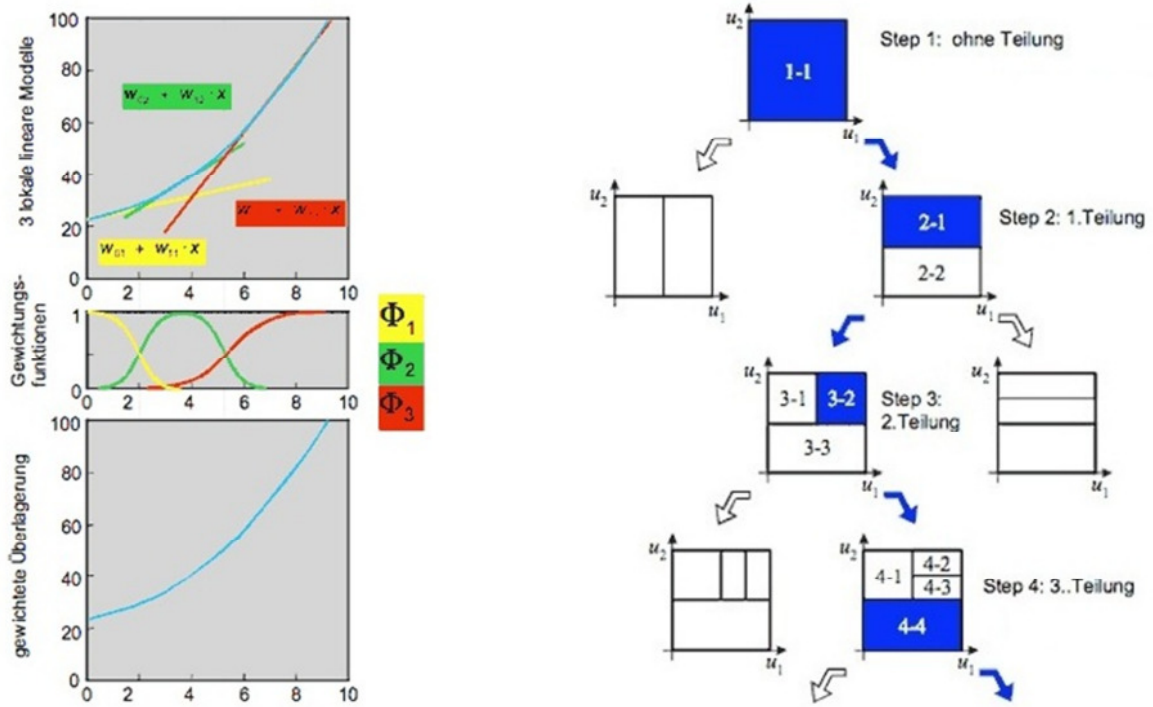


Abbildung 2.13: a.) Approximation der nichtlineare Funktion

b.) Teilung

Quelle: Nelles [26]

Das Prinzip des LOLIMOT (siehe Nelles [26]) sieht wie folgt aus:

- 1.) Im gesamten Eingangsraum wird die gesuchte nichtlineare Funktion durch eine einfache lineare Funktion angenähert. Dies geschieht mit Hilfe einer Parameteroptimierung z.B. durch die Least Square Methode.
- 2.) Ist die Annäherung nicht ausreichend genau, wird der Eingangsraum orthogonal zu den Achsen in 2 Hälften geteilt (siehe Abbildung 2.13 b).
- 3.) Es werden nun alle Teilräume des Eingangsraumes getestet. Der Teilraum mit der größten Abweichung von der gesuchten Funktion wird ausgewählt.
- 4.) In diesem neuen Raum wird wieder eine neue einfache lineare Funktion zu der nichtlinearen Kurve approximiert. Mit Hilfe einer normierten Gaußschen Glockenkurve wird die Gültigkeit des Teilmodells innerhalb des Eingangsraumes beschrieben.
- 5.) Wurde eine geeignete lineare Funktion gefunden, so wird sie dem Modellbaum (engl. Model Tree) hinzugefügt.
- 6.) Sollte die Approximation des Ausgangs an die gesuchte Kurve noch nicht genau genug sein, so wird der gewählte Teilraum wieder geteilt und der Algorithmus beginnt wieder bei 3.

Ist aber eine bestimmte Anzahl von linearen lokalen Teilfunktionen erreicht oder die Abweichung zu der gesuchten Kurve klein genug, so ist das Netz fertig trainiert.

Das INN ist eine Erweiterung des LOLIMOTs (FNN), indem die Teilung nicht mehr nur orthogonal zu den Achsen gemacht wird, sondern auch schräge Teilungen möglich sind (siehe Keuth [19]).

2.5.3 dINN (dynamic Intelligent Neural Network)

Um die Dynamik des Motors zu berücksichtigen, wird bei den dINN der Ausgang zurückgeführt (siehe Abbildung 2.14). Anstelle der statischen LOLIMOT Eingänge x_1, x_2, \dots, x_n werden nun die zeitdiskreten physikalischen Eingänge $u_1(k), \dots, u_q(k)$ und der rückgekoppelte Ausgang dem neuronalen Netz zugeführt (siehe Hametner [17]).

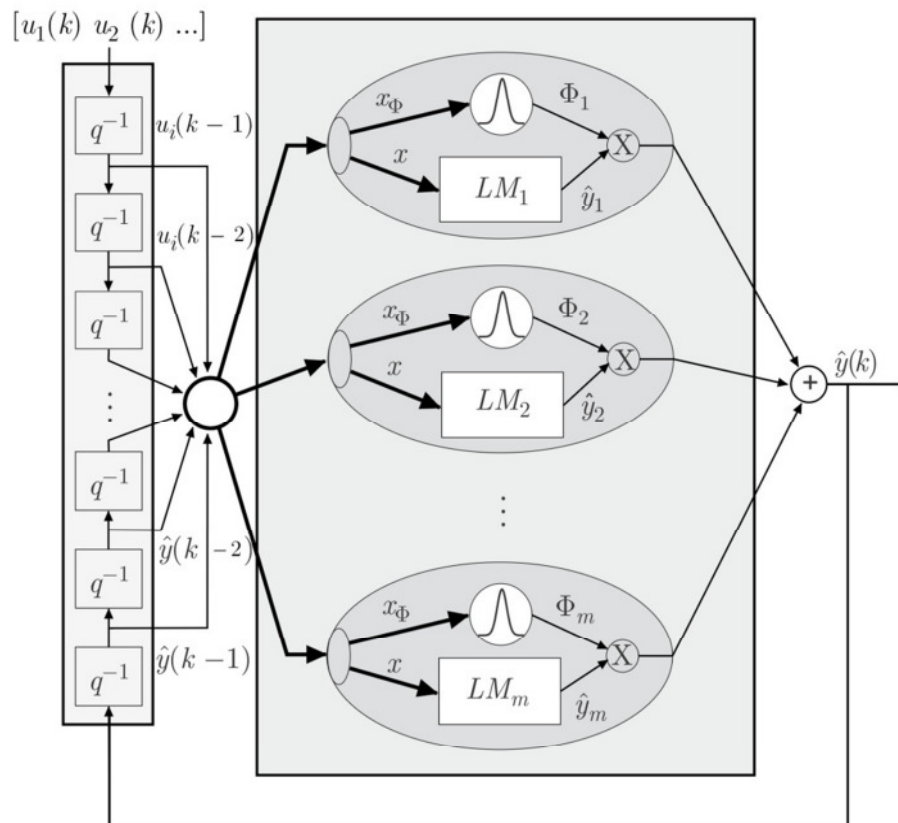


Abbildung 2.14: Baumstruktur dINN

Quelle: Hametner [17]

Das grau hinterlegte Rechteck zeigt die Struktur des statischen neuronalen Netzes nach der LOLIMOT Methode. Durch die Möglichkeit der schrägen Teilung des Eingangsraums wird das neuronale Netz zu einem INN (Intelligent Neural Network) und durch die Rückführung wird es zu einem dINN (dynamic Intelligent Neural Network).

Die Berechnung des statischen LOLIMOT Modellausgangs \hat{y} erfolgt mit Hilfe folgender Formel (siehe Nelles [26]):

$$\hat{y} = \sum_{i=1}^M (w_{0,i} + w_{1,i} \cdot x_1 + \dots + w_{n,i} \cdot x_n) \cdot \Phi_i(x, c_i, \sigma_i) \quad (2.1)$$

M	<i>Teilmodelle</i>
n	<i>Eingänge</i>
x_1, \dots, x_n	<i>LOLIMOT Eingangsgrößen</i>
$w_{0,i}, \dots, w_{n,i}$	<i>Parameter des i – ten linearen Teilmodells</i>
Φ_i	<i>normierte Gaußsche Glockenfkt. des i – ten Teilmodells</i>

Der Modellausgang ist die Summe aller lokalen Modelle ($LM_1 \dots LM_m$), welche mit der normierten Gaußschen Glockenfunktion Φ_i gewichtet und für jedes Teilmodell wie folgt berechnet wird (siehe Nelles [26]).

$$\Phi_i(x, c_i, \sigma_i) = \frac{z_i}{\sum_{j=1}^M z_j} \quad (2.2)$$

$$\text{mit } z_j = e^{\left(-\frac{1}{2} \cdot \left(\frac{(x_1 - c_{1j})^2}{\sigma_{1,j}^2} + \frac{(x_2 - c_{2j})^2}{\sigma_{2,j}^2} + \dots + \frac{(x_n - c_{nj})^2}{\sigma_{n,j}^2} \right) \right)}$$

c_j ... *Zentrumskoordinaten*

σ_j ... *Standardabweichung*

Um die dynamischen Effekte des Verbrennungsmotors abbilden zu können, wird eine Rückführung des LOLIMOT Modellausgangs \hat{y} benötigen. Als Eingangsgröße des statischen LOLIMOT Modells (Formel 2.1) wird nun folgender Eingangsvektor verwendet.

$$x_{(k)}^T = \left[\begin{array}{l} u_1(k), u_2(k), \dots, u_q(k) \\ u_i(k-1), u_i(k-2), \dots, u_i(k-m) \\ \hat{y}_{(k-1)}, \hat{y}_{(k-2)}, \dots, \hat{y}_{(k-n)} \end{array} \right] \quad (2.3)$$

$u_1(k), u_2(k), \dots, u_q(k)$	<i>physikalische Eingänge</i>
q	<i>Anzahl der physikalischen Eingänge</i>
$u_i(k-1), u_i(k-2), \dots, u_i(k-m)$	<i>zeitversetzter physikalischer i – ter Eingang</i>
$\hat{y}_{(k-1)}, \hat{y}_{(k-2)}, \dots, \hat{y}_{(k-n)}$	<i>zeitversetzter Modellausgang</i>

Man sieht also, dass der dynamic Intelligent Neural Network (dINN) Modellausgang $\hat{y}_{(k)}$ abhängig ist von den physikalischen Eingängen $u_1(k), \dots, u_q(k)$, deren zeitversetzten Eingängen $u_i(k-1), \dots, u_i(k-m)$ und der zeitversetzten Modellausgängen $y_{(k-1)}, \dots, y_{(k-n)}$ durch die Rückführung.

2.5.4 Dynamische Identifikation

Zur dynamischen Identifikation von Single Input Single Output (kurz: SISO) Modellen wird häufig ein APRB (engl. Amplitude-modulated Pseudo Random Binary) Signal verwendet. Das APRB Signal ist dabei eine sprunghafte pseudo stochastische dynamische Erregung mit unterschiedlichen Haltezeiten und Amplituden (Siehe Abbildung 2.15).

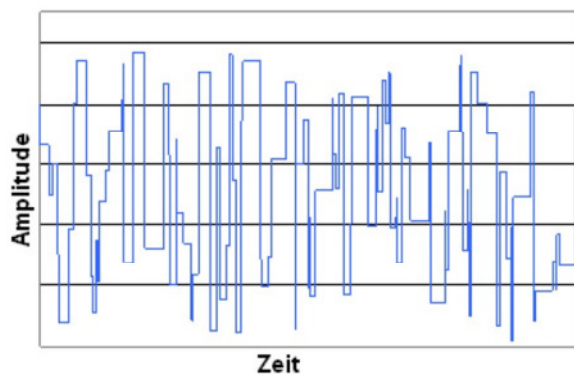


Abbildung 2.15: APRBS Signal

Quelle: Nebel [25]

In der wissenschaftlichen Arbeit [25] von Nebel wird dieses Signal ebenfalls für Multi Input Single Output (kurz: MISO) Modelle erfolgreich angewendet. Für jede Eingangsgröße und jedes lokale Modell wird ein dafür geeignetes ARBP Signal zum Trainieren des dINN generiert.

3 Modellbasierte Regleroptimierung

In diesem Kapitel geht es um den theoretischen Hauptteil dieser wissenschaftlichen Arbeit. Hierzu zählt die richtige Auswahl des Fahrmanövers für die Optimierung, die detaillierte Zielsetzung und die Beschreibung des ausgewählten Optimierungsalgorithmus.

3.1 Fahrmanöver

Bei der traditionellen Reglerkalibrierung wird der Ladedruckregler zuerst kalibriert und danach werden die eingestellten Parameter validiert. Dies erfolgt, indem verschiedene Fahrmanöver abgefahren werden und eine Beurteilung der unterschiedlichen Kriterien (z.B. Überschwingen, Anstiegszeit, usw.) für die eingestellten Parameter stattfindet. Die Fahrzyklen sind entweder kundenspezifische oder durch Richtlinien für eine Abnahmeprüfung vorgegeben. Einer dieser Tests besteht darin, das Fahrzeug bei Vollastbeschleunigung („TipIn“) (siehe Abbildung 3.1) aus verschiedenen Anfangszuständen der Motordrehzahl zu beschleunigen. Der Applikateur steigt hierfür zuerst vom Gas (0% Gaspedalstellung), um dann Vollgas (100% Gaspedalstellung) zu geben. Die folgenden Abbildung 3.1, 3.2, 3.3 stammen aus realen Messdaten und dem entsprechenden Verhalten des Fahrzeugs.

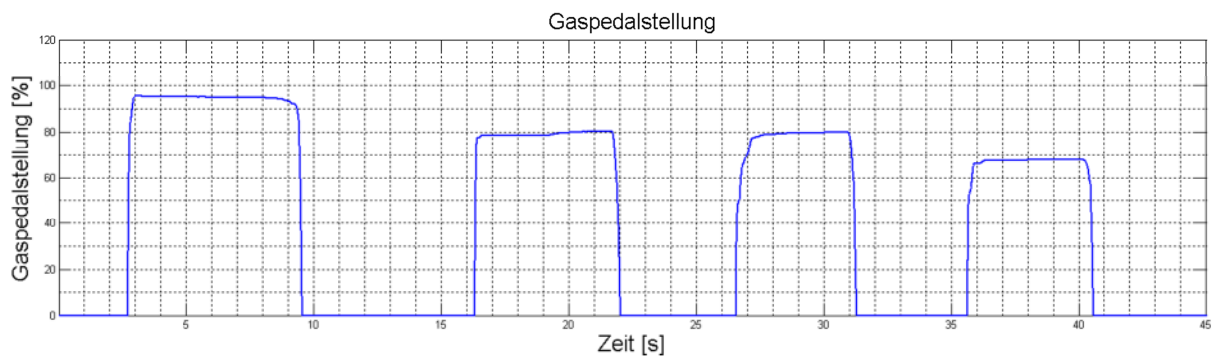


Abbildung 3.1: Fahrmanöver: Gaspedalstellung

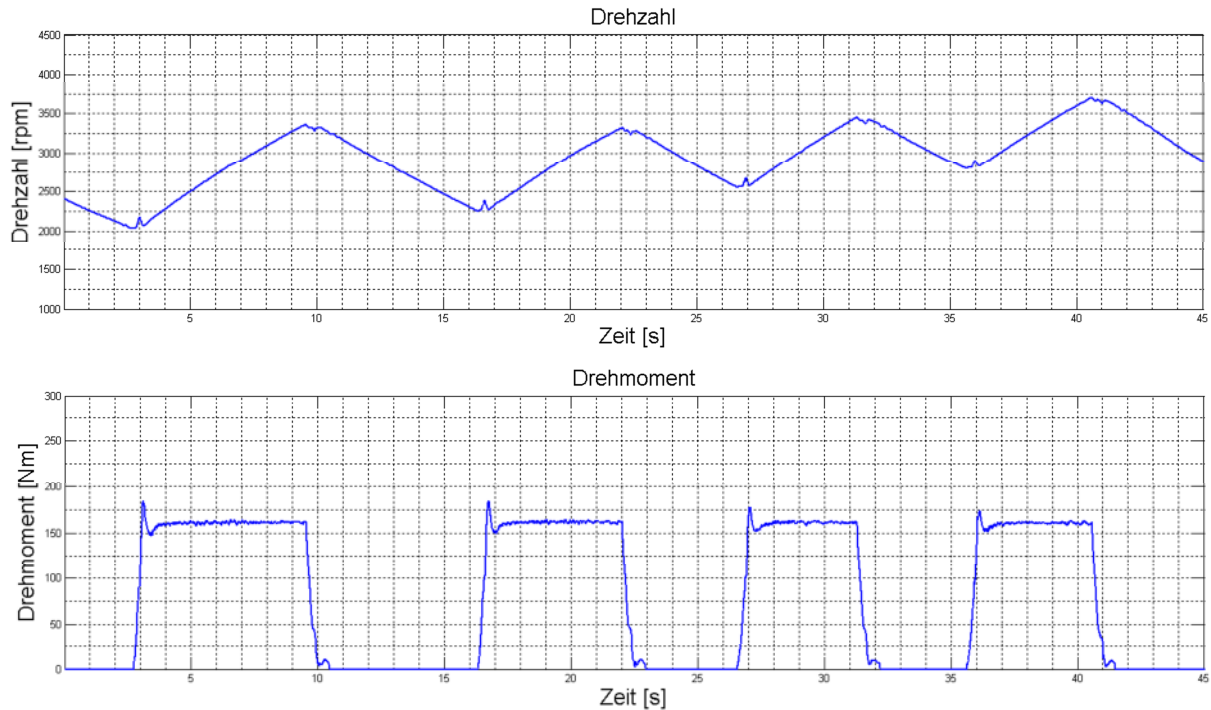


Abbildung 3.2: Fahrmanöver: Drehzahl/Drehmoment

In Abbildung 3.1 sieht man, dass vier solche TipIn's mit 98%, 80% und 70%iger Gaspedalstellung durchgeführt wurden. Das Verhalten des Fahrzeugs in Bezug auf Drehzahl und Drehmoment aufgrund der Gaspedalbetätigung lässt sich in Abbildung 3.2 erkennen. Man sieht, dass durch die Wahl dieses Fahrmanövers in einem Drehzahlbereich von 2000 bis 3700 Umdrehungen pro Minute (engl. revolutions per minute, kurz rpm) und einem Drehmoment bis 160 Newtonmeter (kurz Nm) optimiert wird.

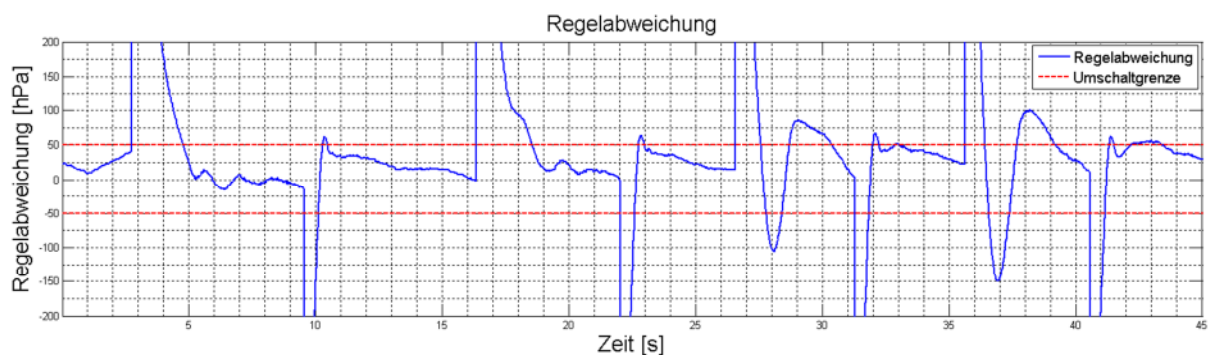


Abbildung 3.3: Fahrmanöver: Regelabweichung/Großsignalbereich

Die Regelabweichung des Ladedruckreglers mit den anfangs eingestellten Parametern ist in Abbildung 3.3 zusehen. Bei jedem sprunghaften Anstieg des Drehmoments steigt die Regelabweichung stark an und es erfolgt eine Umschaltung zwischen verschiedenen Parametersätzen (siehe

Kapitel 2.4.2 erweiterte ECU Funktionen). Um den Fahrkomfort/Verbrauch usw. zu verbessern, ist es nun notwendig, die Regelabweichung so klein wie möglich zu halten. Als erste zu minimierende Zielfunktion bietet sich daher die Minimierung der Fehlerquadratsumme (Formel 3.1) an. Diese Gütefunktion wird sehr oft in der Regelungstechnik angewendet.

$$ISE = \int_0^{T_{end}} e^2(t) dt \dots \text{Integral square error} \quad (3.1)$$

3.2 Zielsetzung

Der Kleinsignalbereich ist der Bereich, wo sich die Regelabweichung nur mehr innerhalb eines kleinen Bereichs um den Sollwert befindet (siehe Abbildung 3.4). Die Reglerparameter für den Kleinsignalbereich (Abbildung 3.3 und Abbildung 3.4) sind nach einer Arbeitspunktlinearisierung bereits mit Hilfe der Linear quadratic Regulator (kurz LQR) Methode (siehe Kapitel 2.2.4 Reglerkalibrierung) optimiert worden. Um aber auf das nichtlineare Verhalten der Strecke besser reagieren zu können, wird im Großsignalbereich auf andere Reglerparameter umgeschaltet. Diese Großsignalparameter besitzen einen größeren P-Anteil, um die Regeldifferenz schneller auszugleichen.

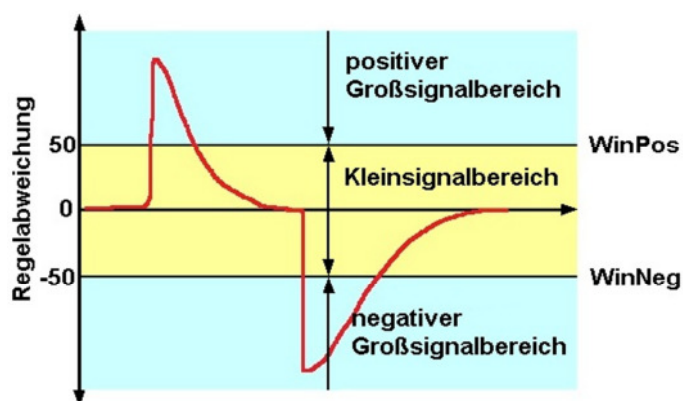


Abbildung 3.4: Kleinsignal-/ Großsignalbereich

Die Umschaltung auf die Großsignal Parameter ist eine interne Funktion der ECU und wird hier nicht näher behandelt. Sollte die Regelabweichung einen bestimmten Schwellwert (WinPos/ WinNeg) über- bzw. unterschreiten, so wird auf die positiven/ negativen PID Großsignal Reglerparameter umgeschaltet. Ziel ist es nun, diese Parameter mithilfe eines geeigneten Optimierungsverfahrens zu optimieren.

Die zu optimierenden Parameter sind nur im Großsignalbereich aktiv und daher wird darauf geachtet, ein Fahrmanöver zur Optimierung zu verwenden, bei dem die Regelabweichung oft und lange im Großsignalbereich ist. Der in Abbildung 3.2 gewählte Fahrzyklus hat eine starke Änderung des Drehmoments und aufgrund dessen auch eine hohe Regelabweichung. Der Bereich in dem die Großsignalparameter aktiv sind, fällt mit 2-3 Sekunden pro TipIn eher kurz aus (siehe Abbildung 3.3). Das heißt, in einem gesamten Fahrzyklus von 50 Sekunden sind jeweils die positiven und negativen Großsignalparameter ca. 8 Sekunden eingeschaltet.

Bei der Reglerauslegung und dem Fahrmanöver (siehe Abbildung 3.2) sei zu beachten, dass ein Überschwingen $> 5\%$ - 10% des Ladedruck Istwerts über den Ladedruck Sollwert zu vermeiden ist. Die Turboladerdrehzahl und der Zylinderdruck würden andernfalls im Volllastbetrieb zu groß werden. Überschwinger kleiner als 200 Hektopascal (kurz hPa; 1 Hektopascal = 1 Millibar) werden als annehmbar angesehen. Ladedruck Unterschwinger sind vor allem im Fahrverhalten spürbar und sollten kleiner als 100 hPa sein. Stärkere Unterschwinger fallen negativ im Fahrverhalten auf, da sie aufgrund von Einspritzmengeneingriffe (z.B. Begrenzung durch Rauchkennfeld), als Rucke des Fahrzeugs empfunden werden.

In Abbildung 3.5 ist zu sehen, wie Kruse *et al.* in ihrer wissenschaftlichen Arbeit [20] den Kompromiss zwischen Fahrkomfort und Dynamik behandeln.

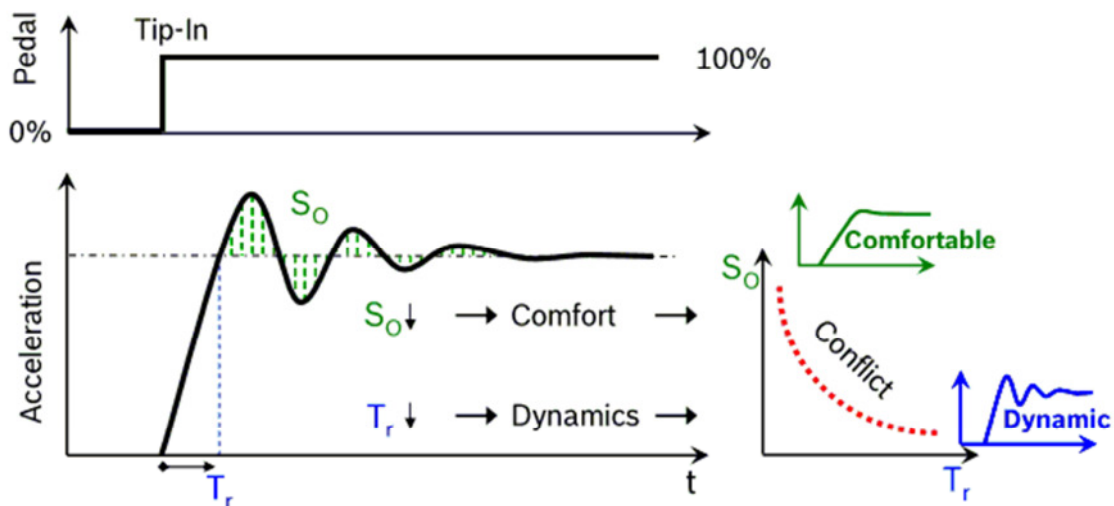


Abbildung 3.5: Multikriterielle Optimierung

Quelle: Kruse *et al.* [20]

Um die Anstiegszeit T_r zur besseren Dynamik und die Fläche S_0 der Über-/ Unterschwinger für den Fahrkomfort zu minimieren, wird hier ein multikriterieller Optimierer eingesetzt. Die Beurteilungskriterien der Sprungantwort (Kapitel 2.2.2), Fehlerabweichung, Anstiegszeit, Überschwingen und Einschwingzeit sind zwar anders definiert, können aber ebenfalls zur Minimierung heran

gezogen werden. Da es sich bei dem Fahrmanöver (Abbildung 3.2) um mehrere unterschiedliche Sprungantworten handelt, wird immer der Mittelwert der einzelnen Überschwinger, Anstiegszeiten und Einschwingzeiten verwendet. Die erste zu minimierende Zielfunktion (Formel 3.2) betrifft die Regelabweichung. Ziel der Regelung ist es, die Abweichung von Istwert zu Sollwert möglichst klein zu halten. Als zweite Zielfunktion (Formel 3.3) werden die Überschwingweiten minimiert, um die Nebenbedingungen (Überschwinger < 200 hPa und Unterschwinger < 100 hPa) einzubauen. Formel 3.4 und 3.5 sind die Zielfunktionen zur Minimierung der Anstiegszeit für eine bessere Dynamik und der Minimierung der Einschwingzeit für eine schnelle Erreichung des stationären Endwerts.

$$f_1 = ISE = \int_0^{T_{end}} e_{(t)}^2 dt \dots \text{Integral square error} \quad (3.2)$$

$$f_2 = \text{mean}(e_{max}) \quad (3.3)$$

$$f_3 = \text{mean}(T_a) \quad (3.4)$$

$$f_4 = \text{mean}(T_s) \quad (3.5)$$

Zielfunktion f_1 (Formel 3.2) hat den Charakter eines Hauptkriteriums. Wenn diese im Idealfall Null ist, dann ist der Istwert gleich dem Sollwert und die anderen Zielfunktionen sind ebenfalls optimal. In der Realität kommt dies aber nicht vor. Dort wird darauf geachtet das Überschwingen usw. möglichst minimal zu machen.

Da es nun darum geht, die Beurteilungskriterien (Formel 3.2 bis 3.5) zu minimieren, das System aus höchst komplexen dINN (dynamic Intelligent Neural Network) besteht und der Fahrzyklus als verrauschte Messdaten vorliegt, wird ein stochastischer multikriterieller Optimierungsalgorithmus eingesetzt.

3.3 Optimierungsalgorithmus

Derzeit verfügbare Optimierungsalgorithmen für die Kalibrierung von Reglerparametern zur Regelung von nichtlinearen, experimentell ermittelten Strecken, sind oft nicht ausreichend oder nicht vorhanden. Adäquate Methoden zur multikriteriellen Optimierung in der Kalibrierung fehlen ebenso. In Optimierungsverfahren mit nur einer Zielfunktion ist die optimale Lösung in der Regel klar definiert. Dies gilt aber nicht für die multikriterielle Optimierung. Das Optimierungsproblem wird durch die Tatsache gekennzeichnet, dass mehrere widersprüchliche Zielwerte (z.B.: Überschwingweite, Anstiegszeit, Einschwingzeit) gleichzeitig berücksichtigt werden müssen. Aus diesem Grund existiert eine Reihe von gleichwertigen Kompromisslösungen.

Klassische Methoden wie gradientenbasierte oder Simplex Algorithmen benötigen eine einzige Zielfunktion. Multikriterielle Optimierungsprobleme können daher nicht zweckmäßig genug mit klassischen Optimierungsverfahren gelöst werden.

Die evolutionären Algorithmen hingegen können leichter für eine multikriterielle Optimierung verwendet werden. Sie basieren auf dem Prinzip der biologischen Evolution. Hierbei werden verschiedene Parameter parallel innerhalb einer Population (= mehrere verschiedene Parametersätze) geändert und diese neu am Modell getestet. Danach wird entschieden, ob die alten oder neuen Parameter behalten werden. Die parallele Suche erlaubt den evolutionären Algorithmen mehrere optimale Lösungen in einem einzigen Optimierungsdurchlauf zu erfassen und sind daher speziell für die multikriterielle Optimierung geeignet [siehe Kruse *et al.* [20]].

3.3.1 Differential Evolution (DE)

Der Differential Evolution Algorithmus von Price *et al.* [29] ist ähnlich aufgebaut wie die evolutionären Algorithmen und arbeitet nach dem Grundprinzip der biologischen Evolution. Der DE ist leicht zu implementieren und auf neuere Probleme erweiterbar. Ebenso schneidet er bei realen numerischen Optimierungsproblemen sehr gut ab (siehe Das und Suganthan [9]) und wird deshalb auch hier als Optimierer verwendet.

3.3.2 Basiskonzept

Der klassische DE Algorithmus besteht aus den vier grundlegenden Schritten Initialisierung, Mutation, Crossover und schließlich Selektion. Abbildung 3.6 zeigt den Programmablauf für den gesamten Algorithmus (siehe Price *et al.* [29]).

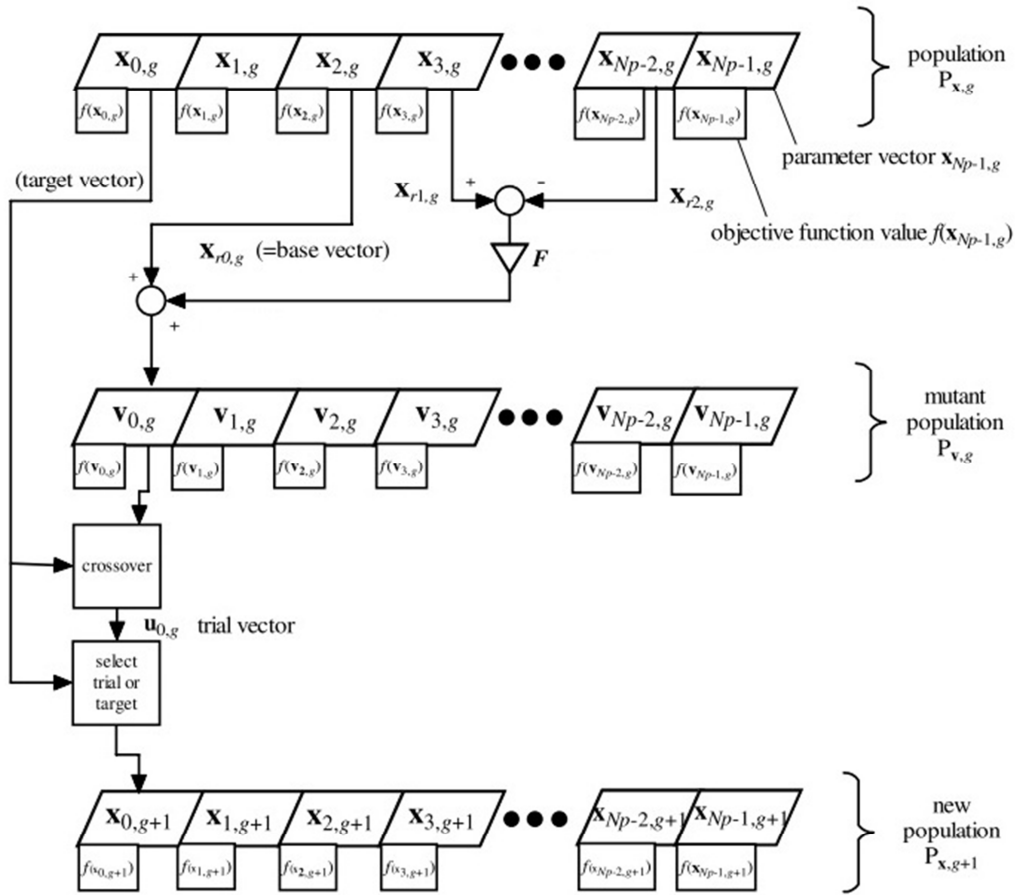


Abbildung 3.6: Programmablauf klassischer DE

Quelle: Price *et al.* [29]

Um den Programmablauf besser zu verstehen, werden folgende Notationen verwendet (siehe Price *et al.* [29]):

$$\begin{aligned}
 P_{x,g} &= (\vec{x}_{i,g}), & i &= 0,1, \dots, Np - 1, & g &= 0,1, \dots, g_{max} & (3.6) \\
 \vec{x}_{i,g} &= (x_{j,i,g}), & j &= 0,1, \dots, D - 1 \\
 \vec{v}_{i,g} &= (v_{j,i,g}), & j &= 0,1, \dots, D - 1 \\
 \vec{u}_{i,g} &= (u_{j,i,g}), & j &= 0,1, \dots, D - 1
 \end{aligned}$$

Dabei ist $P_{s,g}$ die aktuelle Population mit der Größe N_p , welche die Zielvektoren („target vector“) $x_{i,g}$ beinhaltet. Dieser Zielvektor ist D -dimensional und enthält die reellwertigen Optimierungsparameter. Je nach Differential Evolution Schema wird der Mutationsvektor $v_{i,g}$ generiert und mit einem Zielvektor $x_{i,g}$ rekombiniert, um einen neuen Nachkommenvektor („trial vector“) $u_{i,g}$ zu bekommen.

3.3.2.1 Initialisierung

Die Suche nach einem globalen Optimum beginnt als erstes mit einer zufällig generierten Population N_p für die erste Generation, wobei jeder Zielvektor $x_{i,0}$ dieser Population einen möglichen Kandidaten der Lösung bildet. Oft sind die Parameter in dem Zielvektor an bestimmte Grenzen gebunden und können daher begrenzt werden (zum Beispiel: Reglerparameter dürfen nicht negativ sein). Deshalb wird der Suchraum für die Parameter beschränkt und wie folgt initialisiert (siehe Price *et al.* [29]).

$$x_{j,i,0} = x_{j,min} + rand_{i,j} [0, 1] \cdot (x_{j,max} - x_{j,min}) \quad (3.7)$$

$$\text{mit } j = 0, 1, \dots, D - 1$$

$$i = 0, 1, \dots, N_p - 1$$

Dabei ist $rand_{i,j}$ eine gleichmäßig verteilte zufällige Zahl zwischen 0 und 1 und wird unabhängig von jedem anderen Vektor initialisiert. Zur besseren Erklärung des Algorithmus wird als Beispiel eine Zielfunktion, welche abhängig von zwei Parametern x_1 und x_2 ist, herangezogen (siehe Abbildung 3.7 a).

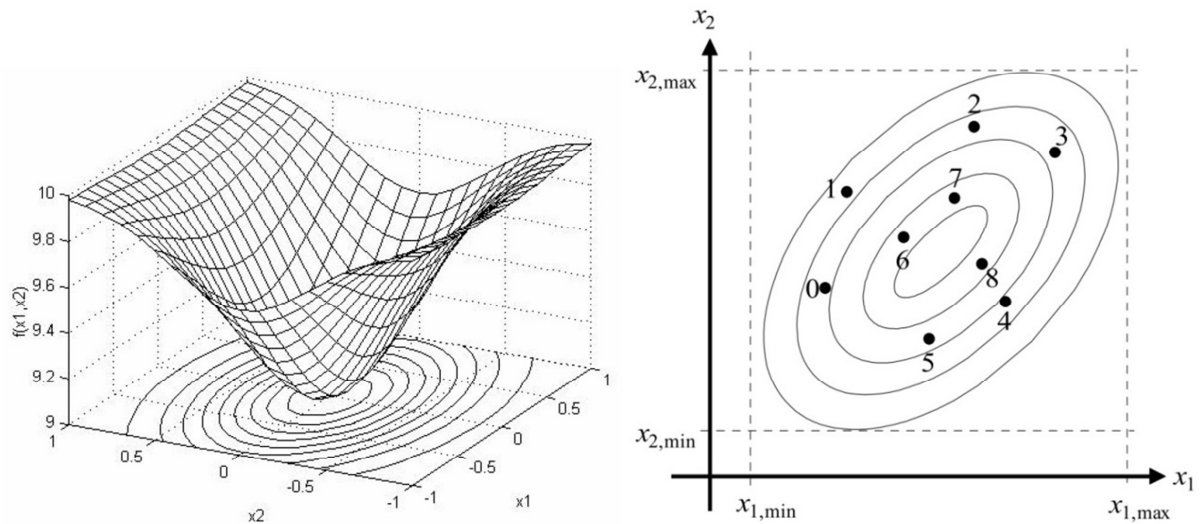


Abbildung 3.7: a.) Zielfunktion

b.) Initialisierung

Quelle: Price *et al.* [29]

Abbildung 3.7 b) zeigt die zufällig verteilte Initialisierung mit einer Populationsgröße von 9 und den Begrenzungen $x_{1,\min}$, $x_{1,\max}$, $x_{2,\min}$ und $x_{2,\max}$.

3.3.2.2 Mutation

Nach der Initialisierung startet der DE mit einer Mutation (Mutationsvektor $v_{i,g}$) und einer Rekombination, um den Nachkommenvektor $u_{i,g}$ für die Population N_p zu erzeugen. Um den Mutationsvektor $v_{i,g}$ zu berechnen, wird ein Skalierungsfaktor F mit der Differenz von zwei zufällig gewählten Vektoren multipliziert und zu einem dritten Vektor hinzu addiert. Formel (3.8) zeigt, wie die drei verschiedenen, zufällig gewählten Vektoren aus der aktuellen Population kombiniert werden, um den Mutationsvektor zu erzeugen (siehe Price *et al.* [29]).

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \quad (3.8)$$

Diese Formel trifft auf das klassische DE Schema zu. Nähere Informationen zu den anderen Schemen findet man unter [19], [20], [21]. Die Vektorindizes r_0 , r_1 , r_2 werden zufällig aus der Population $[1, N_p]$ ausgewählt, sind verschieden zueinander und zu dem Zielvektorindex. Der Zielvektor $x_{i,g}$ („target vector“) stammt ebenfalls von der aktuellen Population und ist für die spätere Rekombination („Crossover“) und Selektion wichtig. Der Skalierungsfaktor F ist ein positiver realer Kontrollparameter, welcher typischerweise zwischen $[0,4 \ 1]$ liegt und die Mutations-

schrittweise bzw. die Entwicklung der Population steuert. Abbildung 3.8 a) zeigt den Differenzvektor aus den zufällig gewählten Parametervektoren x_6 und x_8 . Die Gewichtung und Addition zum Vektor x_4 , um den Mutationsvektor v_0 zu erhalten, sieht man in Abbildung 3.8 b) (siehe Price *et al.* [29]).

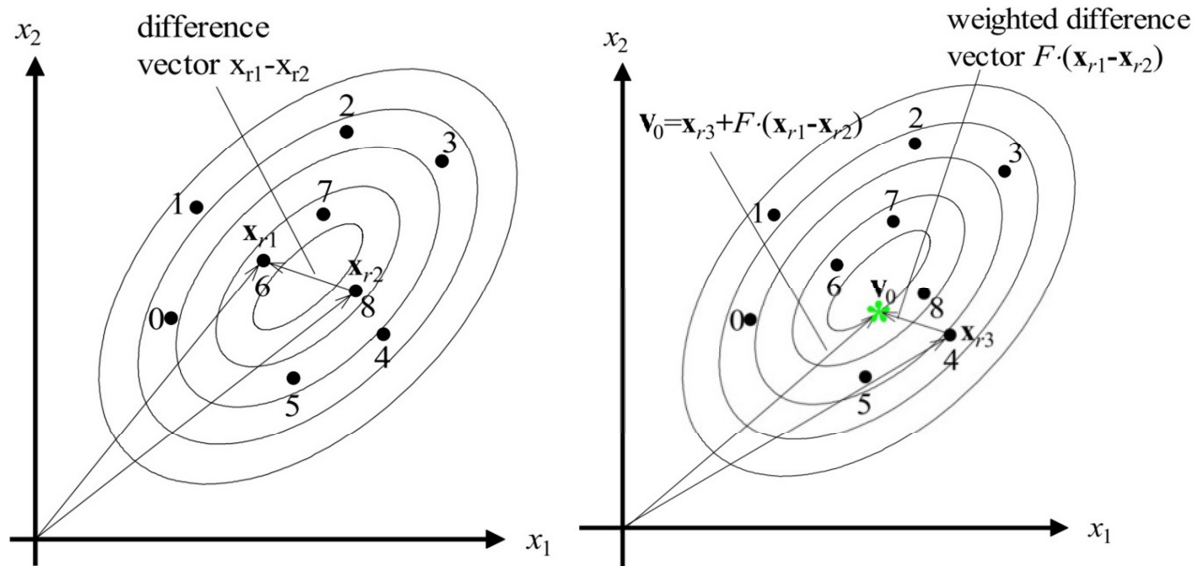


Abbildung 3.8: a.) Differenzvektor b.) Mutation

Quelle: Price *et al.* [29]

Abbildung 3.10 zeigt wie sich der Differenzvektor verhält, wenn man sich Richtung globales Minimum bewegt. Als Zielfunktion (Abbildung 3.9) wird wieder eine einzelne Funktion (Peak Funktion, Formel 3.9) verwendet, die abhängig ist von den zwei Parametern x_1 und x_2 (vergleiche Uday [33]).

$$\begin{aligned}
 f(x_1, x_2) = & 3 \cdot (1 - x_1)^2 \cdot \exp(x_1^2 + (x_2 + 1)^2) - \\
 & 10 \cdot \left(\frac{x_1}{5} - x_1^3 - x_2^5\right) \cdot \exp(x_1^2 + x_2^2) - \\
 & \frac{1}{3} \cdot \exp((x_1 + 1)^2 + x_2^2)
 \end{aligned} \tag{3.9}$$

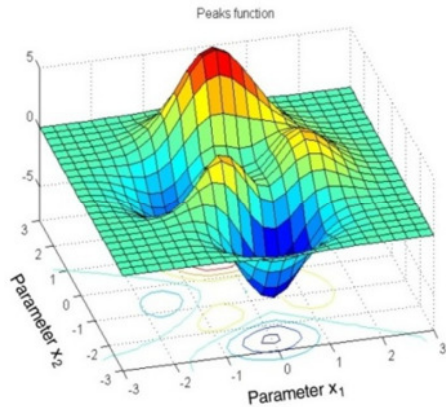


Abbildung 3.9: Zielfunktion

Quelle: Uday [33]

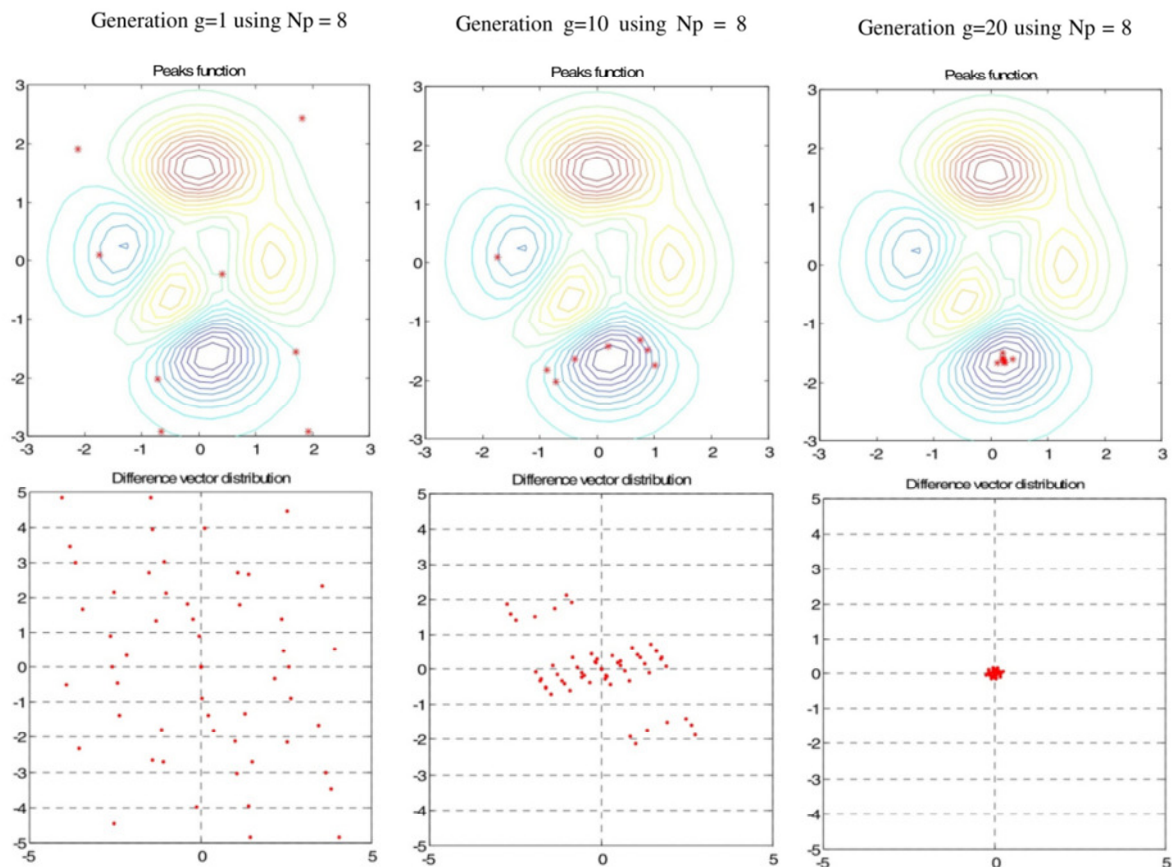


Abbildung 3.10: Differenzvektor Verteilung

Quelle: Uday [33]

In Abbildung 3.10 sieht man, wie die Population (erste Reihe) immer mehr Richtung globales Minimum wandert und sich zusammen zieht. Die untere Reihe zeigt die Verteilung der Differenzvektoren. Anfangs sind die Populationen noch mehr verstreut und die Differenzvektoren

daher auch größer und mehr verteilt. Wenn sich die Population im globalen Minimum zusammenzieht, werden sich die Differenzvektoren ebenfalls um den Nullpunkt ansiedeln. Durch den zuvor genannten Skalierungsfaktor F ist es möglich, die Mutationsschrittweite zu kontrollieren und aufgrund dessen die Konvergenzgeschwindigkeit der Population zu steuern.

3.3.2.3 Crossover

Die Rekombination („Crossover“) erzeugt einen neuen Nachkommenvektor $u_{i,g}$ („trial vector“) aus den Parametern des Zielvektors $x_{i,g}$ und des Mutationsvektors $v_{i,g}$. Der Kontrollparameter C_r steuert den Anteil von Parameterwerten, welche aus dem Mutationsvektor kopiert werden sollen und welche aus dem Zielvektor kommen. Am häufigsten wird die binomial Form verwendet, die wie folgt definiert ist (siehe Price *et al.* [29]).

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{wenn } \text{rand}_j[0,1] \leq C_r \text{ oder } j = j_{\text{rand}} \\ x_{j,i,g} & \text{ansonst} \end{cases} \quad (3.10)$$

$$\begin{aligned} \text{mit } j &= 0, 1, \dots, D - 1 \\ i &= 0, 1, \dots, Np - 1 \\ g &= 0, 1, \dots, g_{\text{max}} \end{aligned}$$

Der benutzerdefinierte Kontrollparameter C_r wird mit einer gleichverteilten Zufallszahl $\text{rand}_j[0,1]$ verglichen. Wenn die Zufallszahl kleiner oder gleich C_r ist, so bekommt der Nachkommenvektor $u_{i,g}$ („trial vector“) die Parameter des Mutationsvektors $v_{i,g}$ vererbt. Wenn zum Beispiel ein C_r Faktor von 0,5 gewählt wird und $\text{rand}_0 [0 \ 1]$ gleich 0,4 ergibt, so verwendet der Nachkommenvektor $u_{0,i,g}$ die Parameter aus dem Mutationsvektor $v_{0,i,g}$. Würde $\text{rand}_0 [0 \ 1]$ größer als 0,5 (C_r) sein, so erhält der Nachkommenvektor $u_{0,i,g}$ den Parameter aus dem Zielvektor $x_{0,i,g}$. Abbildung 3.11 zeigt, wie die Rekombination bei 2 Parametern aussieht.

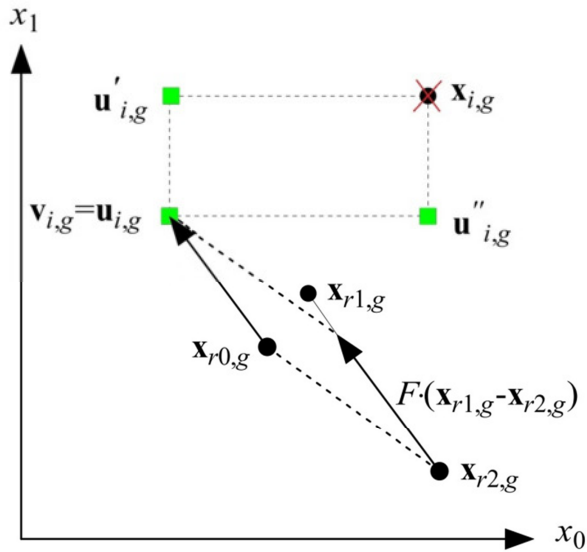


Abbildung 3.11: Crossover

Quelle: Price *et al.* [29]

Die Vektoren $u_{i,g}$, $u'_{i,g}$, $u''_{i,g}$ (grüne Rechtecke) sind in der Abbildung 3.11 mögliche Kandidaten für den Nachkommenvektor $u_{i,g}$, je nachdem, welche Parameter aus dem Zielvektor $x_{i,g}$ gewählt werden. Der mögliche Nachkommenvektor $u'_{i,g}$ erhält den Parameter x_0 von dem Mutationsvektor $v_{i,g}$ und den Parameter x_1 von dem Zielvektor $x_{i,g}$. Bei dem anderen möglichen Nachkommenvektor $u''_{i,g}$ wird der Parameter x_0 von dem Zielvektor $x_{i,g}$ und der Parameter x_1 von dem Mutationsvektor $v_{i,g}$ genommen. Bei einem Crossover Faktor Cr von 1 wird der Mutationsvektor $v_{i,g}$ komplett als Nachkommenvektor $u_{i,g}$ verwendet. Der zufällig gewählte Index j_{rand} wird aus dem Mutationsvektor $v_{i,g}$ genommen, um sicher zu stellen, dass nicht der gesamte Zielvektor $x_{i,g}$ in den Nachkommenvektor $u_{i,g}$ kopiert wird. Dies würde auch bei der nachfolgenden Selektion im nächsten Kapitel keinen Sinn machen, weil dadurch zweimal dieselben Funktionswerte miteinander verglichen werden. Die Mutation aus dem vorherigen Kapitel sieht man ebenfalls in Abbildung 3.11. Hierbei wird der Differenzvektor zwischen $x_{r1,g}$ und $x_{r2,g}$ mit dem Skalierungsfaktor F gewichtet und zu dem dritten Vektor $x_{r0,g}$ hinzu addiert, um den Mutationsvektor $v_{i,g}$ zu erhalten.

Da es eine Vielzahl von Differential Evolution Schemen in Bezug auf Mutation (rand, best, current usw.) und Crossover (binomial, exponentiell, usw.) gibt, wird auf folgende Arbeiten verwiesen [9], [29], [12].

3.3.2.4 Selektion

Bei der Selektion einer Einzelfunktion wird der Funktionswert des Nachkommenvektors $u_{i,g}$ mit dem Funktionswert des Zielvektors $x_{i,g}$ verglichen. Sollte der Nachkommenvektor $u_{i,g}$ einen gleichen oder kleineren Funktionswert ergeben, so ersetzt der Nachkommenvektor $u_{i,g}$ den Zielvektor $x_{i,g}$ in der Population der nächste Generation (siehe Price *et al.* [29]).

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{wenn } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{ansonst} \end{cases} \quad (3.11)$$

Nachdem die neue Population (Generation + 1) generiert wurde, startet der Algorithmus wieder von vorne mit der Mutation, Rekombination und Selektion, solange bis ein Optimum gefunden oder ein Abbruchkriterium erreicht wurde.

Abbruchkriterien:

Folgende Abbruchbedingungen können wie folgt definiert werden (siehe Price *et al.* [29]):

- 1.) Feste Anzahl von Iterationen g_{\max} , Funktionsaufrufen oder Simulationszeit
- 2.) Wenn sich der beste Funktionswert einer Population nicht mehr wesentlich ändert
- 3.) Wenn ein vorgegebener Wert der Zielfunktion erreicht wurde
- 4.) Wenn sich die Population zusammen zieht bzw. der Differenz Vektor gegen Null geht (gilt nur bei einer einzelnen Zielfunktion)

Die Wahl der richtigen Differential Evolution Kontrollparameter (Populationsgröße N_p , Mutationskalibrierungsfaktor F und Crossover Faktor Cr) haben einen großen Einfluss auf die Konvergenzgeschwindigkeit und das Auffinden des globalen Minimums und werden daher kurz erläutert.

Kontrollparameter:

Es gibt drei wesentliche Kontrollparameter des DE Algorithmus (siehe Price et al [29], Das und Suganthan [9]).

- Mutation Skalierungsfaktor F : Steuert, wie schnell die Population zusammen wächst
- Crossover Faktor Cr : sagt, wie viele Parameter vom Mutationsvektor $v_{i,g}$ oder vom Zielvektor $x_{i,g}$ in den Nachkommenvektor $u_{i,g}$ kopiert werden.
- Populationsgröße N_p

Price *et al.* haben in [29] gezeigt, dass ein vernünftiger Wert für N_p zwischen $5 \cdot D$ und $10 \cdot D$ (D ist die Anzahl der zu optimierenden Parameter) liegt. Der effektive Bereich des Skalierungsfaktors F liegt in der Regel zwischen 0,4 und 1. Als ersten guten Wert kann man 0,85 wählen. Für den Crossover Faktor hat sich ebenfalls ein Wert von 0,5 etabliert (siehe Price *et al.* [29]).

3.3.2.5 Nebenbedingungen

Die Nebenbedingungen können entweder die Parameter oder andere Funktionen betreffen. Da die Parameter bereits wie in Kapitel 3.3.2.1 begrenzt werden, wird hier auf die Nebenbedingungen eingegangen, die andere Funktionen miteinbeziehen. Die Nebenbedingungen lassen sich wie folgt in den Optimierungsalgorithmus einbauen (siehe Price [29]).

- Bestrafungsterme: wird eine Randbedingung nicht erfüllt, so wird der Zielfunktion ein großer Wert (Bestrafungsterm) hinzuaddiert (bei Minimierung).
- Reparaturalgorithmus
- Multikriterielle Optimierung - Paretomenge

Wie bereits in Kapitel 3.2 begründet, werden die Nebenbedingungen jedoch als zu minimierende Funktionen eingebaut, um so eine Menge von Lösungen zu bekommen. Es ergibt sich also folgendes Minimierungsproblem durch die vier Zielfunktionen f_1 , f_2 , f_3 und f_4 (siehe Formel 3.12).

$$\begin{aligned}\vec{Y} &= f(\vec{X}, \text{Engine}, \text{ECU set}, \text{time}) \\ &= \left(f_1(\vec{X}, \text{Engine}, \text{ECU set}, \text{time}), \dots, f_4(\vec{X}, \text{Engine}, \text{ECU set}, \text{time}) \right)\end{aligned}\quad (3.12)$$

$$\text{mit } \vec{X} = [K_{p \text{ Pos}}, K_{i \text{ Pos}}, K_{p \text{ Neg}}, K_{i \text{ Neg}}]^T$$

wobei *Engine* und *ECU set* konstante Parameter beinhalten

3.3.3 Multikriterielle Optimierung

Ein wesentlicher Punkt bei der stochastischen Optimierung ist die Selektion. Bei der Einzelfunktion ist die Entscheidung, welche Parameter behalten werden, einfach. Dort überlebt ein Parametervektor aufgrund seines besseren Funktionswertes, bis das Optimum gefunden wurde (siehe Formel 3.11). Bei der multikriteriellen Optimierung ergeben sich aber folgende zwei Probleme (siehe Alotto [3]):

- Was ist der beste Wert?
- Wie selektiert man diesen heraus?

Eine einfache Lösung dieser Probleme wäre die Gewichtung der einzelnen Zielfunktionen und Zusammenfassung zu einer neuen skalaren Gütefunktion. Die Wahl der Gewichte ist dabei kritisch zu betrachten. Durch die Abhängigkeit der Lösung von den Gewichten gibt man bereits eine gewisse Richtung der Lösung vor (siehe Alotto [3]).

Eine weitere Möglichkeit, das multikriterielle Optimierungsproblem zu lösen, ist durch die Zuhilfenahme der Pareto Front. Die Lösung des Optimierungsproblems ist dadurch nicht mehr ein Parametervektor, sondern eine Menge von Vektoren (Paretomenge).

3.3.3.1 Pareto Front

Die Lösung eines multikriteriellen Problems liefert keine Einzel-, sondern eine Menge von Lösungen, welche im Folgenden als Pareto-Menge bezeichnet wird. Die Pareto Front beinhaltet die Zielfunktionswerte der entsprechenden Parameter (siehe Alotto [2]).

Ziel der multikriteriellen Minimierung ist es, möglichst nahe an die optimale Pareto Front heran zu kommen (Konvergenz) und einen großen Bereich dieser abzubilden (Diversität). Bei Lösungen auf der Pareto Front ist es nicht möglich, eine Zielfunktion (f_1) zu verbessern, ohne eine andere zu verschlechtern (f_2) (siehe Abbildung 3.12 a) (siehe Altendorfner [4]).

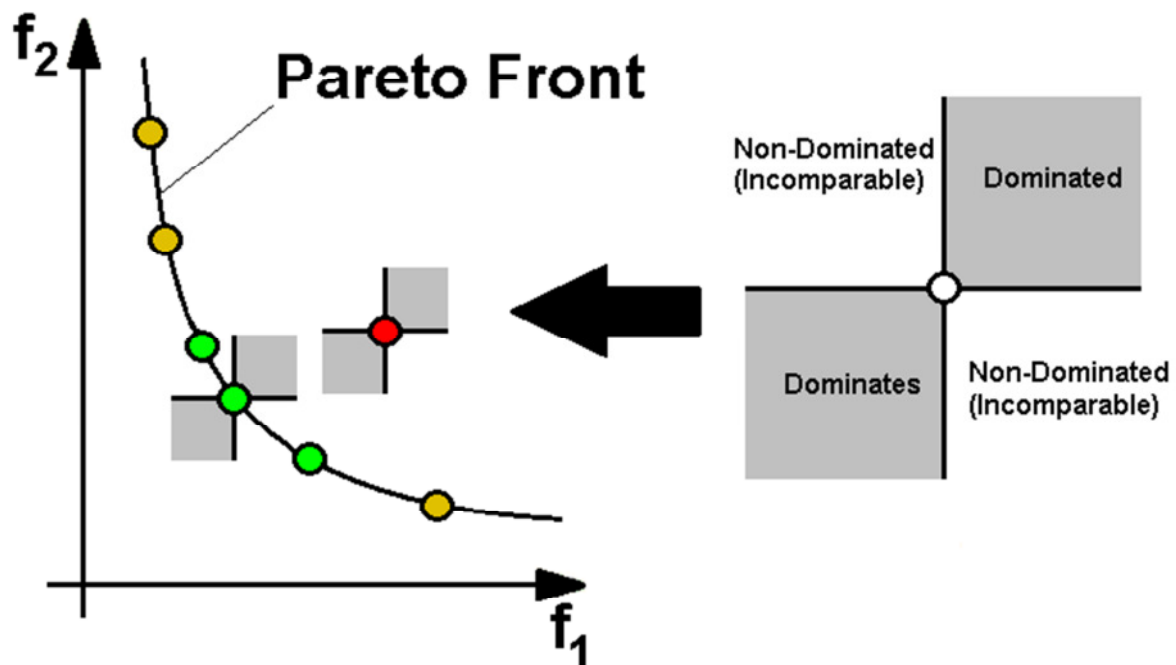


Abbildung 3.12: a.) Pareto Front

b.) Pareto Dominanz

In Abbildung 3.13 a) ist zu sehen, dass die grünen Punkte den roten Punkt dominieren. Die grünen und gelben Punkte bilden die Pareto Front. Bei diesen kann man aber nicht sagen, welcher besser ist. Entweder es ist nur Zielfunktionswert 1 (f_1) besser oder es ist nur Zielfunktionswert 2 (f_2) besser, aber es sind niemals beide Funktionswerte besser („Non-Dominated“).

3.3.3.2 Non-Dominated Sorting Differential Evolution (NSDE)

Der Non-Dominated-Sorting Differential Evolution (NSDE) von Antony und Xiadong in [5] ist eine Verknüpfung des Standard DE (wie in Kapitel 3.3.1 beschrieben) und dem Non-Dominated-Sorting Algorithmus, wie er im NSGA-II (Non-Dominated Sorting Genetic Algorithm) von Deb *et al.* in [10] eingesetzt wird. Der NSGA-II verwendet zwei Sortiermethoden, um die Lösungsmengen zu ordnen:

- Pareto Rangordnung durch die Pareto Dominanz (siehe Abbildung 3.12 b) und
- Crowding Distanz.

Die Population mit den Zielvektoren $x_{i,g}$ wird im folgenden Elternpopulation („Parent“) und die des Nachkommenvektors $u_{i,g}$ Nachkommenpopulation („Child“) genannt. Diese zwei werden zu einer kombinierten Population mit einer Größe von 2 mal N_p verknüpft, danach sortiert und wieder auf eine Populationsgröße von N_p zurück geteilt. Folgendes Schema soll den Verlauf des Non-Nominated Sorting Algorithmus genauer erklären.

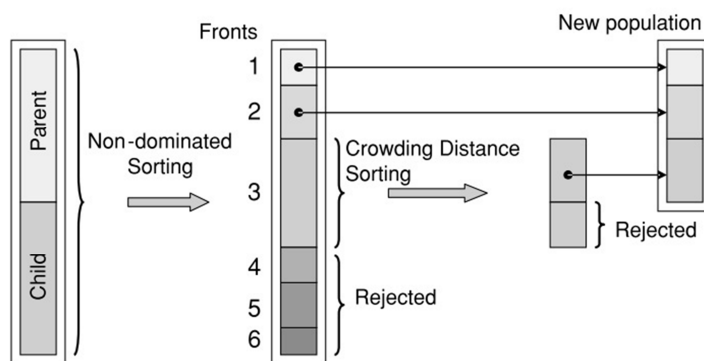


Abbildung 3.13: Schema des NSGA-II

Quelle: Deb *et al.* [22]

Der gesamte NSDE Algorithmus, wie er in [10] beschrieben ist, setzt sich wie folgt zusammen:

- 1.) Initialisiere eine zufällige Population N_p und berechne die Zielfunktionswerte
- 2.) Generiere neue Nachkommen nach der DE Strategie (siehe Kapitel 3.3.1)
- 3.) Berechne die Funktionswerte der Nachkommen
- 4.) Verknüpfe die Population der Eltern mit denen der Nachkommen zu einer kombinierten Population der Größe $2 N_p$.
- 5.) Non-Dominated Sorting:**
 - a.) Sortiere die kombinierte Population in zunehmender Rangfolge nach dem non-dominated Rang (1 bester Rang, 2 nächst bester Rang usw.)
 - b.) Die einzelnen Pareto Fronten aus den verschiedenen Rängen werden in absteigender Reihenfolge nach der Crowding Distanz sortiert.
 - c.) Kürzen der sortierten Population auf die Originalgröße N_p .
- 6.) Gehe zu 2, bis das Abbruchkriterium des Optimierers erfüllt ist.

Abbildung 3.14 zeigt eine genauere Darstellung der Schritte 5 a) und 5 b) (Pareto Rangordnung und Berechnung der Crowding Distanz).

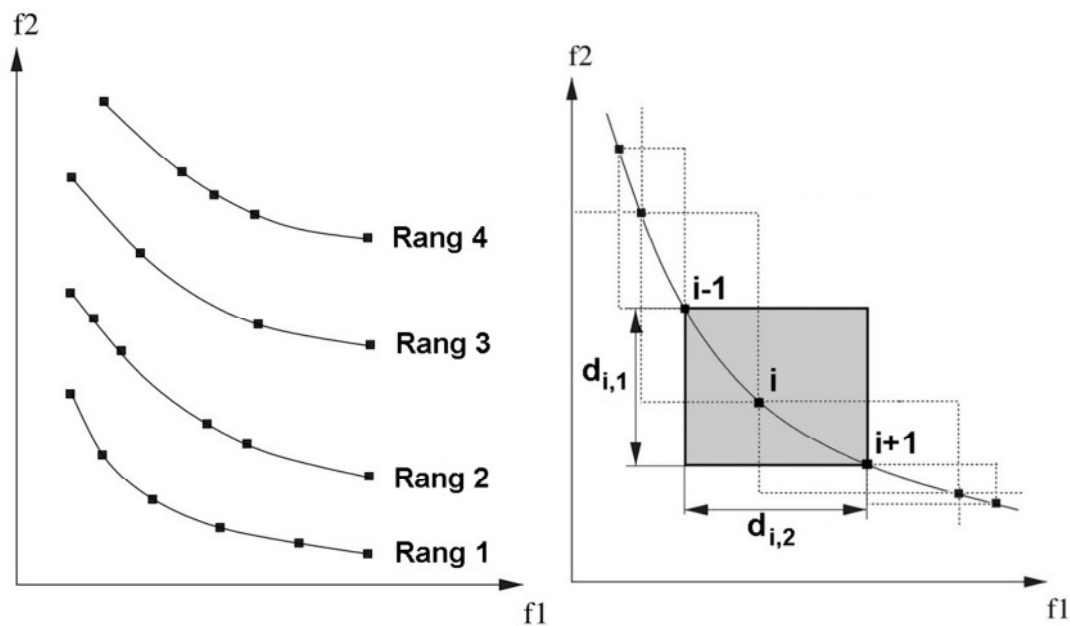


Abbildung 3.14: a.) Pareto Rangordnung b.) Crowding Distanz

Rang 1 bekommen jene Funktionswerte, welche nicht von anderen Werten dominiert werden (Non-Dominated, siehe Abbildung 3.12 b). Danach wird die nächste Front gesucht, welche den nächst schlechteren Rang 2 erhält. Dies passiert solange, bis alle Funktionswerte einen Rang zugeordnet bekommen haben. Die Sortierung nach dem Pareto Rang erfolgt in zunehmender Reihenfolge. Innerhalb der verschiedenen Fronten wird die Crowding Distanz nach folgender Formel berechnet:

$$distance_{(i)} = d_{i,1} + d_{i,2} \quad (3.13)$$

Für jeden Funktionswert (i) wird die Crowding Distanz berechnet, indem die zwei Seitenlängen $d_{i,1}$, $d_{i,2}$, welche die benachbarten Funktionswerte (i-1), (i+1) bilden, addiert werden (siehe Abbildung 3.14 b). Innerhalb der einzelnen Fronten (Rang 1, Rang 2, usw.) wird in absteigender Reihenfolge der Crowding Distanz sortiert. Das heißt, die Funktionswerte, welche am weitesten auseinander (größten Crowding Distanzen) liegen, werden bevorzugt. Damit ist es möglich, einen großen Bereich der Pareto Front abzubilden (Diversität).

Im letzten Schritt 5 c) wird die nun sortierte Population nach Abschluss der 2 Sortiermethoden (Rangordnung und Crowding Distanz) wieder auf ihre Originalgröße N_p zurück gekürzt und der Optimierungsalgorithmus wird bis zur Erfüllung des Abbruchkriteriums weiter durchlaufen.

Mit den nun erhaltenen theoretischen Grundlagen aus Kapitel 2 und 3 ist es möglich, das Motormodell, den Laderuckregler und den Optimierungsalgorithmus in eine Simulationsumgebung praktisch zu implementieren und zu testen.

4 Implementierung

In diesem Kapitel startet der praktische Teil dieser wissenschaftlichen Arbeit. Er beinhaltet die Implementierung des Motormodells, den Nachbau des Ladedruckreglers aus dem Motorsteuergerät in einer Simulationsumgebung und der Ausführung des Optimierungsverfahrens.

4.1 Entwicklungsumgebung

Der Nachbau des Motorsteuergeräts erfolgt in MATLAB/Simulink und auch das global dynamische Motormodell liegt als Subblock in Simulink vor. Der Differential Evolution Algorithmus wurde als MATLAB m File programmiert.

Die ganze Optimierung läuft auf einen Intel core i7-2820QM Laptop mit 4 Kernen zu jeweils 2,3 GHz und einem 64 Bit Windows 7 Betriebssystem. Für einen Fahrzyklus mit einer Dauer von 50 Sekunden benötigt der Rechner ca. 4,5 Sekunden für eine Simulation. Bei einer Populationsgröße von 50 Zielvektoren und 200 Generationen ergibt sich so eine ungefähre Dauer von 12 Stunden für die komplette Optimierung der 4 Parameter (K_{pPos} , K_{iPos} , K_{pNeg} , K_{iNeg}).

4.2 Modellgenerierung

Das global dynamische Motormodell (siehe Abbildung 4.1) besteht aus folgenden Blöcken:

- Eingänge
- Aktuatordynamik (Wastegate mit pneumatischem Stellglied)
- Global dynamisch (Motormodell)
- Global statisch
- Sensordynamik
- Ausgänge (Ladedruck nach dem Ladeluftkühler, siehe Abbildung 2.2)

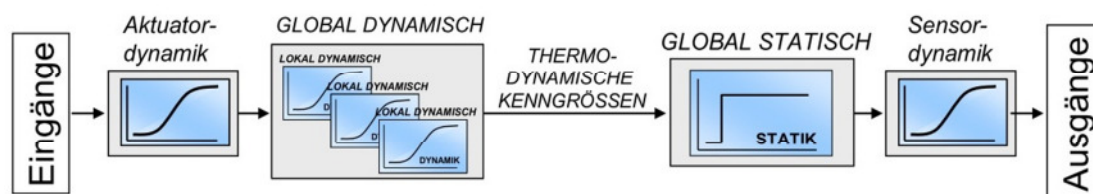


Abbildung 4.1: global dynamisches Modell

Quelle: Nebel *et al.* [24]

Die Eingänge des Modells sind:

- 1.) Haupteinspritzmenge,
- 2.) Haupteinspritzzeitpunkt (SOI = „Start of injection“),
- 3.) Wastegate Position und
- 4.) Drehzahl.

Das Modell wurde mit Hilfe von APBR Signalen an diesen Eingängen und den entsprechenden Messungen trainiert (siehe Kapitel 2.5.4). Der Eingangsbereich in dem das gesamte Motormodell gültig ist, erstreckt sich von einer Drehzahl von 800 bis 4150 rpm und einer Einspritzmenge von 5 mg/Hub bis Vollast (300Nm \approx 60mg/Hub). Der Haupteinspritzzeitpunkt (SOI) ist abhängig vom Basiskennfeld und hat eine relative Abweichung von $\pm 4^\circ$. Die Wastegate Position wird zwischen 0 und 100% des Öffnungswinkels begrenzt.

Der Block „Aktuator Dynamik“ bildet die Nichtlinearität des Wastegates und dessen pneumatischen Stellgliedes ab (siehe Kapitel 2.1.3.1). Dieser Block beinhaltet ein einzelnes dINN (dynamic Intelligent Neural Network, Kapitel 2.5.3).

Der Luftpfad des Motormodells ist in dem Block „global dynamisch“. Es besteht aus 22 lokalen nichtlinearen Teilmodellen, welche als dINN realisiert sind. Auf die Blöcke „global statisch“ und „Sensordynamik“ wird hier nicht näher eingegangen, weil in dieser Arbeit der Ausgang des Blocks „global dynamisch“ als Istwert (Ladedruck) verwendet wurde.

4.3 Implementierung in Simulink

Um die Dynamik des Motors abbilden zu können, muss das Modell Rückführungen mit Speicherblöcken (Regressoren) beinhalten (siehe 2.5.3, Abbildung 2.14: Baumstruktur dINN). Da die Anfangswerte dieser Speicherblöcke erst erlernt werden müssen, werden dem Modell die ersten 1,5 Sekunden der Messdaten initialisiert.

4.3.1 Global Dynamisches Modell

Wie bereits im vorherigen Kapitel erwähnt, wurden die Blöcke „global statisch“ und „Sensordynamik“ weggelassen. Abbildung 4.2 zeigt den tatsächlichen Aufbau des benutzten Modells in Simulink.

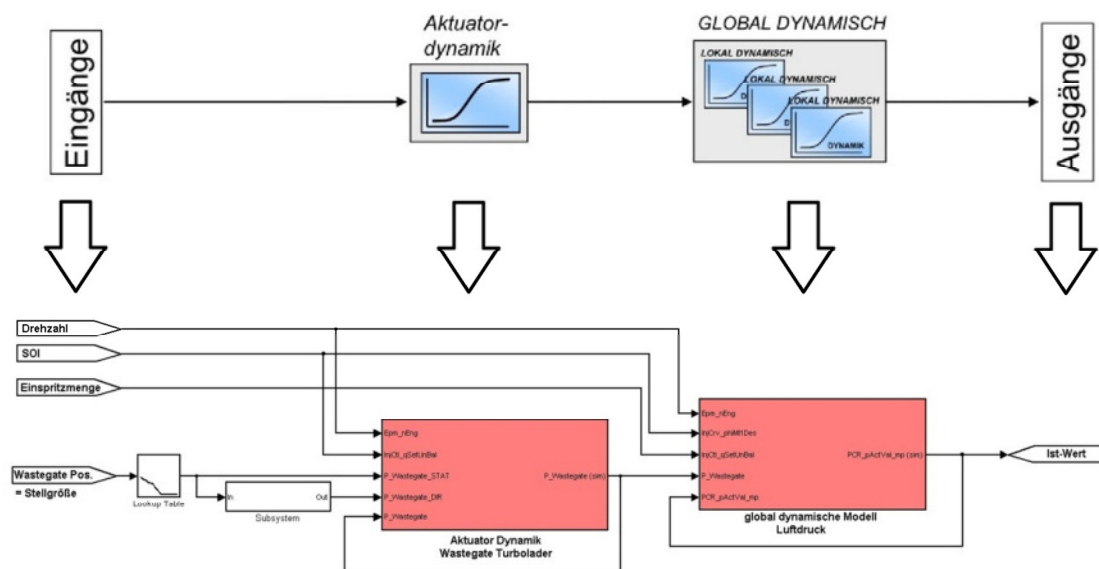


Abbildung 4.2: Simulink Aufbau des global dynamischen Motormodell

Zur Generierung des Motormodells wurden die Trainingsdaten (APRBS) mit 10 Hz (100ms) abgetastet. Aufgrund der Tatsache, dass der Ladedruckregler der ECU mit einer Abtastrate von 20ms arbeitet (siehe Kapitel Ladedruckregler 2.4.1), musste das Motormodell um 5-Speicherblöcke erweitert werden, damit die gleiche Dynamik vorhanden bleibt. Das global dynamische Modell beinhaltet nur den Luftpfad (siehe Kapitel 2.1.2) und ist abhängig von Drehzahl, Einspritzmenge, SOI und Wastegate Position. Die 22 lokalen dINN's sind als c-Files in dem Modell eingebunden.

4.3.2 Motorsteuerung ECU

Die Ladedruckregelung der ECU sieht wie in Abbildung 4.3 aus und besteht aus den folgenden wichtigen Blöcken:

- Basissollwertkennfeld
- Limits

- Parameteradaption
- SOLL-IST Vergleich
- PID Regler (aus I-Anteil, P-Anteil, DT1-Anteil, DT1_2-Anteil)
- Vorsteuerwert
- Anti Wind up

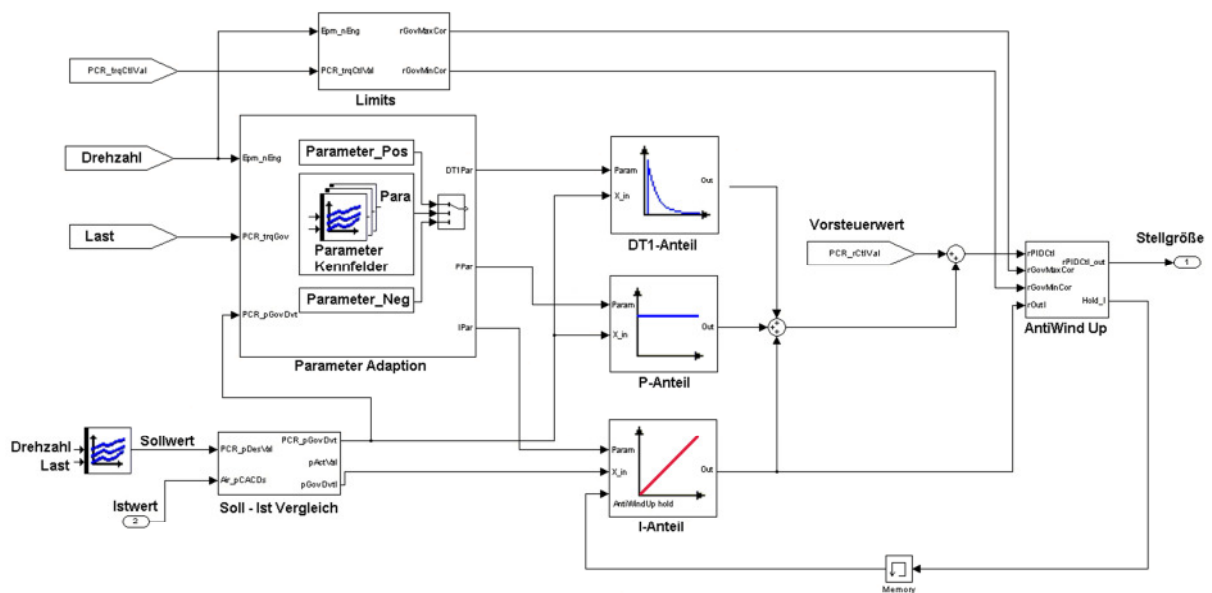


Abbildung 4.3: Simulink Aufbau des Ladedruckregler ECU

Das Basissollwertkennfeld ermittelt den Ladedruck in Abhängigkeit von der Motordrehzahl und der eingespritzten Kraftstoffmenge. Tabelle 4.1 zeigt, wie ein Betriebspunkt (Drehzahl/Last) abhängiges Kennfeld aussieht, anhand eines Beispiel Kennfelds für die Reglerparameter. Die Sollwerte für das Basissollwertkennfeld werden ebenfalls von erfahrenen Applikateuren kalibriert und sind bei der Optimierung fix vorgegeben. Der Block Limits begrenzt die Reglerausgänge und ist ebenfalls vom Betriebspunkt (Drehzahl/Last) abhängig. Diese Begrenzungen werden über einen vom Außendruck abhängigen Korrekturfaktor richtig gestellt.

Der Block Parameteradaption beinhaltet die Parameter des PID-Reglers als auch die Betriebspunkt abhängigen Anpassungswerte, welche mithilfe des Gain Scheduling (Kapitel 2.4.2 Zusätzliche ECU Funktionen) durchgeschaltet werden. Da dieser Block ebenfalls die zu optimierenden Großsignalparameter beinhaltet, wird dieser in Kapitel 4.3.3, Abbildung 4.4, näher betrachtet. Der Kern des PID Ladedruckreglers besteht aus der Parallelschaltung der P-, I-, D- Glieder (siehe Kapitel 2.2.1).

Zu der berechneten Stellgröße wird in dem Steuergerät noch ein Vorsteuerwert hinzuaddiert. Dieser wird über ein Vorsteuerkennfeld (ähnlich dem Basiskennfeld) in Abhängigkeit des Betriebspunktes und durch verschiedene Korrekturen ermittelt. In diesem Vorsteuerkennfeld stehen die Stellgrößen, welche das Stellglied benötigt, um den vorgegebenen Ladedruck-Sollwert im stationären Endpunkt (Abbildung 2.9: Sprungantwort) zu erreichen. Die Aufgabe der Regelung ist es also, das Verhalten der Steuerung (Sollwert zu Vorsteuerwert) zu verbessern. Wenn aber der Ladedruck Sollwert mit dem Vorsteuerwert nicht erreicht wird, so ist eine bleibende Regelabweichung (schlechte Genauigkeit) vorhanden, trotz des I-Glieds im Regler.

Der Soll-Ist Vergleich ist eine einfache Differenz, wie er in dem Standard Regelkreis (Kapitel 2.2, Abbildung 2.6) abgebildet ist. Für den I-Anteil des PID Reglers wird der Sollwert vorher mit Hilfe eines PT1 Gliedes geglättet und erst dann mit dem Istwert verglichen. Das P und DT1 Glied erhalten das nicht geglättete Signal, da diese Glieder für die Schnelligkeit wichtig sind und das PT1 Glied sie langsam machen würde.

4.3.3 Parameteradaption

Die Streckenverstärkung der Ladedruckstrecke ändert sich sehr stark über den Arbeitsbereich und daher sollten die Regelparameter an den jeweiligen Motorbetriebspunkt angepasst werden. Hierfür dient der in Abbildung 4.4 abgebildete Block Parameteradaption. Dieser beinhaltet die zu optimierenden Parameter K_{pPos} , K_{iPos} , K_{pNeg} , K_{iNeg} (rote Blöcke) für den Großsignalbereich und auch die bereits optimierten Kleinsignalparameter K_p , K_i , usw. (grüne Blöcke). Die Kleinsignalparameter das DT1 Glied und das Drehzahl Last abhängige Anpassungswertkennfeld (siehe Tabelle 4.1) wurden bereits mit Hilfe des LQR Verfahrens (Kapitel 2.2.4) optimiert.

P-Parameter	Drehzahl [rpm]							
	Drehmoment [Nm]	750	1000	1250	1750	2000	2250
50	0.18	0.18	0.18	0.16	0.12	0.12	0.11
100	0.18	0.18	0.18	0.16	0.125	0.125	0.11
150	0.16	0.14	0.13	0.13	0.11	0.11	0.125
200	0.125	0.12	0.12	0.10	0.08	0.08	0.13
....
Volllast	0.10	0.10	0.10	0.09	0.07	0.07	0.11

Tabelle 4.1: Kennfeld

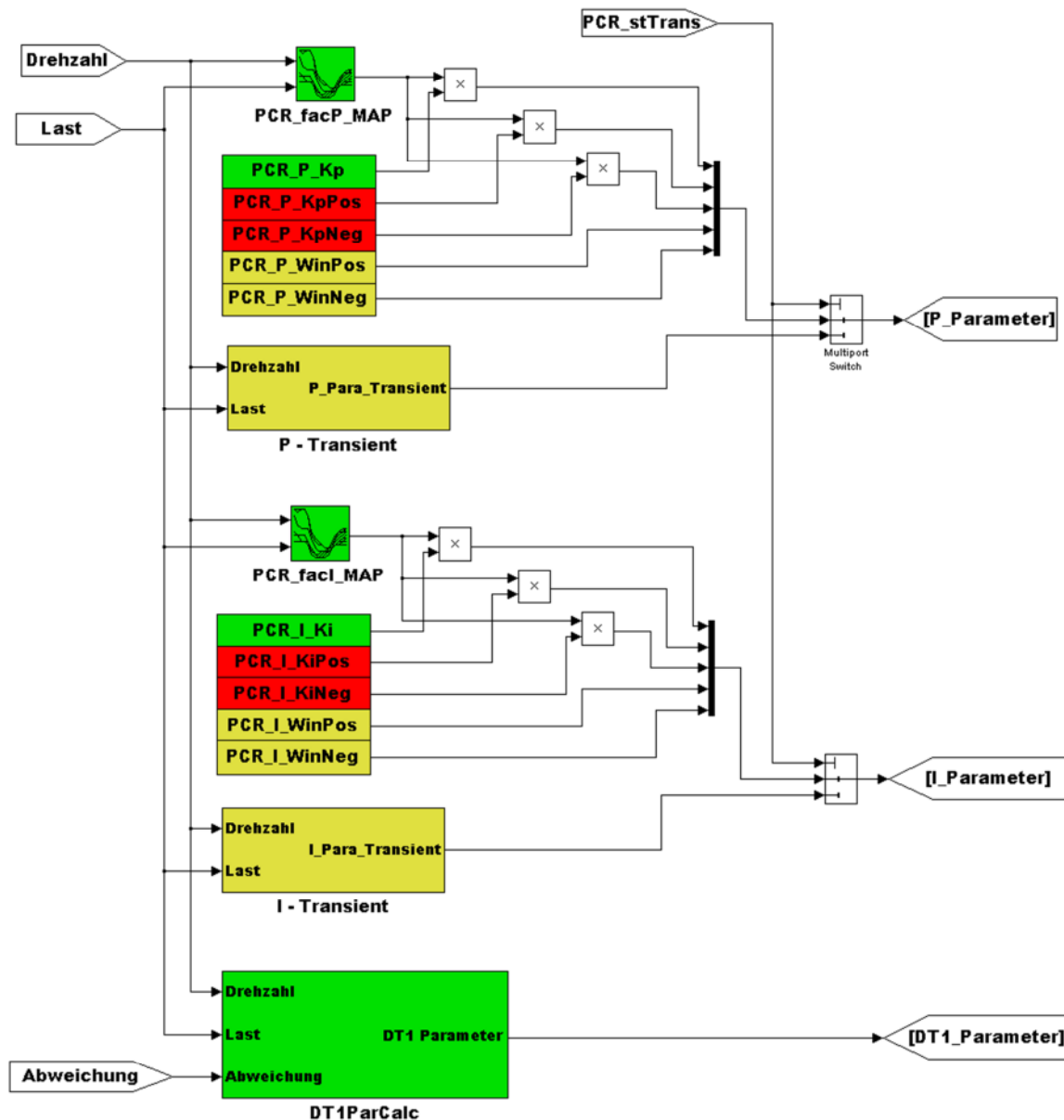


Abbildung 4.4: Simulink Aufbau der Parameteradaption

Die Werte, bei denen die Umschaltung zwischen Kleinsignal- und Großsignalparameter (Fensterumschaltung) stattfindet (gelb hinterlegte Blöcke), wurden mit +/- 50 hPa beibehalten. Die Parameter der transienten Blöcke (P-Transient, I-Transient, DT1 Transient) wurden ebenfalls in Ruhe gelassen. Diese Blöcke beinhalten die gleiche Struktur mit Kennfeld, K_p , K_{pPos} , K_{pNeg} usw. jedoch stehen sie für den transienten Betrieb zur Verfügung. Die Umschaltung von normalen auf transienten Betrieb erfolgt mit Hilfe des ECU internen Status Bits **PCR_stTrans**, welches bei einem zu starken Anstieg des Drehmoments gesetzt wird (siehe Kapitel 2.4.2 zusätzliche ECU Funktionen). Durch diesen transienten Betrieb würde sich die Anzahl der zu optimierenden Parameter verdoppeln, weshalb nur die Parameter für den normalen Betrieb optimiert werden.

4.4 Validierung

Bevor eine Optimierung durchgeführt werden kann, müssen das Motormodell und auch die nachgebildete ECU validiert werden. Dies erfolgt mit Hilfe von Messdaten (Eingangs- und Ausgangsdaten) und den gleichen Parametersätzen, wie sie in der realen ECU während der Messung eingestellt waren.

Abbildung 4.6, 4-7 zeigt die Validierung mit offenem Regelkreis (Steuerung). Hierzu wurden die Eingangsdaten (Sollwert und Istwert) in das ECU Simulink Modell geschickt und der Ausgang des simulierten Motorsteuergeräts mit den Messdaten verglichen.

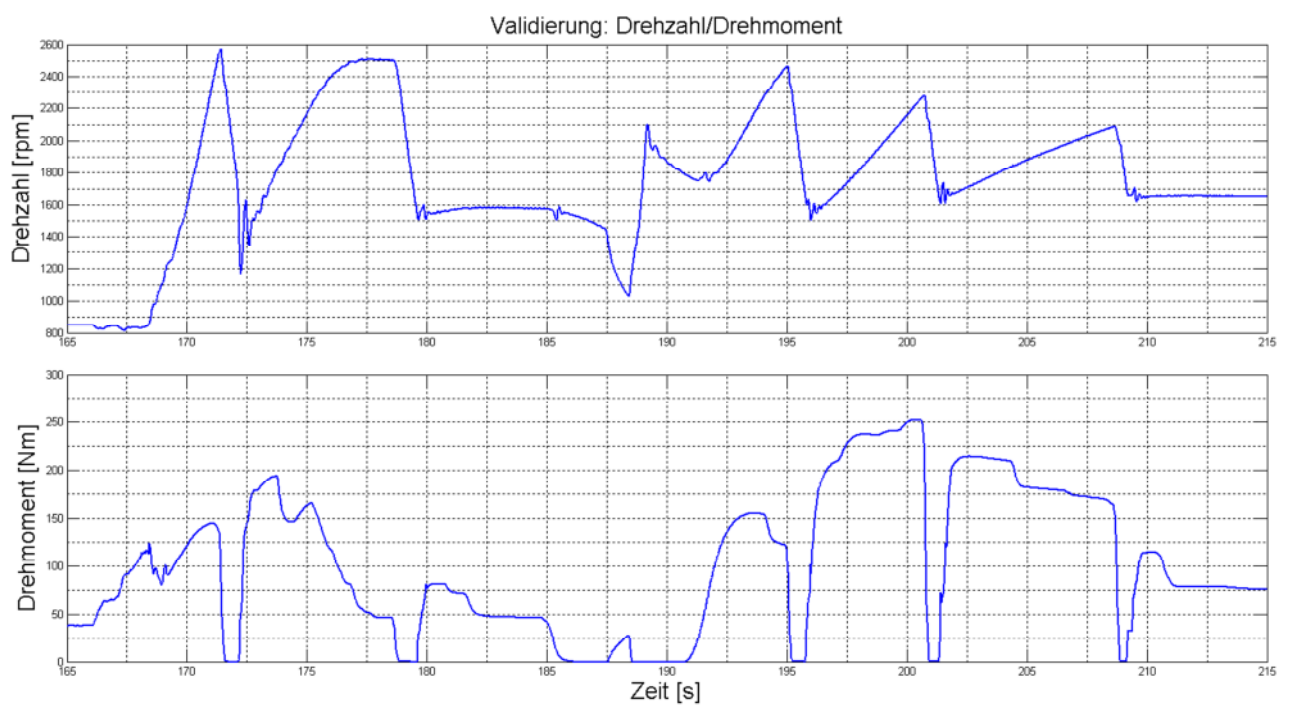


Abbildung 4.5: Validierung: Drehzahl/Drehmoment

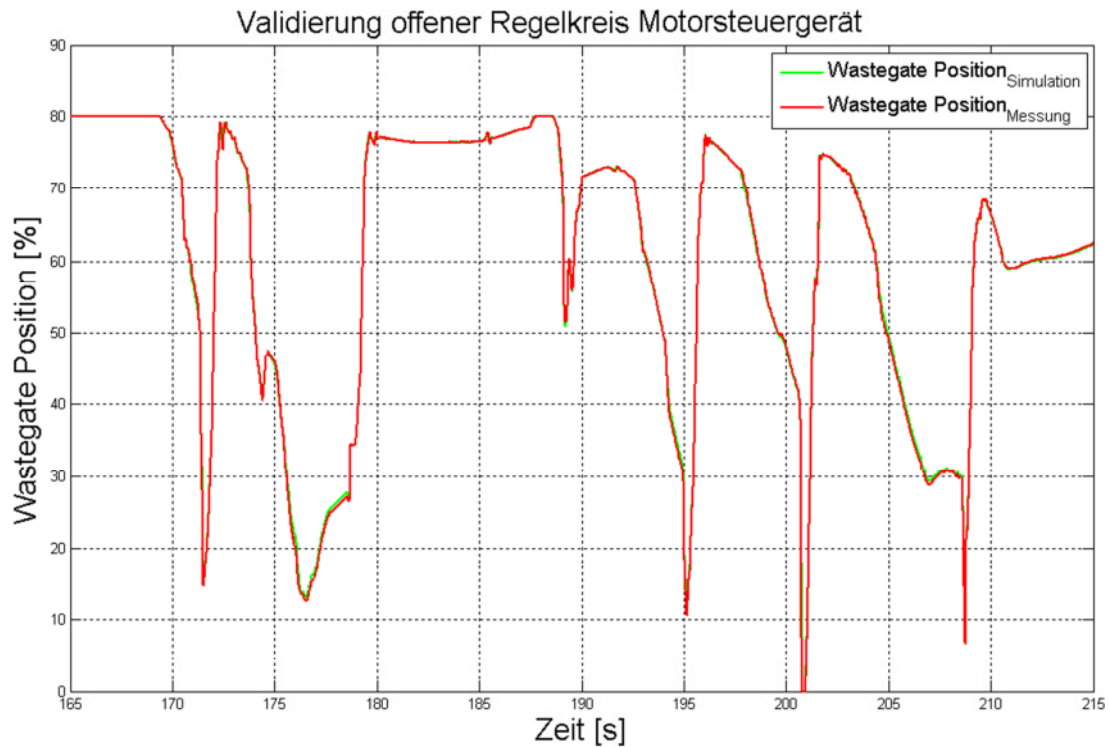


Abbildung 4.6: Validierung offener Regelkreis: Motorsteuergerät

Die minimale Abweichung des simulierten Motorsteuergerätes zu der realen ECU (Abbildung 4.6) lässt sich durch die zeitlich verschiedenen Schaltvorgänge bei den Entwicklungsumgebungen (MATLAB und ECU) erklären (siehe Homepage MSR [40]). Ebenso wurden die Messdaten nicht immer exakt bei 10ms abgetastet.

Die Validierung der geregelten Strecke erfolgte, indem der Sollwert der Messdaten in das Simulink Modell geschickt wird. Der simulierte Istwert des Modells wird zurückgeführt und in Abbildung 4.7 mit dem gemessenen Istwert aus den Messdaten verglichen.

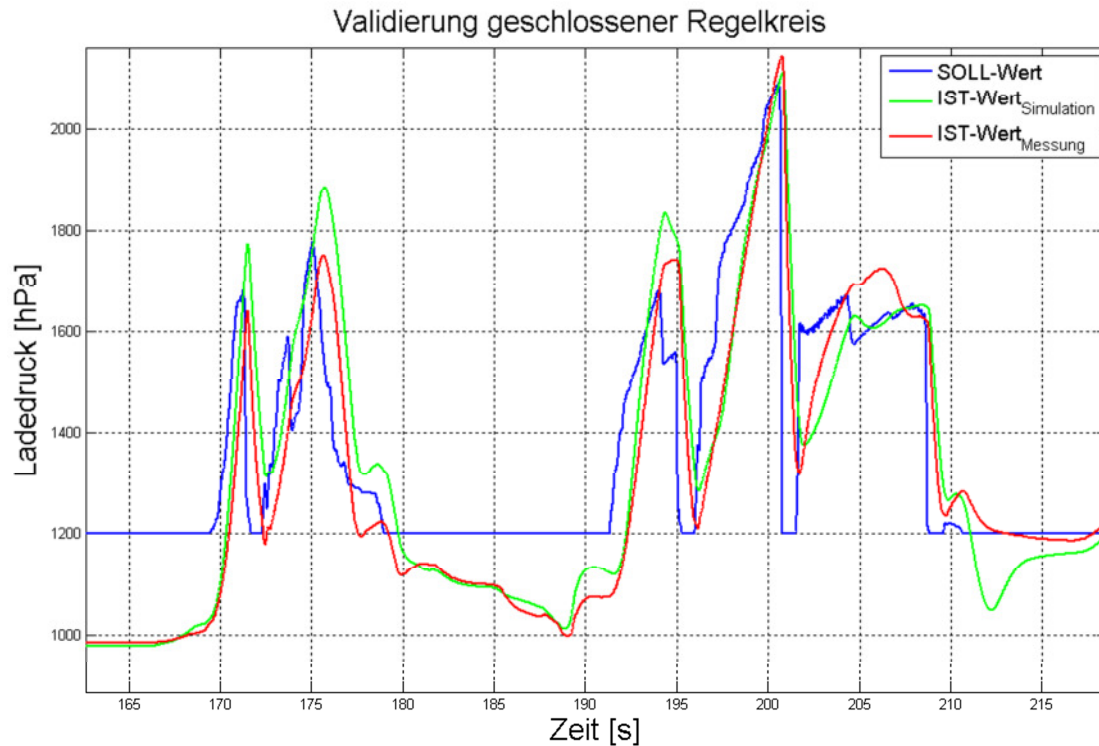


Abbildung 4.7: Validierung geschlossener Regelkreis

In Abbildung 4.7 ist vor allem der simulierte (grüne Linie) und der gemessene Istwert (rote Linie) wichtig. Der Vollständigkeit halber wird der Sollwert (blaue Linie) ebenfalls dargestellt, um zu sehen, wohin sich der Istwert bewegen soll. Zu beachten ist, dass es eine Abweichung zwischen dem simulierten Istwert des Motormodells und dem gemessenen Istwert des realen Motors gibt. Die Rückführung des simulierten Istwerts führt daher zu einer anderen Regelabweichung und infolge dessen zu einem anderen Verhalten der ECU und der Wastegate Position. Der Vorsteuerwert (siehe Kapitel 4.3.2) musste ebenfalls für das Modell neu berechnet werden, damit der stationäre Endzustand keine bleibende Regelabweichung aufweist. Man sieht aber, dass die Dynamik des Modells sehr gut mit der des realen Motors überein stimmt.

4.5 Modellbasierte Regleroptimierung

In diesem Kapitel wird darauf eingegangen, wie der Differential Evolution (DE) Algorithmus mit dem Modell verknüpft wird und welche Einstellungen für den Algorithmus vorgenommen wurden.

4.5.1 Programmablauf

Der Optimierungsalgorithmus simuliert den Regelkreis bestehend aus der virtuellen ECU und dem Motormodell (siehe Abbildung 4.8). Als Fahrzyklus dienen reale Messdaten, welche in das Motormodell und die nachgebildete ECU eingespeist werden. Der Algorithmus wird als Skript File aufgerufen, in welchem das Simulink Modell (ECU und Motormodell) simuliert wird. Der Non-Dominated-Sorting Differential Evolution (NSDE) übergibt die variierenden Optimierungsparameter (K_{pPos} , K_{iPos} , K_{pNeg} , K_{iNeg}) und erhält nach der Simulation die Zielfunktionen (f_1 , f_2 , f_3 , f_4) aus der Systemreaktion.

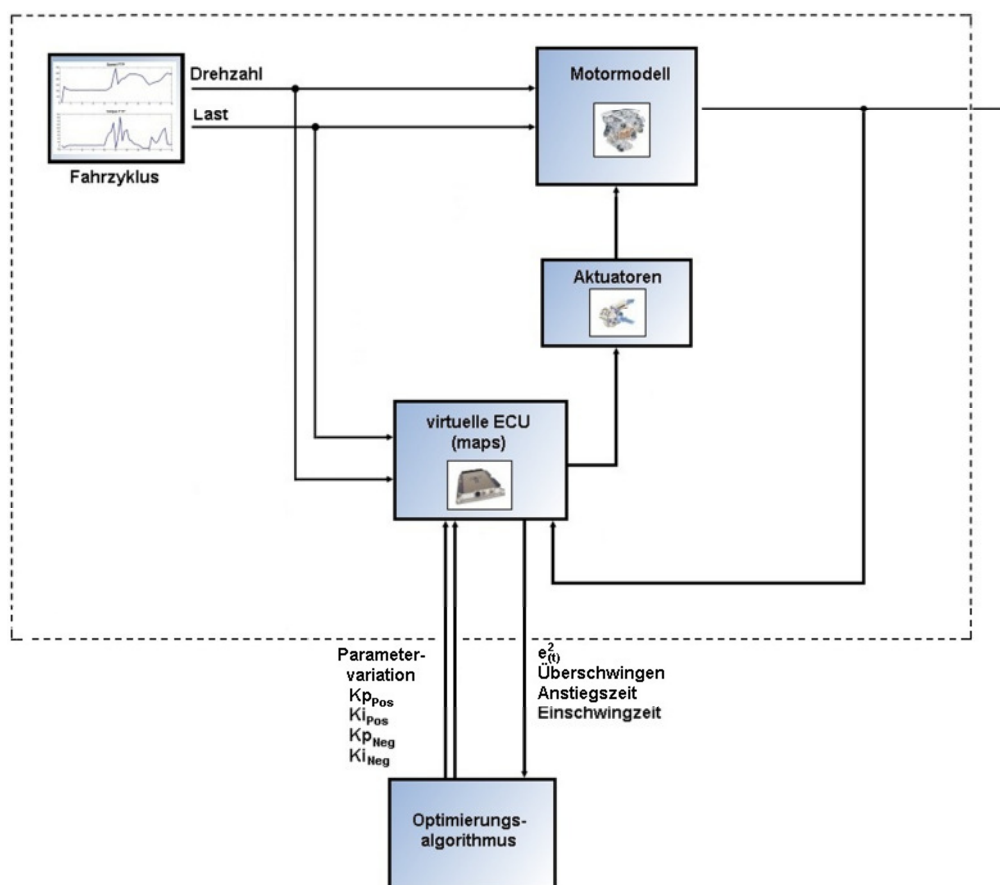


Abbildung 4.8: Regelstrecke

Zur besseren Darstellung des Programmablaufs zeigt Abbildung 4.9 ein Nassi-Shneiderman Struktogramm.

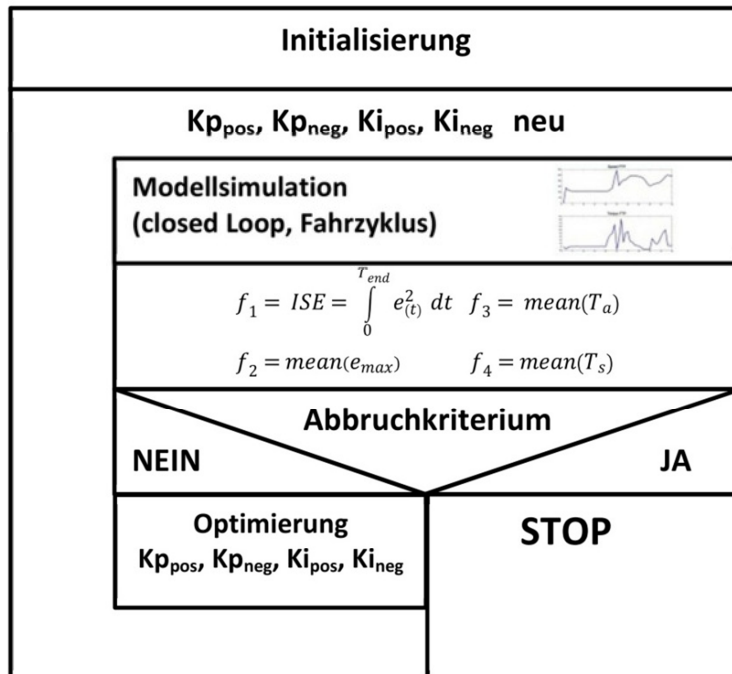


Abbildung 4.9: Programmablaufplan

Während der Initialisierung werden die benötigten Modellparameter von der ECU und dem Motormodell geladen und die Einstellungen des Differential Evolution (DE) eingerichtet. Danach beginnt der Optimierungsalgorithmus, wie in Kapitel 3.3.1 beschrieben. Hierbei werden die Optimierungsparameter ($K_{p_{pos}}, \dots$) der Modellsimulation übergeben. Die Reaktion des Systems mit den neuen Parametern wertet wieder das Skript File aus und es ergeben sich die Zielfunktionswerte. Der Optimierungsalgorithmus endet, sobald ein bestimmtes Abbruchkriterium, in diesem Fall die maximale Anzahl der Generation g_{max} , erreicht wird. Ansonsten werden die Parameter nach dem DE Schema weiter variiert und ein neuer Simulationsdurchlauf beginnt.

4.5.2 Optimierungsalgorithmus

Der Non-Dominated-Sorting Algorithmus wurde genauso implementiert, wie er in der wissenschaftlichen Arbeit von Deb *et al.* [10] des NSGA-II zu finden ist. Der besseren Übersicht halber werden kurz die Einstellungsparameter für den Non-Dominated-Sorting Differential Evolution (NSDE) zusammengefasst.

Optimierungsparameter: 4 (K_{pPos} , K_{iPos} , K_{pNeg} , K_{iNeg})

Grenzen aller Parameter: 0-2

Populationsgröße N_p : 50

Differential Evolution Schema: DE/rand/1/bin

Skalierungsfaktor F : 0.85

Crossoverfaktor Cr : 0.5

Max Generation: 200

Umschaltfenster auf Großsignalbetrieb bei +/- 50 hPa

Zielfunktionen:

$$f_1 = ISE = \int_0^{T_{end}} e_{(t)}^2 dt \dots Integral \ square \ error$$

$$f_2 = mean(e_{max})$$

$$f_3 = mean(T_a)$$

$$f_4 = mean(T_s)$$

5 Experimente und Resultate

Wie bereits in Kapitel 3.1 beschrieben, erfolgt die Optimierung anhand eines Fahrzyklus (Abbildung 3.2). Dieses Fahrmanöver stammt aus realen Messungen, dauert 45 Sekunden und befindet sich in einem Drehzahlbereich von 2000 bis 3700 rpm mit einer sprunghaften Änderung des Drehmoments von 0 auf 160 Nm.

In dem folgenden Kapitel werden die optimierten Parameter diskutiert und deren Auswirkung auf andere Fahrzyklen besprochen. Die Beurteilungskriterien für die Reglerauslegung (Kapitel 3.2) werden ebenfalls eingehalten. Dort werden Überschwinger kleiner 200hPa und Unterschwinger kleiner 100 hPa als annehmbar angesehen.

5.1 Auswahl der optimierten Parameter

Nachdem nun mit Hilfe des multikriteriellen Optimierungsalgorithmus NSDE die 4 Zielfunktionen aus Kapitel 3.2 (Formel 3.2 ... 3.5) minimiert wurden, stellt sich nun die Frage, welche Parameter zur weiteren Betrachtung herangezogen werden. Aus der Menge der Lösungen (Paretomenge) werden die Parameter der Extremwerte (Minima) von den 4 Zielfunktionen als auch das Minimum, welches durch folgende Gütefunktion (Formel 5.1) berechnet wird, näher betrachtet:

$$J = \int_0^{t_{final}} |e_{(t)}| dt + 10 \cdot \text{mean}(e_{max}) + 0,5 \cdot \text{mean}(T_a) + 0,5 \cdot \text{mean}(T_s) \quad (5.1)$$

Diese Gütefunktion gewichtet alle 4 Kriterien und wurde bereits in [8] erfolgreich verwendet.

5.1.1 Multikriterielle Funktionswerte

In den folgenden Abbildungen werden die 4 Zielfunktionen zueinander dargestellt. Es wird die Pareto Front über alle Zielfunktionswerte abgebildet, welche ein Überschwingen kleiner 15% besitzen. Die einzelnen Zielfunktionen (f_1, f_2, f_3, f_4) werden dabei über die Fehlerquadratabweichung (f_i) aufgetragen. Diese Zielfunktion dient als Hauptkriterium. Sollte diese gleich Null (Sollwert = Istwert) sein, so sind die anderen Funktionen ebenfalls optimal.

Die roten Punkte in den Abbildungen (5.1 bis 5.5) sind die Zielfunktionswerte der Pareto Front (siehe Kapitel 3.3.3.1). Man sieht zwar auch Punkte innerhalb der Pareto Front, aber der Grund hierfür ist, dass die Pareto Front über alle Zielfunktionen gebildet wurde und in den Abbildungen jeweils nur 2 Zielfunktionen dargestellt werden. Die farblich hervorgehobenen rechteckigen Punkte sind die Minima der einzelnen Zielfunktionen (f_1, f_2, f_3, f_4) und der Gütefunktion (Formel 5.1).

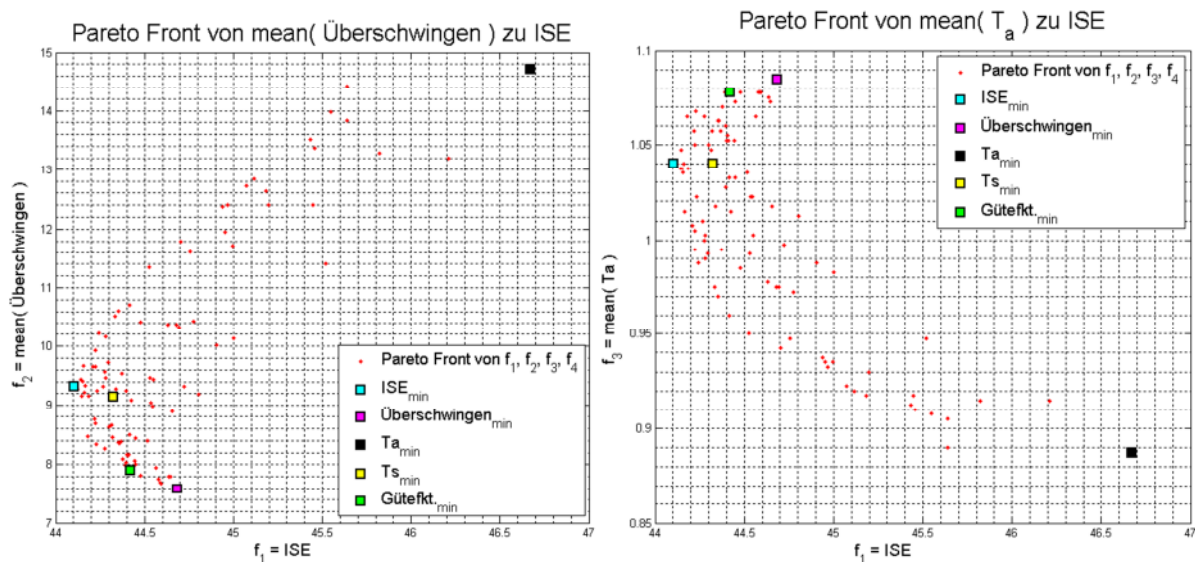


Abbildung 5.1: a.) Pareto Front f_2 zu f_1

b.) Pareto Front f_3 zu f_1

Abbildung 5.1 a) zeigt, wie sich das Überschwingen zu der Fehlerquadratabweichung (engl. Integral Square Error, kurz ISE) verhält. Man sieht, wie die Fehlerabweichung bei einem Überschwingen von 15% bis 9% abnimmt, jedoch bei einem Überschwingen von kleiner 9% wieder ansteigt.

In Abbildung 5.1 b) sieht man ein ähnliches Verhalten mit der Anstiegszeit T_a . Bis zu einer Anstiegszeit von 1,05 Sekunden nimmt das ISE ab und darunter wieder zu. Die minimale Anstiegszeit hat dabei ein doppelt so hohes Überschwingen (15%) wie das kleinste Überschwingen (7,5%). Der Regelkreis wird dabei aber nur um 0,2 Sekunden schneller.

Aus der Theorie (Kapitel 2.2.2 Sprungantwort) ist bekannt, dass sich das Überschwingen indirekt proportional zur Anstiegszeit verhält. Wenn die Anstiegszeit T_a sinkt, wird der Regelkreis schneller, aber das Überschwingen steigt und infolge dessen auch die Fehlerquadratabweichung. Bei einem zu kleinem Überschwingen ist der umgekehrte Effekt zu beobachten. Dort ist der Regelkreis langsamer, die Anstiegszeit daher größer und die Fehlerabweichung steigt ebenfalls an.

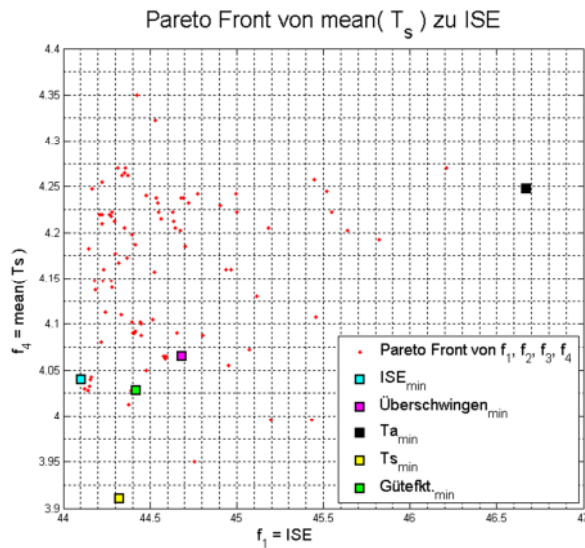


Abbildung 5.2: Pareto Front f_4 zu f_1

In Abbildung 5.2 sieht man, welchen Einfluss die minimale Anstiegszeit Ta bzw. das minimale Überschwingen zur Einschwingzeit hat. Die minimale Anstiegszeit benötigt um 0,2 Sekunden länger als das minimale Überschwingen zum Einschwingen. Durch die kurze Anstiegszeit ergibt sich ein größeres Überschwingen und der Regelkreis benötigt länger zum Einschwingen. Die Fehlerquadratabweichung zwischen minimaler Einschwingzeit Ts und minimaler Überschwingen fällt dagegen nicht so sehr ins Gewicht.

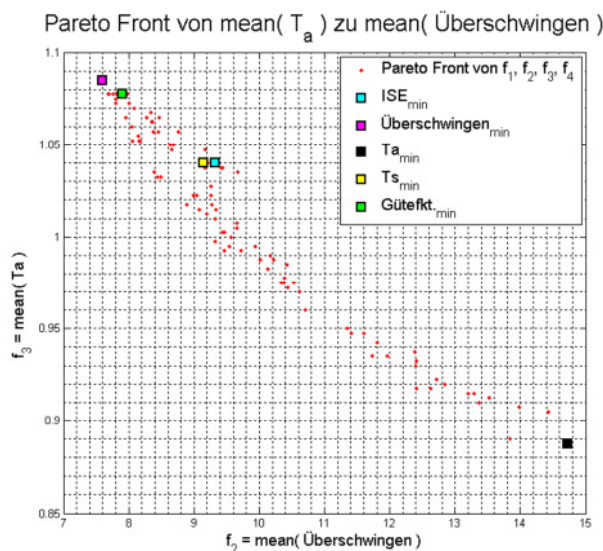


Abbildung 5.3: Pareto Front f_3 zu f_2

Das indirekt proportionale Verhalten zwischen Überschwingen und Anstiegszeit spiegelt sich auch in Abbildung 5.3 wider. Bei einem schnellen Regler ist die Anstiegszeit kleiner, aber das

Überschwingen größer und bei einem langsamen Regler verhalten sich die Anstiegszeit und das Überschwingen genau umgekehrt. Die stärkere Gewichtung der Überschwinger in der Gütefunktion (Formel 5.1) ist ebenso ersichtlich. Man sieht aus den vorherigen Abbildungen, dass sich eine stärkere Gewichtung des Überschwingens bezahlt macht. Bei einer Halbierung des Überschwingens von 15% auf 7,5% ergibt sich ein um 0,2 Sekunden schnellerer Regelkreis. Die Gewichtung der Gütefunktion (Formel 5.1) ist daher ein guter Kompromiss zwischen den einzelnen Beurteilungskriterien Überschwingen, Anstiegszeit und Einschwingzeit.

5.1.2 Auswahl der optimierten Parameter

In den folgenden Abbildungen werden die optimierten Parameter zu der Fehlerquadratabweichung aufgetragen und diskutiert.

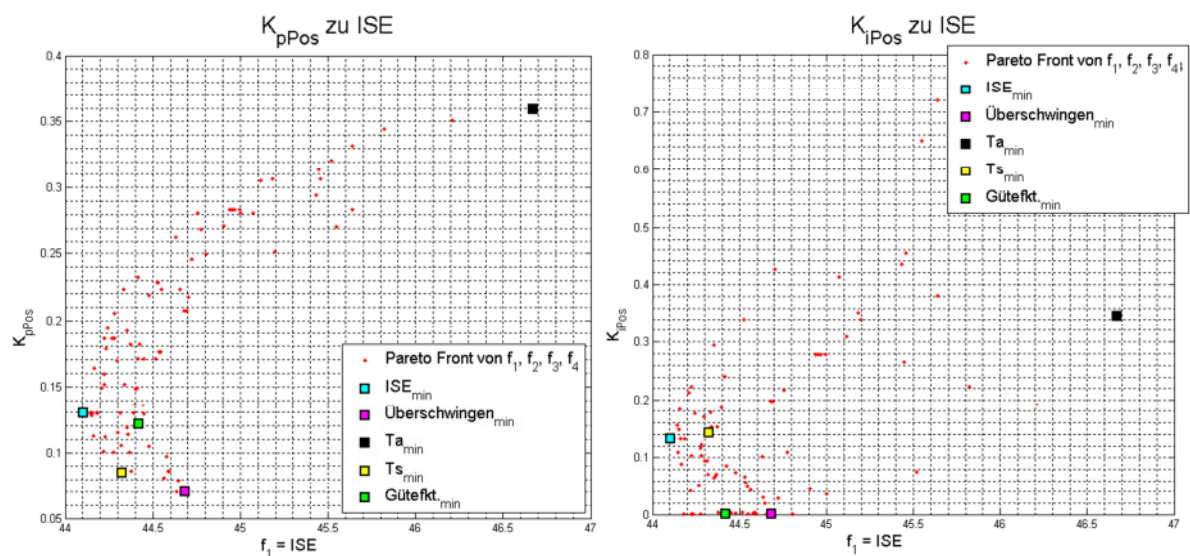


Abbildung 5.4: a.) $K_{p\text{Pos}}$ zu ISE

b.) $K_{i\text{Pos}}$ zu ISE

In Abbildung 5.4 a) sieht man, welchen Einfluss der P-Anteil auf den Regelkreis ausübt. Bei einem größeren $K_{p\text{Pos}}$ ergibt sich ein schneller Regelkreis, daher ist die Anstiegszeit kleiner, aber auch das Überschwingen größer. Aus diesem Grund tritt auch das kleinste $K_{p\text{Pos}}$ bei dem minimalen Überschwingen auf.

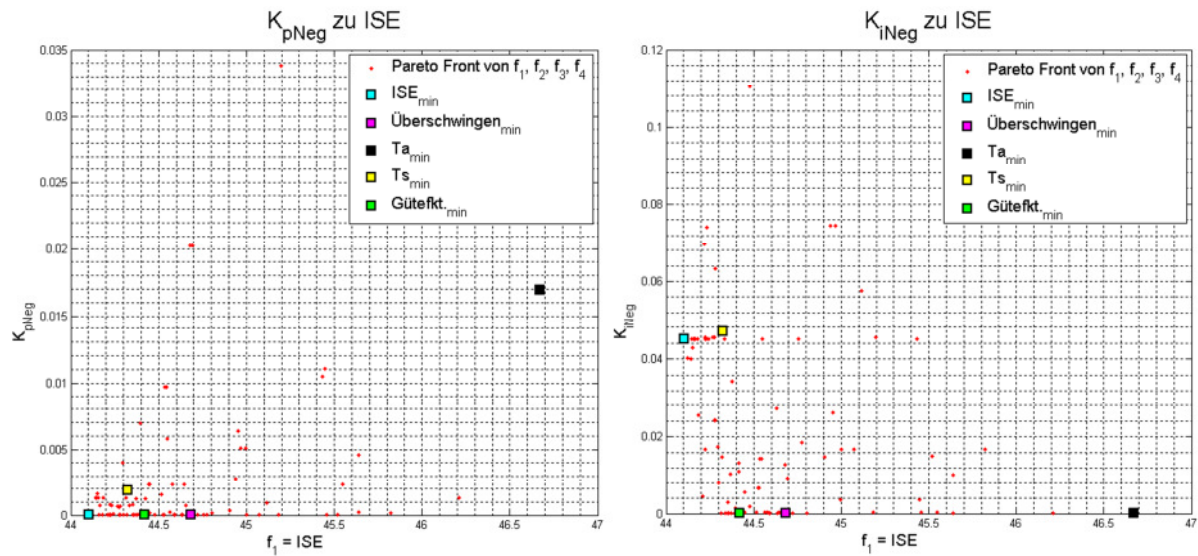


Abbildung 5.5: a.) K_{pNeg} zu ISE

b.) K_{iNeg} zu ISE

Bei den negativen Großsignalparametern zeigt sich ein ähnliches Verhalten wie bei den positiven. Auch hier ist der P-Anteil ausschlaggebend für die Geschwindigkeit des Regelkreises.

Tabelle 5.1 zeigt nun die Kleinsignalparameter, welche mit der LQR Methode optimiert wurden. Diese Parameter sind vom Betriebsbereich abhängig, aber wie man aus Kapitel 4.3.3, Abbildung 4.4 erkennen kann, werden sowohl die Kleinsignalparameter als auch die Großsignalparameter mit dem Drehzahl/Last Kennfeld multipliziert. Das heißt, es müssen nur die einzelnen Werte miteinander verglichen werden, um die Ergebnisse zu diskutieren. Tabelle 5.2 zeigt sowohl die optimierten Werte als auch die anfangs eingestellten Parameter aus der ECU.

Kleinsignalwerte	K _p	K _i
	0.464025	0.552586

Tabelle 5.1: Kleinsignalparameter

Funktionswerte	K _{pPos}	K _{iPos}	K _{pNeg}	K _{iNeg}
Anfangswerte	0.464025	0.552586	0.464025	0.552586
min Gütefunktion	0.121069	0.001030	0.00005721	0.00001263
min ISE	0.129322	0.1328	0.00005816	0.0449263
min Überschwingen	0.0697056	0.0010307	0.0000005	0.00000535
min T _a	0.35896	0.346245	0.016944	0.00002478
min T _s	0.084955	0.142439	0.001886	0.04688316

Tabelle 5.2: Großsignalparameter

Wenn man die Großsignal- und Kleinsignalparameter miteinander vergleicht, so ist zu erkennen, dass die eingestellten Parameter identisch mit den Kleinsignalparametern sind, da bei diesen noch keine Optimierung stattfand.

5.2 Anwendung der optimierten Parameter

Um zu sehen, ob sich eine Verbesserung durch die optimierten Parametersätze ergibt, werden die verschiedenen Parameter der Minima (f_1 , f_2 , f_3 , f_4 , Gütefunktion) mit den anfangs eingestellten Parametern aus der ECU verglichen. Hierzu werden wieder die ersten 1,5 Sekunden der Messdaten zum Erlernen des Motormodells genutzt.

5.2.1 Optimierten Zyklus

Als erster Fahrzyklus wird der verwendet, welcher bereits zur Optimierung herangezogen wurde (siehe Kapitel 3.2, Abbildung 3.2).

Beschreibung des Zyklus:

In Abbildung 5.6 sieht man den gesamten Fahrzyklus, von welchem die ersten 50 Sekunden zur Optimierung verwendet wurden. Der vergrößerte Ausschnitt zeigt den Ladedruck in der Zeit von 16 bis 25 Sekunden, in der eine Vollgasbeschleunigung („TipIn“) stattfand. Der Teilausschnitt befindet sich in einem Drehzahlbereich von 2250 bis 3250 rpm bei einer sprunghaften Drehmomentänderung von 0 auf 160 Nm. Der untere Teil in Abbildung 5.6 zeigt, wie sich das Wastegate verhält. Es öffnet sich zwischen 30 und 55% und ist im stationären Bereich zu ca. 40% offen.

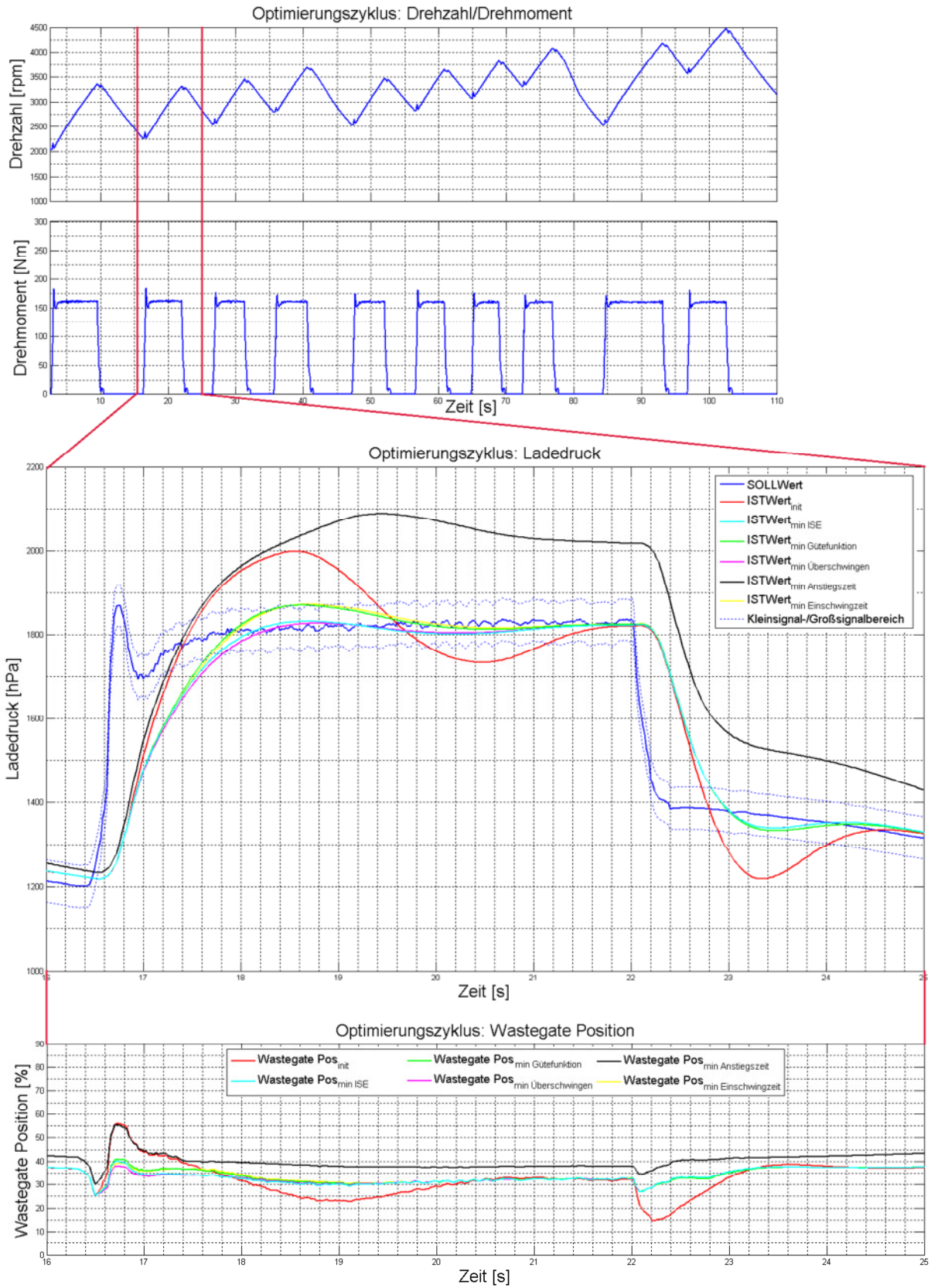


Abbildung 5.6: Optimierungszyklus

Beschreibung der Signale:

Die blaue Linie in der mittleren Abbildung 5.6 zeigt den Ladedruck Sollwert aus dem Basiskennfeld, welcher sich bis auf 1800 hPa erhöht und aus einer realen (verrauschten) Messung stammt. Die blau strichlierte Linie rund um den Sollwert soll die Grenzen zwischen Kleinsignal- und Großsignalbereich veranschaulichen. Die anderen Farben stellen die simulierten Istwerte mit den entsprechenden Parametersätzen dar. Man sieht, gleich nach der sprunghaften Änderung des Ladedruck Sollwerts befindet sich der Istwert in dem Großsignalbereich und die optimierten Werte werden aktiv.

Verhalten der Signale:

Die Anfangsparameter erzwingen in der Simulation ein Überschwingen von 200 hPa und sind deshalb nicht mehr zufriedenstellend. Wie erwartet haben die Parameter des minimierten Überschwinger Funktionales f_2 das kleinste Überschwingen, aber dafür auch die längste Anstiegszeit. Man sieht, dass sich das Überschwingen stark auf die Fehlerquadratabweichung (ISE) auswirkt, da das minimale Überschwingen fast gleich ist dem minimalen ISE. Das minimale ISE ist etwas schneller als das minimale Überschwingen, hat aber dafür ein klein bisschen mehr Überschwingen, was man ebenfalls aus der Abbildung 5.1 erkennen kann. Dort liegt das minimale Überschwingen nahe an dem minimalen ISE, welches schneller ist, aber dafür mehr Überschwingen aufweist. Der Istwert der minimalen Gütefunktion ist der schnellste von den drei Extremwerten (Überschwingen, ISE und Einschwingzeit) und mit einem Überschwingen von ca. 50 hPa als annehmbar zu bewerten. Die minimierte Anstiegszeit ist von allen am schnellsten, aber sie schwingt nicht ein und ist mit einer Abweichung von ca. 300 hPa nicht zu gebrauchen.

Das Regelverhalten lässt sich auch anhand der Tabelle 5.2 erklären. Wenn es darum geht, ein System schnell zu machen, so ist der K_p Anteil dominant. Mit einem zunehmenden K_p wird der Regelkreis schneller gemacht. Eine große Integrationskonstante K_i führt dazu, dass auf eine Regeldifferenz rasch reagiert wird, somit aber auch größere Überschwinger möglich sind. Man kann aber nicht sagen, dass die Parametersätze mit dem größten K_p den Regler am schnellsten machen, da die einzelnen Parameter K_p , K_i , K_d voneinander abhängig sind.

Man sieht auch die Abhängigkeit der Kleinsignalparameter von den Großsignalparametern. Sind zum Beispiel die positiven Großsignalparameter zu groß, dann landet der Istwert woanders im Kleinsignalbereich, als wie wenn sie zu klein wären. Im Extremfall könnte es sogar vorkommen, dass der Istwert direkt von den positiven Parametern zu den negativen springt.

Wenn man Abbildung 5.2 noch einmal betrachtet, so sieht man, dass bei der minimalen Anstiegszeit eine Einschwingzeit von ca. 4,25s benötigt wird. In Abbildung 5.6 wurde zwar der eingeschwingene Zustand nicht erreicht, aber dafür das Ende des Sprunges. Hier sieht man den Nachteil der Zyklus Optimierung. Wenn man auf den Mittelwert optimiert (siehe Kapitel 3.2), so kann es vorkommen, dass 3 der 4 Einschwingzeiten von den TipIn's in Ordnung sind, der letzte aber nicht und der Parametersatz trotzdem verwendet wird. Umgekehrt ist es auch, wenn man das Maximale der 4 Einschwingzeiten zum Optimieren hernimmt. Die schlechte Einschwingzeit eines einzelnen TipIs's kann hierbei alle anderen Einschwingzeiten schlechter aussehen lassen, als sie in Wirklichkeit sind.

5.2.2 Robustheitsuntersuchung

Da es sich um einen stochastischen Optimierer handelt, wurde auch eine Robustheitsuntersuchung durchgeführt. Ziel dieser Untersuchung ist es zu sehen, ob eine Verbesserung nur genau in den zufällig optimierten Parametern auftritt oder es auch eine Verbesserung bei einer Abweichung dieser Parameter gibt. Die ECU ist eine digitale Einheit und besitzt je nach Datentyp nur eine endliche Anzahl an Nachkommastellen. Bei einer Änderung der hintersten Nachkommastellen (10^{-4}) durch die Optimierung kann es also vorkommen, dass man diese Änderungen in der ECU gar nicht mehr einstellen kann.

Für die nachfolgenden Validierungen wurden deshalb die Parameter, welche durch die minimale Gütefunktion (Formel 5.1) gefunden wurden, mit einer Abweichung von 0,04 zufällig variiert. Insgesamt wurden 200 Simulationen durchgeführt, in denen die 4 Parameter (K_{pPos} , K_{iPos} , K_{pNeg} , K_{iNeg}) zufällig abwichen. Die Parametergrenzen mit 0 und 2 werden dabei trotzdem eingehalten.

5.2.3 Validierung

Bei den folgenden Abbildungen (5.7 bis 5.9) werden nun die Istwerte des Ladedrucks der Parametersätze

- 1.) Anfangsparameter aus der ECU,
- 2.) optimierte Parameter bei der minimalen Gütefunktion,
- 3.) und den variierten Parametern von 2.)

dargestellt, um zu sehen, ob eine Verbesserung nicht nur im Optimierungszyklus, sondern auch in anderen Fahrzyklen zu erkennen ist und ob diese Verbesserung auch bei Parametern auftritt, die etwas von den Optimierten abweichen.

Beschreibung der Zyklen:

Der Validierungszyklus 1 (Abbildung 5.7) beinhaltet Drehzahlen von 1500 bis 4500 rpm und eine sprunghafte Änderung des Drehmoments von 0 auf 100 Nm und eine nochmalige Erhöhung auf 160 Nm ohne das Gaspedal loszulassen. Bei einem maximalen Drehmoment von 300 Nm des Motors befindet man sich bei diesem Zyklus im mittleren Drehmomentbereich. Der vergrößerte Teilausschnitt zeigt eine zweimalige Steigerung der Drehzahl von 1500 auf 3750 rpm bzw. von 1750 bis 4000 rpm.

Der gesamte Validierungszyklus 2 (Abbildung 5.8) spielt sich ebenfalls im oberen Drehzahlbereich (1500-4500 rpm) ab. Bei dem vergrößerten Teilausschnitt ist zuerst eine Drehzahländerung von 2350 bis auf 3100 rpm zu sehen und dann eine nochmalige von 2400 bis auf 4500 rpm. Vor allem der steile Anstieg der zweiten Drehzahlsteigerung lässt darauf schließen, dass man sich in einem höheren Drehmomentbereich befindet. Die Drehzahländerung springt zuerst auf 125 Nm danach wird das Gaspedal losgelassen und erst dann erfolgt eine zweifache sprunghafte Änderung auf 125 Nm und von dort auf 225 Nm (ähnlich dem Validierungszyklus 2).

Der genormte New European Driving Cycle (kurz: NEDC) ist der letzte Zyklus an dem getestet wird, ob die optimierten Parameter auch bei anderen Fahrzyklen zu gebrauchen sind. Abbildung 5.9 zeigt einen Teil aus dem insgesamt 20 Minuten dauernden NEDC. Dieser Fahrzyklus besteht aus einem städtischen Bereich und einem außerstädtischen Bereich und dient zur Ermittlung des Kraftstoffverbrauchs des Fahrzeugs.

Beschreibung der Signale:

In allen 3 Abbildungen (5.7 bis 5.9) ist der Ladedruck Sollwert blau und die Grenze zwischen Kleinsignal- und Großsignalbereich blau strichliert dargestellt. Damit ist die Umschaltung zwischen Kleinsignalparameter und Großsignalparameter leichter zu erkennen. Die grüne Linie ist der simulierte Ladedruck Istwert, welcher zustande kommt, wenn man die optimierten Parameter (Tabelle 5.2) aus dem Minimum der Gütefunktion (Formel 5.2) in das nachgebildete Motorsteuergerät lädt. Die türkise Linie zeigt den Istwert aus der Robustheitsuntersuchung. Hierbei wurden die Parameter aus der minimalen Gütefunktion mit einer Abweichung von 0,04 zufällig variiert. Als Parametersatz wurde dann derjenige genommen, welcher die größte Fehlerquadratabweichung zu dem Sollwert aufweist. Es wurden also die Werte aus der Robustheitsuntersuchung genommen, welche die größte Abweichung zu dem Sollwert aufweisen.

Ergebnisse:

Bei allen 3 Abbildungen (5.7 bis 5.9) sieht man eine Verbesserung des Istwerts der optimierten Parameter (grün) zu den anfangs eingestellten Parametern (rot). Selbst der Parametersatz aus der Robustheitsuntersuchung ist nur unwesentlich schlechter als der optimierte Parametersatz. Diese zwei Parametersätze (grün, türkis) besitzen fast immer ein Überschwingen kleiner 50 hPa und erfüllen daher die Anforderungen des Applikateurs aus Kapitel 3.2. In Validierungszyklus 2 fällt auf, dass bei einer sprunghaften Drehmomentänderung auf 100 Nm sich der Ladedruck Sollwert nur rampenförmig ändert und nicht wie bei den anderen Zyklen sprunghaft. Daher kommt es auch nur kurz zu der Aktivierung der Großsignalparameter. Bei dem NEDC Fahrzyklus ist in dem oberen Drehzahlbereich (> 1500 rpm) eine Verbesserung zu sehen, jedoch hat sich gezeigt, dass die optimierten Parameter im unteren Bereich schlechter waren als die anfangs eingestellten Parameter. Der Grund hierfür ist, dass der Optimierungszyklus nur den oberen Drehzahlbereich abdeckt.

Man sieht auch sehr gut, dass die optimierten Parameter nur kurz (gleich nach dem Sollwert Sprung) aktiv sind und dann in dem Kleinsignalbereich landen, wo die fix vorgegebenen bereits optimierten Parameter (LQR Methode) das Einschwingen usw. bestimmen. Die optimierten Großsignalparameter beeinflussen daher das Kleinsignalverhalten und daher auch die Regelung im Kleinsignalbetrieb. Zum Beispiel erfolgt die Berechnung des neuen I-Anteils durch die aktuelle Regelabweichung (gewichtet mit der Integrationskonstante K_i) und der alten Abweichung. Bei falsch eingestellten Großsignalparametern kann die alte Regelabweichung zum Beispiel zu groß werden und infolge dessen den neu berechneten I-Anteil im Kleinsignalbereich stark beeinflussen. Im Extremfall kann es auch vorkommen, dass der neu berechnete I Anteil den Ausgang des Reglers über die zweite Grenze hinaus schießen lässt und daher die anderen Großsignalparameter aktiv werden. Man sieht also, dass die Groß- und Kleinsignalparameter sich gegenseitig stark beeinflussen und eine gemeinsame Optimierung aller abhängigen Variablen wahrscheinlich sinnvoll ist.

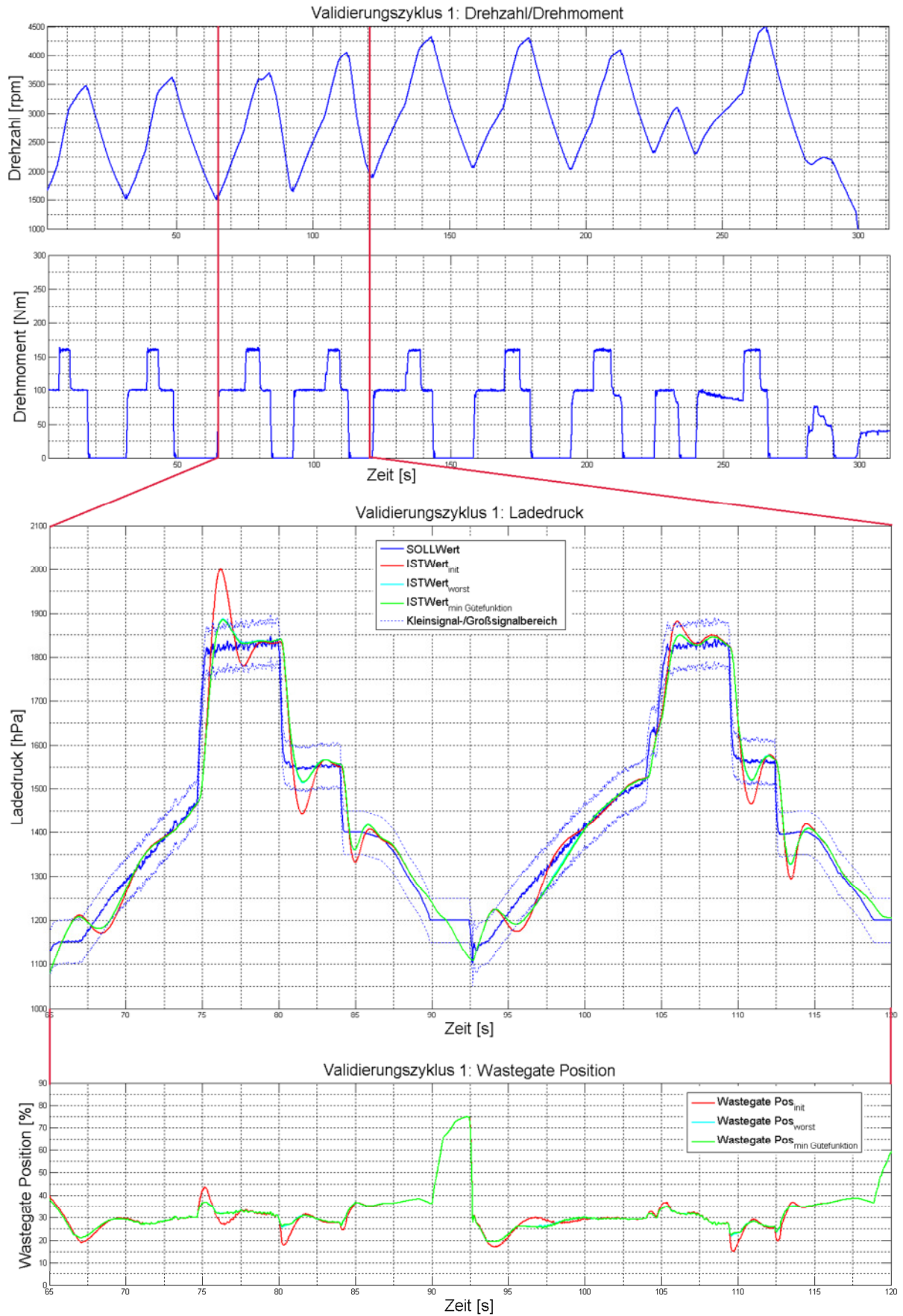


Abbildung 5.7: Validierungszyklus 1

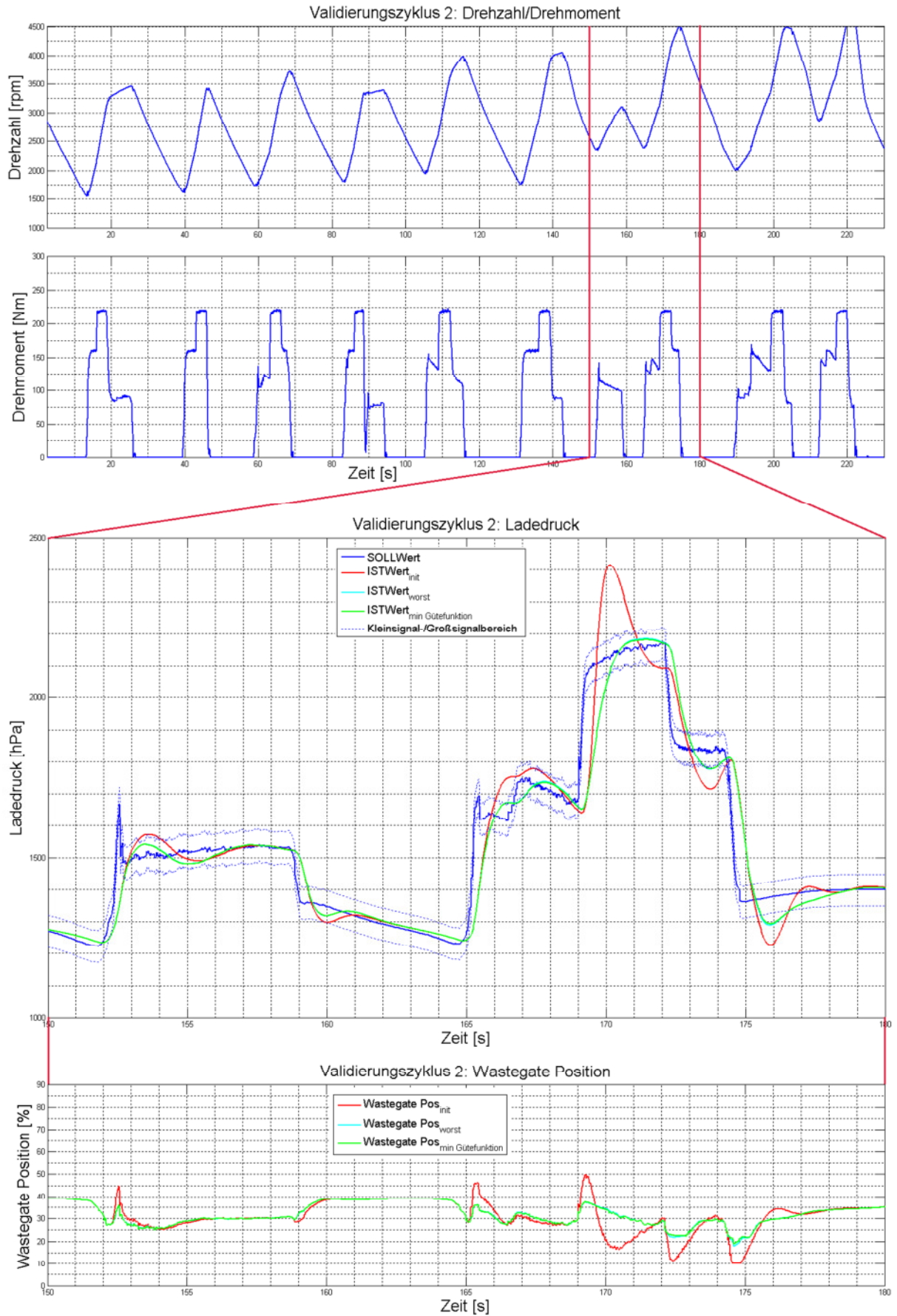


Abbildung 5.8: Validierungszyklus 2

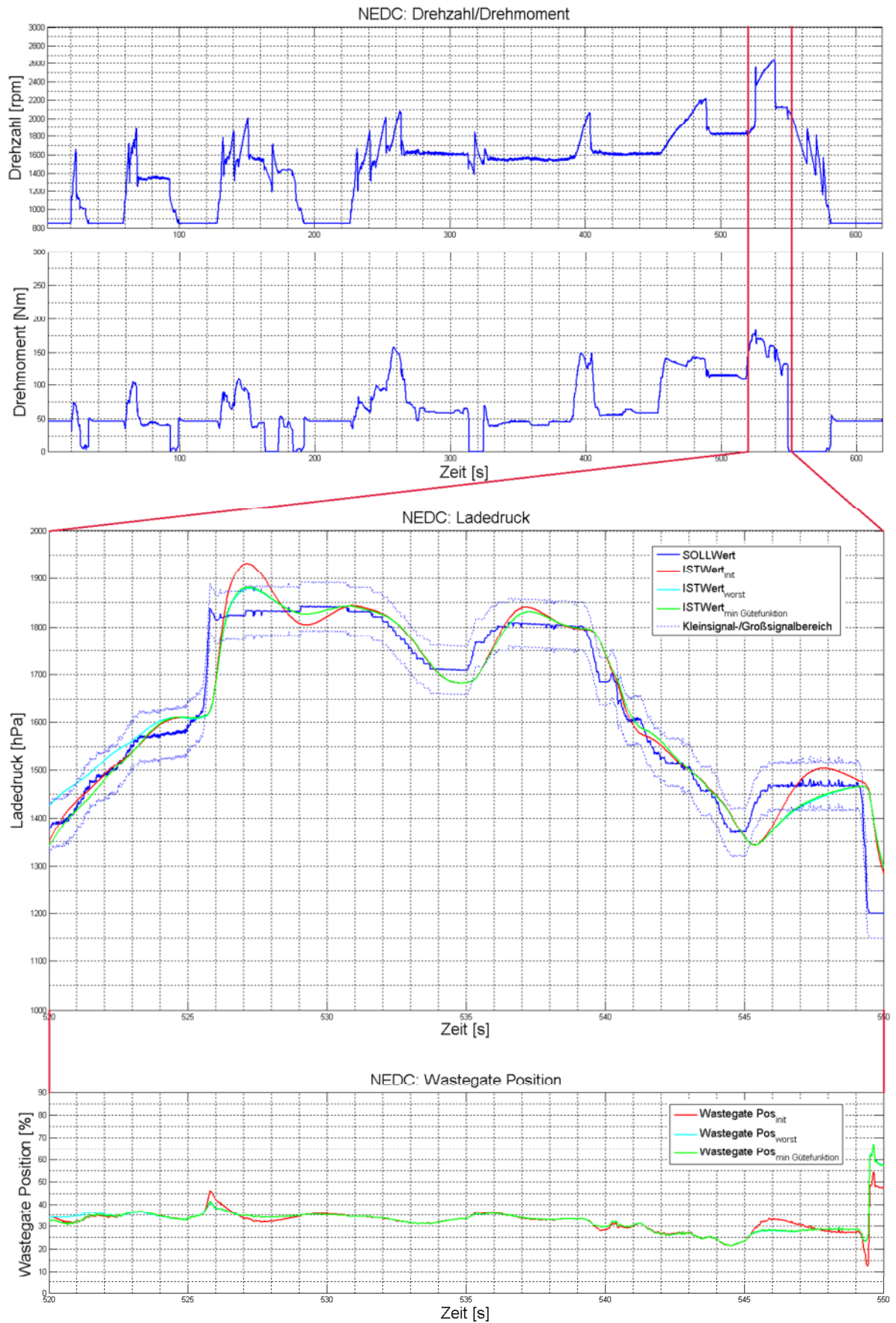


Abbildung 5.9: NEDC

6 Schlussfolgerung und Ausblick

In dieser Forschungsarbeit wurde gezeigt, dass es möglich ist, eine komplexe nichtlineare Strecke, bestehend aus mehreren mathematischen Funktionen, mit Hilfe eines stochastischen Optimierungsverfahrens gegenüber den initial eingestellten Werten zu verbessern. Die nichtlineare Strecke ist hierbei eine neue Modell Architektur, welche als Grundlage die LOLIMOT (LOcal LInear MOdel Tree) Methode verwendet. Auf Basis von APRBS (Amplitude-modulated Pseudo Random Binary Signal) Sequenzen und den Messdaten des realen Dieselmotors wird dieses neuronale Netz erlernt. Als Optimierungsmethode kommt ein stochastischer Optimierer aus den Bereich des Differential Evolution (DE) Algorithmus zum Einsatz. Die Nebenbedingungen, welche für eine akzeptable Reglerkalibrierung nötig sind, werden als zu minimierende Zielfunktionen eingebaut. Um den Differential Evolution auch für mehrere Zielfunktionen (z.B. Überschwingen, Anstiegszeit, Einschwingzeit) zu verwenden, wird dieser mit dem Non-Dominated-Sorting Algorithmus verknüpft. Durch die modellbasierte Regloptimierung ist es möglich, den Optimierungsalgorithmus am Computer laufen zu lassen und die verschiedenen optimalen Parametersätze (Pareto Menge) automatisch zu suchen. Lediglich am Ende des Programms muss der Applikateur einen einzelnen geeigneten Parametersatz wählen und diesen am Fahrzeug testen.

Bei den einzelnen Validierungszyklen hat man gesehen, dass eine Optimierung mit Hilfe eines Zyklus auch eine Verbesserung bei anderen Fahrzyklen mit sich bringt. Aufgrund des stochastischen Optimierers, wurde auch untersucht, ob genau diese zufällig gefundenen Parameter die Regelung verbessern oder ob abgewichene Parameter ebenfalls zu einer Verbesserung beitragen.

Da nur ein einziger Parametersatz in dem Motorsteuergerät (ECU) eingestellt werden kann, ist es wichtig, die einzelnen Beurteilungskriterien (Überschwingen usw.) zu gewichten. Abbildung 5.6 zeigt, welche Unterschiede sich ergeben, wenn man die Gewichte der Gütefunktion falsch wählt bzw. man die Extremwerte (Minima) der einzelnen Kriterien als Parameter auswählt. Bei den Abbildungen 5.1 bis 5.3 sieht man, wie stark der Einfluss einzelner Kriterien auf die Fehlerquadratabweichung ist und welches Kriterium man eventuell stärker bewerten muss. Zum Beispiel wird der Regelkreis um nur 0,2 Sekunden schneller, wenn sich das Überschwingen von 7,5% auf 15% verdoppelt.

Es ist auch möglich, den Differential Evolution Algorithmus durch die Directional Spread Methode [6] oder durch eine Kombination mit der Sequential quadratic programming (kurz SQP) Methode [35] weiter zu verbessern. Die Simulationsberechnungsdauer lässt sich ebenfalls verkür-

zen, indem man die MATLAB Simulink Blöcke als Dynamic Link Library (kurz dll) einbaut und den DE als c File ausführt. Diese Verbesserungen gehen alle in Richtung Konvergenzgeschwindigkeit, also wie schnell man der optimalen Pareto Front näher kommt und wie lange die Berechnungsdauer des Algorithmus ist. Vor allem bei einer hohen Anzahl von Optimierungsparametern (>100) machen diese Verbesserungen Sinn. Für den Applikateur ist es wichtiger, eine größere Auswahl an Werten auf der Pareto Front zu erhalten. Eine große Rolle dabei spielt die Definition der Beurteilungskriterien. Wie soll der Regelkreis bewertet werden, wenn er gerade nicht eingeschungen ist? Bekommt er dieselbe Einschwingdauer wie ein Regelkreis, der eine dauerhafte Schwingung hat? Oder wird die Anstiegszeit als schlecht bewertet, nur weil der Istwert 1% unter dem Sollwert liegt und diesen daher nie vollständig erreicht? Sollten diese „knappen“ Beurteilungskriterien schlechter bewertet werden als sie in Wirklichkeit sind, so fallen die dazugehörigen Parameter aus der Population und aus der Pareto Front raus. Die in dieser Arbeit verwendeten Beurteilungskriterien lassen sich bei sprunghaften Änderungen gut auswerten. Wenn man aber zum Beispiel den NEDC Zyklus optimieren möchte, so muss man die Kriterien neu und vermutlich anders definieren. Bevor man also mit der Verbesserung des Algorithmus beginnt, muss man sich zuerst Gedanken über ein geeignetes Fahrmanöver und über die Definition der Beurteilungskriterien machen. Der in dieser Arbeit verwendete Optimierungszyklus hat die Einschränkung, dass er nur den Drehzahlbereich von 2000 bis 3800 rpm abdeckt. Auch wenn die optimierten Parameter in einem Bereich über 2000 rpm besser sind als die anfangs eingestellten Werte, so sind sie im unteren Bereich schlechter, weil sie dort nicht optimiert wurden. Es ist also notwendig, einen Optimierungszyklus zu finden, welcher den gesamten Betriebsbereich abdeckt, um so ein Optimum zu finden, welches überall eine Verbesserung bringt.

Da in Zukunft Dieselmotoren und Steuergeräte noch komplexer werden und immer mehr Stellgrößen voneinander abhängig sind, ist der Weg Richtung stochastischen Optimierungsverfahren sicherlich der Richtige. Auch wenn in dieser Arbeit nur eine Verbesserung im oberen Drehzahlbereich ersichtlich ist, so sieht man doch, welches Verbesserungspotenzial durch den modellbasierten Ansatz möglich ist. Der Einsatz des modellbasierten Ansatzes erbringt Zeit- und Kostenersparnisse von bis zu 70%, die der Applikateur normalerweise zur Kalibrierung im Fahrzeug aufwendet. Ein weiterer Vorteil des Algorithmus ist die einfache Handhabung und Implementierbarkeit bzw. seine Adaptierbarkeit. Da es sich um einen stochastischen Algorithmus handelt, können auch mehrere Variable, welche zu verschiedenen voneinander abhängigen Systemen (z.B. Turbolader und Abgasrückführventil) gehören, optimiert werden. Solange das Motormodell das richtige Systemverhalten nachbildet, ist dem Optimierer die Abhängigkeit der einzelnen Systeme

egal. Durch die leichte Adaptierbarkeit des Optimierers und der Möglichkeit mehrere verschiedene Zielfunktionen zu minimieren, ist er auch für andere Aufgabenbereiche interessant. So ist es zum Beispiel möglich auf mehrere verschiedene Kompromisse wie Verbrauch vs. Emissionen vs. Leistung einzugehen. Je nach Kundenwunsch oder gewünschtem Fahrverhalten (Sport vs. Eco) kann der Applikateur entscheiden, welche Parameter er schlussendlich aus der Menge der Kompromisslösungen nimmt.

Quellenverzeichnis

Literaturverzeichnis

- [1] Alonso José M., Alvarruiz Fernando, Desantes José M., Hernández Leonor, Hernández Vicente, Germán Moltó (2007): „*Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions*“, IEEE Transactions on Evolutionary Computation, Vol 11, No.1, February 2007
- [2] Alotto Piergiorgio (2011): „*Multiobjective optimization (Pareto)*“, Lecture, TU Graz, November 2011
- [3] Alotto Piergiorgio (2011): „*Single to multiobjective optimization (nondominated sorting)*“, Lecture, TU Graz, November 2011
- [4] Altendorfer Mirjam (2008): „*Pareto-Optimierung komplexer thermo-fluiddynamischer Systeme auf der Basis numerischer Berechnungen*“, Dissertation, TU Erlangen-Nürnberg, 2008
- [5] Antony W. Iorio, Xiaodong Li (2004): „*Solving Rotated Multi-objective Optimization Problems Using Differential Evolution*“, AI'04 Proceedings of the 17th Australian joint conference on Advances in Artificial Intelligence, 2004, pp 861-872
- [6] Antony W. Iorio, Xiaodong Li (2006): „*Incorporating directional information within a differential evolution algorithm for multi-objective optimization*“, in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), 2006, pp 691-698
- [7] Åström Karl J., Hägglund Tore (1995): „*PID Controllers: Theory, Design and Tuning*“, International Society for Measurement and Control, 1995, 2nd edition
- [8] Birnstingl Johannes (2011): „*Controller calibration using a global-dynamic engine model*“, Diplomarbeit, FH Campus 02, 2011
- [9] Das Swagatam, Suganthan Ponnuthurai Nagaratnam (2011): „*Differential Evolution: A Survey of the State-of-the-Art*“, IEEE Transactions on Evolutionary Computation, Februar 2011, Vol. 15 Issue 1, pp 4-31
- [10] Deb K., Amrit Pratap, Sameer Agarwal, Meyarivan T. (2002): „*A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II*“, IEEE Transactions on Evolutionary Computation, April 2002, Vol. 6 Issue 2, pp 182-197
- [11] Deb Kalyanmoy (2009): „*Multi-Objective Optimization Using Evolutionary Algorithms*“, Wiley, Auflage 1, 2009
- [12] Efrén Mezura-Montes, Jesús Velázquez-Reyes, Carlos A. Coello Coello. (2006): „*A Comparative Study of Differential Evolution Variants for Global Optimization*“, In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), 2006, pp 485-492

- [13] Föllinger Otto (2005): „*Regelungstechnik. Einführung in die Methoden und ihre Anwendung*“, Hüthig; Auflage: 8. A. (Juli 2005)
- [14] Golloch Rainer (2005): „*Downsizing bei Verbrennungsmotoren: Ein wirkungsvolles Konzept zur Kraftstoffverbrauchssenkung*“, Springer Berlin Heidelberg; Auflage: 1, 2005
- [15] Hack Gert (2003): „*Turbo- und Kompressormotoren: Entwicklung und Technik: Entwicklung, Technik, Typen*“, Motorbuch Verlag, Auflage 3 (Juli 2003)
- [16] Hafner Michael, Isermann Rolf (2003): „*Multiobjective optimization of feedforward control maps in engine management systems towards low consumption and low emissions*“, Transactions of the Institute of Measurement & Control, März 2003, Vol. 25 Issue 1, pp 57
- [17] Hametner C. (2007): „*Nonlinear Dynamic System Identification Using Local Model Architectures*“, Dissertation, TU Wien, 2007
- [18] Horn Martin, Dourdoumas Nicolaos (2004): „*Regelungstechnik – Rechnerunterstützter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise*“, 2004, Pearson Studium
- [19] Keuth N. (2004): „*An Improved Neuro-Fuzzy Training Algorithm for Automotive Applications*“, Dissertation, TU Wien, 2004
- [20] Kruse T., Kurz S., Lang T. (2011): „*Modern Statistical Modeling and Evolutionary Optimization Methods for the Broad Use in ECU Calibration*“, ETAS ASCMO Whitepaper, Oktober 2011
- [21] Lochau Malte, Sun Bo, Goltz Ursula, Huhn Petra (2009): „*Model-based Parameter Optimization of an Engine Control Unit using Genetic Algorithms*“, Symposium on Automotive/Avionics Systems Engineering (SAASE), Oktober 2009, UC San Diego
- [22] Lunze Jan (2008): „*Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*“, Springer Berlin Heidelberg; Auflage: 5., neu bearb. Aufl. (2. April 2008)
- [23] Mollenhauer Klaus, Tschöke Helmut (2007): „*Handbuch Dieselmotoren*“, Springer Berlin Heidelberg, Auflage 3, 2007
- [24] Nebel Manuel, Vogels Marie-Sophie, Combé Timo, Winsel Thomas, Pfluegl Horst, Hametner Christoph (2010): „*Global dynamic models for xil-based calibration*“, April 2010, SAE International (Society of Automobile Engineers)
- [25] Nebel Manuel (2010): „*Entwicklung einer virtuellen Streckenbeschreibung einer Verbrennungskraftmaschine für Aufgaben der Applikation*“, Dissertation, TU Graz, Oktober 2010
- [26] Nelles Oliver (1997): „*LOLMOT – Lokale, lineare Modelle zur Identifikation nichtlinearer dynamischer Systeme*“, In: Automatisierungstechnik. 45 (1997), Nr. 4, pp 163-174, Artikel, 1997

-
- [27] Nelles Oliver (1999): „*Nonlinear System Identification with Local Linear Neuro-Fuzzy Models*“, Dissertation, TU Darmstadt, 1999
- [28] Pischinger Rudolf, Klell Manfred, Sams Theodor (2009): „*Thermodynamik der Verbrennungskraftmaschine: Der Fahrzeugantrieb*“, Springer Vienna, Auflage 3
- [29] Price Kenneth, Storn Rainer M., Lampinen Jouni A. (2005): „*Differential Evolution: A Practical Approach to Global Optimization*“, Natural Computing Series, Springer Verlag Berlin Heidelberg 2005
- [30] Robert Bosch GmbH (2011): „*Automotive Handbook*“, Wiley, Auflage 8 (Juli 2011)
- [31] Rojas Raul (1996): „*Neural Networks - A Systematic Introduction*“, Springer Berlin Heidelberg, Auflage 1, 1996
- [32] Rückert J., Abel D. (2002): „*Simultane Regelung von Ladedruck und AGR-Rate beim Pkw-Dieselmotor*“, 36. Regelungstechnisches Kolloquium, Februar 2002
- [33] Uday K. Chakraborty (2008): „*Advances in Differential Evolution*“, Studies in Computational Intelligence, Vol. 143, Springer Verlag Berlin Heidelberg 2008
- [34] Vossoughi G.R., Rezazadeh Siavash (2005): „*Optimization of the calibration for an internal combustion engine management system using multi-objective genetic algorithms*“, The 2005 IEEE Congress on Evolutionary Computation, September 2005, Vol. 2, pp 1254 – 1261
- [35] Wenhui Shou, Wenhui Fan, Zhenxia Gao, Boyuan Liu (2011): „*Hybrid Differential Evolution and Sequential Quadratic Programming Algorithm*“, in: Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science, pp 3-6

Verzeichnis der Internetquellen

- [36] „modeFRONTIER“,
http://www.esteco.com/home/mode_frontier/by_industry/automotive.html, 16.5.2012
- [37] „ASCMO“, <http://www.etas.com/de/products/ascmo.php>, 16.5.2012
- [38] „Modellgestützte Prädiktive Regelung“,
<http://www.irt.rwth-aachen.de/29/anwendungen/automotive/diesel/>, 16.5.2012
- [39] „Dieselmotor“, <http://www.lehrerfreund.de/in/technik/1s/dieselmotor/3558>, 16.5.2012
- [40] „MSR“, <http://www.msr-wg.de/megma/index.html>, 18.5.2012