



**Interdisziplinäres Teamprojekt**

# **Parameterschätzung und Entwurf einer Modellprädiktiven Regelung eines autonomen Modellfahrzeuges**

von

Hannes Heinemann  
Matrikelnummer: 184102

André Pieper  
Matrikelnummer: 184960

**20. Februar 2015**

**Betreuer:**

M. Sc. Juan Pablo Zometa \*

M. Sc. Michael Maiworm \*

## **Kurzdarstellung**

In diesem Bericht werden Werkzeuge und Verfahren aufgezeigt, um für ein autonomes Modellfahrzeug im Maßstab 1:10 eine Fahrspurverfolgung, sowie eine Distanzregelung zu entwickeln. Diese Arbeit ist Teil des studentischen Teamprojekts „oTTocar“ der OvGU und wurde unter der Berücksichtigung der Teilnahme an dem internationalen Wettbewerbs „Carolo-Cup“ [2] entworfen. Die Betreuung und Zusammenarbeit erfolgt dabei durch die Fakultäten für Informatik, für Elektrotechnik und Informationstechnik und für Maschinenbau. Die Vorbereitung dieser Arbeit ist maßgeblich an das Forschungsprojekt [15] von Viktoria Wiedmeyer und Andreas Himmel geknüpft, auf das sich das im folgenden verwendete Modell der Fahrzeugdynamik hauptsächlich bezieht.

---

\*Otto-von-Guericke-Universität Magdeburg, Fakultät für Elektro- und Informationstechnik, Institut für Automatisierungstechnik - IFAT, Lehrstuhl für Systemtheorie und Regelungstechnik, Prof. Dr.-Ing. Rolf Findeisen, Universitätsplatz 2, 39106 Magdeburg

---

**Inhaltsverzeichnis**

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Parameterschätzung</b>	<b>3</b>
2.1	Messbare Parametern . . . . .	4
2.2	Bestimmung von Parametern mithilfe von Schätzverfahren . . . . .	5
	<b>Literaturverzeichnis</b>	<b>10</b>

## 1 Einleitung

Im Zuge der „digitalen Revolution“<sup>1</sup> die durch die rasante Entwicklung der Halbleitertechnologien (beschrieben durch das mooresche Gesetz<sup>2</sup>) ermöglicht wird, übernehmen Computer immer mehr Aufgaben in nahezu allen Bereichen einer modernen Gesellschaft. Neben der industriellen Nutzung finden sich immer mehr Computer in Form mobiler Geräte im Alltag eines jeden Menschen wieder und sie nehmen dabei dem Nutzer viele einfache Aufgaben ab, wie z.B. die Navigation auf einer Straße zu einem Ziel. Durch die steigende Leistungsfähigkeit der Rechensysteme wäre der nächste Schritt auch die höheren Aufgaben auf die Maschinen zu verteilen und somit den Menschen weiter zu entlasten. Eine Aufgabe ist dabei das Führen eines Fahrzeuges ohne menschliches Handeln. Das autonome Fahren ist hierbei keine Fiktion mehr, wie in Filmen wie „I, Robot“ [1], oder „Demolition Man“ [8] suggeriert wird, sondern der Prozess ist schon soweit fortgeschritten, dass der autonome Straßenverkehr in naher Zukunft Realität wird. Dafür spricht zum einen, dass schon die ersten autonomen Fahrzeuge des „EUREKA-PROMETHEUS-Projekts“ [5] vor gut 20 Jahren weit mehr als 1758 Km auf öffentlichen Straßen zurückgelegt haben. Und zum anderen, dass in den USA erste autonome Fahrzeuge für den Straßenverkehr zugelassen [6] und in Europa zumindest dafür neue Gesetze entworfen werden [7]. So bleibt nur noch die Frage, wann sich der Straßenverkehr auf das autonome Fahren umstellt und wie die Lösung am Ende aussieht.

Vor diesem Hintergrund findet der Carolo-Cup in Braunschweig seit nun mehr acht Jahren statt, welcher studentische Teams aller Fachrichtungen und Universitäten in einem Wettbewerb gegeneinander antreten lässt, um das beste Konzept und die beste Umsetzung eines autonomen Fahrzeuges zu präsentieren. Die Otto-von-Guericke-Universität Magdeburg beteiligt sich ebenfalls an diesem Wettbewerb mit dem Projekt „oTToCAR“. Die einzelnen Disziplinen sind Einpark-, Spurverfolgungs- und Hindernisszenarien mit Fahrspurwechsel, wofür neben der Hardware- und Software-Entwicklung auch ein Regelungskonzept entwickelt werden muss. Die vorliegende Arbeit beschäftigt sich mit einer modellbasierten Regelung für alle Szenarien und das dafür verwendete Modell mit der nötigen Parameterschätzung. In einer vorangegangenen Arbeit [15] wurde bereits ein Modell entwickelt, deren Parameter jedoch aufgrund eines fehlenden realen Fahrzeuges nicht bestimmt werden konnten. Die jetzige Arbeit ist zeitlich später einzuordnen, in der ein fertiger Prototyp bereits zur Verfügung stand und die Parameterschätzung vollendet werden konnte.



Abb. 1: Ausschnitt vom Parcours im Carolo-Cup [3]

## 2 Parameterschätzung

Um die unbekannten Parameter eines Modells schätzen zu können, wird zunächst ein valides Modell und einen Experimentierstand benötigt. Das Modell wird dem Bericht oTToCAR [15, Seite 12] entnommen, welches dort

---

<sup>1</sup>Ähnlich der industriellen Revolution vor 200 Jahren, eine tiefgreifender Wandel von allen gesellschaftlichen Bereichen, wie Politik, Wirtschaft und Kultur. Hervorgerufen durch die Entwicklung von Computern und deren enormen Verbreitung, wird die heutige Zeit auch als „Zweite Moderne“ bezeichnet

<sup>2</sup>Formuliert 1965 von Gordon Moore, der eine Verdopplung der Anzahl von Schaltkreiskomponenten auf einem integrierten Schaltkreis alle 12 bis 24 Monate vorhersagte und damit bis heute Recht bewies.

auch schon auf seine grundlegende physikalische Korrektheit mit Simulationen überprüft wurde. Das gegebene Modell besitzt dabei zwei Arten von Parametern. Zum einen sind es direkt bestimmbare Parameter, zum anderen sind es Parameter die nur durch modellbasierte Schätzverfahren ermittelt werden können. Um die Simulation mit dem realen Fahrzeug in seinem Verhalten vergleichen zu können, werden dabei Messungen von allen Zuständen benötigt. Da diese Experimente jedoch nicht immer einfach zu realisieren sind und um den technischen Aufwand der Messungen so gering wie möglich zu halten, sollten die zu messenden Zustände beschränkt und so gewählt werden, dass:

- sie charakteristisch für das Verhalten sind
- durch sie andere Zustände bestimmt werden können
- die wichtigsten Regelgrößen auch direkt gemessen werden

## 2.1 Messbare Parametern

Die direkte Bestimmung von Parametern erfolgt weitgehend durch die Messung der selbigen, oder deren Berechnung durch weiteren messbaren Größen. Im Falle der Eingangsgröße  $u_1$ , dem Einschlagswinkel der Räder, musste zum Beispiel das Ansteuerungssignal des Servomotors in einen Winkel für die Räder umgerechnet werden. Das Ansteuerungssignal ist hierbei ein 256-stufiges PWM-Signal, welches in eine Winkelangabe umgerechnet werden muss. Für die Versuchsanordnung wurde das Fahrzeug zuerst so erhöht, dass die Räder nicht mehr auf dem Boden auflagen. Diese Vorgehensweise war nötig, da bei Kontakt der Räder mit dem Boden bei einem stehenden Fahrzeug der Haftwiderstand so hoch ist, dass sich eine kleinere Winkeländerung ergeben und die Messung verfälschen würde. Denn bei schneller Fahrt verringert sich dieser Widerstand auf nahezu Null und kann somit vernachlässigt werden. Für die Messung wurde längs an das linke Vorderrad eine Verlängerung befestigt und für jede Servoeinstellung eine Winkelauslenkung gemessen und die Kalibrierung des Servomotors vorgenommen.

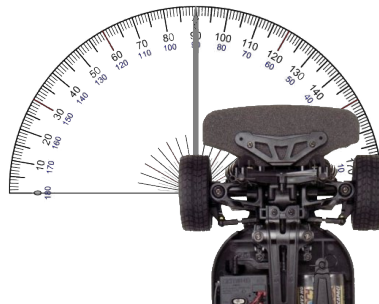


Abb. 2: Messung der Relation PWM-Stufe des Servos zum Einschlagswinkel der Räder (aus [14] und [12])

Als Ergebnis dieses Experimentes kam heraus, dass die Erhöhung um eine PWM-Stufe des Servomotors konstant eine Drehung der Räder um  $0.0028 \text{ rad}$  entspricht. In der Tabelle 1 aufgeführten Größen wurden hingegen direkt am oTTToCAR gemessen, bzw. das Trägheitsmoment aus einem CAD-Modell [13] berechnet.

Physikalische Größe	Exakter Wert	Einheit
Dichte der Luft	$\rho_L = 1,204$	$[\text{kg s}^{-1}]$
Erdbeschleunigung	$g = 9,806$	$[\text{m s}^{-2}]$
Masse	$m = 2,8$	$[\text{kg}]$
Radabstand	$l = 0,257$	$[\text{m}]$
Radius des Rades	$r = 0,0335$	$[\text{m}]$
Übersetzungsfaktor Drehzahl Motor $\rightarrow$ Räder	$\epsilon = 5,52$	$[-]$

Tabelle 1: Direkt bestimmbare Parameter

## 2.2 Bestimmung von Parametern mithilfe von Schätzverfahren

Die verbliebenen zu ermittelnden Parameter erfordern weit umfangreichere Messungen, wie zum Beispiel die Parameter für den Rollwiderstand und die Schräglauflübersetzung der Räder, die sich auch nicht direkt messen lassen. Für die Suche nach geeigneten Parametersätzen wird dabei ein Schätzverfahren benötigt, das anhand von anderen direkt messbaren Größen die von den gesuchten Parametern abhängig sind, die zu bestimmenden Parameter indirekt ermitteln kann. Dabei sind die wohl charakteristischsten Messgrößen des Fahrzeuges seine Position und Geschwindigkeit. Anhand der gegebenen Zustandsgleichungen und aus logischen Überlegungen erkennt man eine starke Abhängigkeit zu allen gesuchten Parametern. In diesem Abschnitt wird sich mit der Frage beschäftigt, wie die Messung dieser Zustände umzusetzen ist und wie danach aus den gewonnen Messdaten die noch fehlenden Parametern bestimmt werden.

### 2.2.1 Geschwindigkeits- und Positionsmessung

Die Geschwindigkeitsmessung wird hierfür intern vom Fahrzeug bereitgestellt. Sie wird dabei ähnlich wie an einem Fahrrad realisiert, indem Neodymmagnete auf alle Räder befestigt und durch Hall-Sensoren die Bewegungen der Magneten erfasst werden. Denn das Magnetfeld am Hall-Sensor ändert sich, sobald die Räder anfangen zu rotieren und die Änderung des Magnetfeldes kann schließlich direkt in eine Geschwindigkeit umgerechnet werden. Die Vorteile der Geschwindigkeitsmessung an jedem einzelnen Rad ist die höhere Auflösung, sowie eine bessere Messung der Geschwindigkeiten in Kurven. Da nämlich das Modellauto über ein Differentialgetriebe verfügt, bewegen sich die Räder in Kurvenfahrten unterschiedlich schnell, sodass die tatsächliche Geschwindigkeit des Fahrzeuges gemittelt werden muss.

Die Positionsbestimmung hingegen wird durch eine Kamera realisiert, die auf ein vorher definiertes Testfeld ausgelegt werden muss. Die technischen Anforderungen an das Tracking sind dabei ein ausreichend großes Testfeld, sowie eine geeignete Kamera. Die Wahl der Kamera hängt dabei von verschiedenen Anforderungen ab, die vor einer Installation des Systems klar definiert werden müssen. In dem Fall eines sich schnell bewegenden Fahrzeuges ist die Bilderanzahl pro Sekunde einer Kamera sehr hoch zu gewichten, um Bewegungsunschärfe und fehlende Bewegungen zwischen zwei Bilder zu minimieren, welche die Messungen verfälschen können. Desweiteren spielt die Auflösung für Positionsgenauigkeit und die Schnittstellen der Kamera für hohe Datenraten und Netzwerkfähigkeit eine große Rolle.

Nach dem Aufbau des Parcours und der Installation des Tracking-Systems, muss die Kamera kalibriert werden. Die Kalibrierung ist notwendig, um die Symmetrien die durch den „Fischaugeneffekt“ der Linse und der perspektivischen Verzerrung entstehen wiederherzustellen. Ein gängiges Verfahren zur Beseitigung der Linsenkrümmung und der perspektivischen Verzerrung ist die Kalibrierung mittels Schachbrettmuster. Die Beseitigung des Effektes der Linsenkrümmung wird mittels einer Vorwärtstransformation der Pixel des entkrümmten Bildes in das gekrümmte realisiert [16]. Der Algorithmus dahinter funktioniert folgendermaßen:

1. Generierung der Kamera-Matrix  $K$ , oder auch Matrix der intrinsischen Parameter mithilfe der „Camera Calibration Toolbox for Matlab“ [7]. Dabei charakterisieren die Einträge  $f_x$ ,  $f_y$  die Brennweite und  $c_x$ ,  $c_y$  bilden den Mittelpunkt der Linse auf dem Kamerabild.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

2. Normierung der Kamera-Matrix (Koordinate  $z'$  entfällt):

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = K^{-1} \cdot \begin{bmatrix} n_{h,1} & n_{h,1} & \dots & n_{h,1} & n_{h,1} & n_{h,2} & \dots & n_{h,2} & \dots & n_{h,i} \\ n_{b,1} & n_{b,2} & \dots & n_{b,j-1} & n_{b,j} & n_{b,1} & \dots & n_{b,j} & \dots & n_{b,j} \\ 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 & \dots & 1 \end{bmatrix}$$

mit

$n_h = 1, 2, 3, \dots$ , Höhe der Kameraauflösung

$n_b = 1, 2, 3, \dots$ , Breite der Kameraauflösung

3. Anwendung eines nichtlinearen Modells für die radiale Linsenkrümmung, um die verzerrten Koordinaten abzubilden. Dabei stammen die Parameter  $k_1, k_2$  (Koeffizienten der radialen Verzerrung) und  $p_1, p_2$  (Koeffizienten der tangentialen Verzerrung) ebenfalls aus Schritt 1 und wurden mit [7] berechnet.

$$x'' = x' \cdot (1 + k_1 \cdot r + k_2 \cdot r^2) + 2p_1 \cdot x'y' + p_2 \cdot (r + 2x'^2)$$

$$y'' = y' \cdot (1 + k_1 \cdot r + k_2 \cdot r^2) + 2p_2 \cdot x'y' + p_1 \cdot (r + 2y'^2)$$

mit

$$r = \sqrt{x'^2 + y'^2}$$

4. Nun lassen sich die Projektionen  $u$  und  $v$  erstellen, durch deren Anwendung mittels einer linearer Interpolation auf das Originalbild ein entzerrtes Bild erstellt werden.

$$u = f_x \cdot x'' + c_x$$

$$v = f_y \cdot y'' + c_y$$

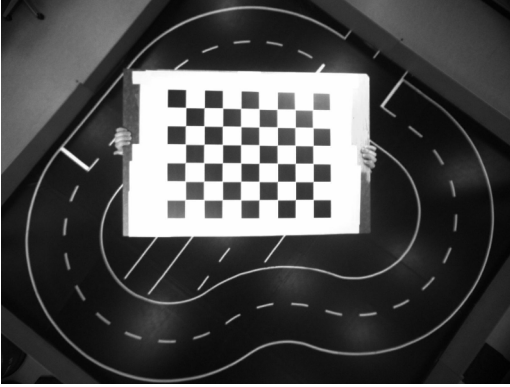


Abb. 3: Rohdaten der Kamera

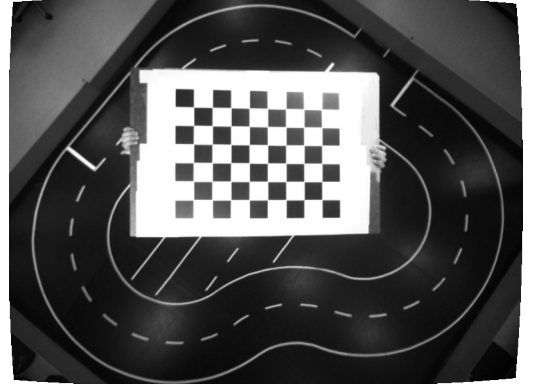


Abb. 4: Kamerabild nach der Entzerrung

Anschließend wird die perspektivische Ansicht in eine orthogonale Draufsicht projiziert. Dafür müssen die Ecken eines Rechtecks, bzw. die Ecken des Parcours (falls alle Winkel rechteckig) auf dem Bild markiert und eine homographische Matrix gebildet werden, damit ein sogenannter „Top View“ des Bildes erstellt werden kann. Diese Vorgehensweise ist bekannt unter dem Namen „Inverse Perspective Mapping“ ([11] und [9]):

1. Generierung der extrinsischen Matrix  $T_{ext}$  mithilfe von [7]. Die Einträge stehen hierbei für die Translation

in die entsprechende Richtung.

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

2. Ermittlung der Eckpunkte  $P_1, P_2, P_3$  und  $P_4$  in Pixelkoordinaten des Rechtecks/Parcours und die Matrix der Seitenlängenverhältnisse in Pixeln oder Meter.

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{mit} \quad P_n = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & d(P_1 \rightarrow P_2) & d(P_3 \rightarrow P_4) \\ 0 & d(P_2 \rightarrow P_3) & d(P_4 \rightarrow P_1) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3. Rotation um den Winkel  $\alpha$  und Translation der Bildpunkte relativ zur Kameraposition mit anschließender Normierung und der Projektion in die Zielkoordinaten.

$$R = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X_{rt} = R \cdot L + T * \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

$$X_{Target} = K \cdot X_{rt}^{-1} \cdot \begin{bmatrix} X_{rt|3,1} & \dots & X_{rt|3,4} \\ X_{rt|3,1} & \dots & X_{rt|3,4} \\ X_{rt|3,1} & \dots & X_{rt|3,4} \end{bmatrix}$$

4. Mit der gewonnenen Projektion  $X_{Target}$  lassen sich mit MATLAB unter der Verwendung von der Funktion *homography2d* [9], *maketform* und *imtransform* [10] das Bild im Anschluss perspektivisch entzerren.



Abb. 5: Perspektivische Ansicht des Parcours

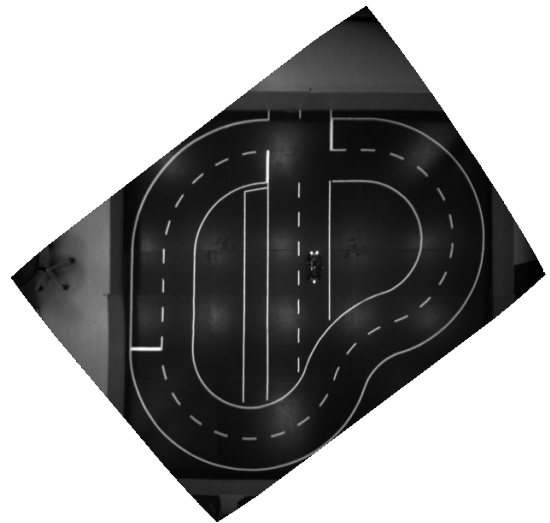


Abb. 6: Top-View Projektion des Parcours

(Workflow)

Nach der Kalibrierung der Kamera stellt sich die Fragestellung nach der Art der Positionserfassung. Die erste Lösung war eine Farberkennung, in der zwei Scheiben mit jeweils einer anderen Farbe auf den vorderen und hinteren Teil des Fahrzeuges befestigt wurden. Dadurch kann die Ausrichtung und der Mittelpunkt des Fahrzeuges bestimmt werden. Diese Methode war jedoch anfällig für Farbreflexionen auf der Fahrbahn, sodass die Farbwiedergabe, Belichtungszeit und Raumbeleuchtung mitunter vom Stand der Sonne abhängig war und ständig Neueinstellungen nach sich zog. Als zweite Lösung wurde im Team die Positionsbestimmung durch Infrarot-LEDs erarbeitet. Dabei wurde die Kamera um einen Filter erweitert der nur noch infrarotes Licht durchlässt. Am Fahrzeug wurden am Vorderteil zwei und hinten eine Infrarot-LED angebracht und deren Position im Bild bestimmt. Durch diese Vorgehensweise wird der Rechenaufwand für die Bildverarbeitung auf ein Drittel reduziert, da keine Farbinformationen mehr gespeichert werden. Denn vorher waren die Farbinformationen in einer  $B \times H \times 3$  Matrix kodiert, welche sich durch eine Schwarz-Weiß-Kodierung auf eine  $B \times H \times 1$  Matrix reduzierte. Zudem ist dieses Verfahren weniger störanfällig, da die einzigen infraroten Lichtquellen die LEDs sind. Somit entfällt auch der Aufwand für die Neueinstellungen der Kamera. Es hat sich sogar gezeigt, dass die Kamera dadurch eine geringere Belichtungszeit benötigt um die LEDs zu erkennen, was wiederum die Anzahl der Bilder pro Sekunde erhöhte.

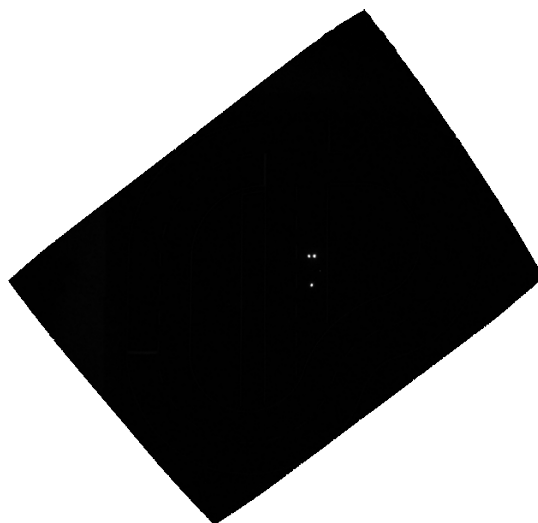


Abb. 7: Top-View Projektion des Parcours mit Infrarot-Filter



s)

### 2.2.2 Synchronisierung interner und externen Messdaten

Nach der Aufnahme der externen und internen Messwerte, müssen diese für die Kostenfunktion aufbereitet werden. Die Aufnahme der Messungen erfolgt dabei durch das Framework ROS (für Robot Operating System), welches alle Messungen über einen Server koordiniert und den Zeitpunkte einer jeder Messung protokolliert. Da die externen Positionsdaten der Kamera einen Takt von 50 Hz und die internen der Geschwindigkeitsmessungen einen Takt von 100 Hz haben, sind diese nicht koexistent. Um diese Daten miteinander zu synchronisieren wurde die Methode der linearen Approximation gewählt, um zu jedem Zeitpunkt Messdaten vorrätig zu haben. Dies ermöglicht einen unkomplizierten Umgang der Daten im Schätzverfahren und sorgt für eine bessere Vergleichsbasis zwischen realen und simulierten Zuständen.

### 2.2.3 Schätzverfahren

Unter der Verwendung der gemessenen Zustände  $y_{mess} = \{x_1, x_2, \phi, \dot{\phi}, v\}$  mit  $x_1, x_2$  für die Position,  $\phi, \dot{\phi}$  für die Winkelauslenkung und die Winkelgeschwindigkeit und  $v$  für die Geschwindigkeit des Fahrzeuges im Weltkoordinatensystem, lassen sich durch die Wahl einer geeigneten Kostenfunktion und Optimierungsverfahrens die gesuchten Parameter identifizieren. Als Kostenfunktion wurde die Differenz aus gemessenen und simulierten Zustände mit entsprechenden Gewichtungen gewählt und diese zusätzlich quadriert, um die Kostenfunktion annähernd parabolisch darzustellen. Die simulierten Zustände werden anhand dem Modell aus [15] und mit einem ODE Solver, basierend auf einem expliziten Runge-Kutta-Verfahren (*ode45*) in Matlab ermittelt und in dem Vektor  $y_{sim} = \{x_{sim|1}, x_{sim|2}, \phi_{sim}, \dot{\phi}_{sim}, v_{sim}\}$  zusammengefasst. Mit den synchronisierten Daten aus den Messreihen lassen sich einzelne Zeitpunkte extrahieren und diese in der Simulation explizit angeben, sodass zu den ausgewählten Zeitpunkten in der Simulation auch reale Messwerte existieren. Die Gewichtungen  $q_k$  für jeden Zustand wurden dabei so gewählt, dass die Güte eines simulierten Zustandes zu gleichen Teilen in die folgende Kostenfunktion mit einfließt:

$$J = \sum_{t=0}^{t_{end}} \sum_{k=1}^5 q_k \cdot (y_{mess,k}(t) - y_{sim,k}(t))^2$$

Zur Optimierung wurden zwei Ansätze verfolgt um ein Optimum gewährleisten zu können. Zum einen wurde die Matlab interne Funktion *fmincon* und zum anderen ein selbst entwickelter Partikel-Schwarm-Algorithmus nach [4] verwendet. Dabei galt es den zu schätzende Parametervektor  $p = \{\gamma, f_R, C_\alpha, T_V\}$ , bestehend aus dem Verhältnis der Kraftübertragung zwischen Vorder- und Hinterräder  $\gamma$ , dem Rollwiderstand  $f_R$ , der Schräglauflübersetzung  $C_\alpha$  und der zeitlichen Verzögerung des Lenkeinschlages  $T_V$  zu bestimmen. Um die Dimension des Vektors zu reduzieren, wurde schon in [15] angedacht den Rollwiderstand allein durch eine Geradeausfahrt zu schätzen, da die anderen Parameter lediglich Einfluss auf die Zustände während Kurvenfahrten besitzen. Diese Trennung der Parameterschätzung ermöglichte es, ein optimales  $f_R$  zu finden, sodass in den Kurvenfahrten alle weiteren Parameter bestimmt werden konnten und beide Optimierungsverfahren stets in das gleiche Optimum liefen. In 8 wurde als Beispiel ein solcher Optimierungsaufwurf durch den Partikel-Schwarm-Algorithmus dargestellt. Zu sehen der Verlauf der Kostenfunktion über jede neue Generation von Partikeln bei einer Geradeausfahrt für den Zustand  $f_R$ . Dabei wird sehr gut die Konvergenz vom Algorithmus ersichtlich. Die qualitativen Ergebnisse für die modellbasierte Berechnung aller Zustände befinden sich im Anhang.

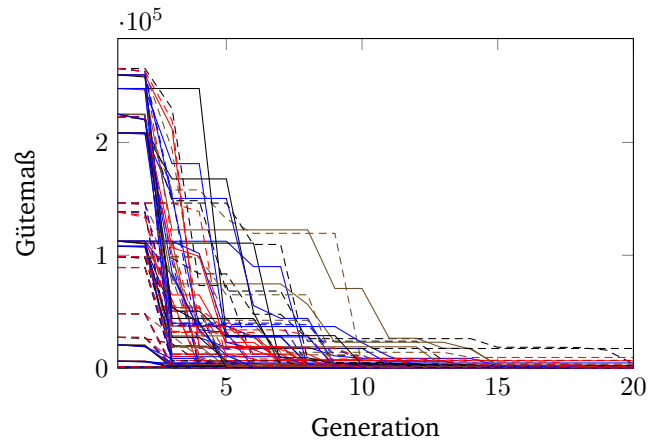


Abb. 8: Verlauf der Kostenfunktion für alle Partikel

## 2.2.4 Ergebnisse

In der Anwendung der Schätzverfahren revidierten sich einige der Annahmen aus [15] und es mussten teilweise neue getroffen werden, um bessere Ergebnisse zu erreichen. Zum Beispiel wurde die Annahme getroffen, dass die Schräglauübersetzung  $C_\alpha$  von Vorder- und Hinterräder gleich ist. Jedoch ergaben sich bei einer Trennung der Schräglauübersetzung von Vorder- und Hinterräder eindeutig bessere Ergebnisse. Zudem musste ein weiterer Parameter  $\eta_M$  eingeführt werden, um den Wirkungsgrad des Drehmomentes zu beschreiben, welches vom Motor erzeugt auf die Räder geleitet wird. Somit ergaben sich folgende Parameter, die das Modell mit einer ausreichenden Genauigkeit beschreiben:

Physikalische Größe	Exakter Wert	Einheit
Kraftverteilung Vorder- zu Hinterrad	$\gamma = 0,5$	[-]
Reibungskoeffizient	$f_R = 1,857$	[-]
Schräglauübersetzung hinten	$C_{h,\alpha} = 0,93$	[-]
Schräglauübersetzung vorn	$C_{v,\alpha} = 1,1$	[-]
Wirkungsgrad Drehmoment Motor $\rightarrow$ Räder	$\eta_M = 0,55$	[-]
Zeitl. Verzögerung von $u_1$	$T_V = 0,1$	[s]

Tabelle 2: Ergebnisse der geschätzten Parameter.

(Bild Ergbnis Realität zu Simulation mit geschätzten Parametern, Aussage „ausreichend“ definieren - sprich das Modell kann dem realen System für 2 Meter exakt auf ... Metern folgen, dass reicht aus um die MPC für einen Zeithorizont von ... Sekunden bei der Geschwindigkeit von ... M/s zu auszulegen. Da wir auch nicht mehr von der Wahrnehmung erhalten, ergibt sich eine korrekte Vorhersage)

## Literatur

- [1] A. PROYAS (REGIE): I, Robot. Film (2004). <http://www.imdb.com/title/tt0343818/>
- [2] CAROLOCUP: Carolo-Cup Regelwerk 2015. <https://wiki.ifr.ing.tu-bs.de/carolocup/system/files/Hauptwettbewerb2015.pdf>. – Homepage Carolo-Cup: [www.carolo-cup.de](http://www.carolo-cup.de)
- [3] CAROLOCUP: Carolocup 2014. <https://wiki.ifr.ing.tu-bs.de/carolocup/wettbewerb/2014/bilder/2Trainingstag>
- [4] D. RINI ; S. SHAMSUDDIN ; A. YUHANIZ: Particle Swarm Optimization: Technique, System and Challenges. In: International Journal of Computer Applications (0975-8887) 14, Nr. 1. <http://ijcaonline.net/volume14/number1/pxc3872331.pdf>
- [5] H. WEISE: Das intelligent geführte Auto. In: Die Zeit <http://www.zeit.de/1992/35/das-intelligent-gefuehrte-auto>
- [6] J. MULLER: With Driverless Cars, Once Again It Is California Leading The Way. In: Forbes Magazin <http://www.forbes.com/sites/joannmuller/2012/09/26/with-driverless-cars-once-again-it-is-california-leading-the-way/>
- [7] J.-Y. BOUGUET: Camera Calibration Toolbox for Matlab. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html)
- [8] M. BRAMBILLA (REGIE): Demolition Man. Film (1993). <http://www.imdb.com/title/tt0106697/>
- [9] P. KOVESI: MATLAB and Octave Functions for Computer Vision and Image Processing. <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>

- 
- [10] S. EDDINS: Spatial transformations: Defining and applying custom transforms / MathWorks. <http://blogs.mathworks.com/steve/2006/08/04/spatial-transformations-defining-and-applying-custom-transforms/>. – Forschungsbericht
- [11] S. TUOHY ; D.O'CUALAIN ; E. JONES ; M.GLAVIN: Distance Determination for an Automobile Environment using Inverse Perspective Mapping in OpenCV. In: ISSC <http://www.shanetuohy.com/fyp/Images/issc.pdf>
- [12] SCIENTIF38: Winkelmesser. [http://commons.wikimedia.org/wiki/File:Protractor\\_Rapporteur\\_Degree\\_V1.jpg](http://commons.wikimedia.org/wiki/File:Protractor_Rapporteur_Degree_V1.jpg)
- [13] T. ROSE ; M. TRESELER: Entwicklung einer Karosserie für das oTToCar.
- [14] TAMIYA INC.: Tamiya TT-01 Type-E (TT-01E) Chassis. <http://www.rcscrapyard.net/de/tamiya-tt-01-type-e.htm>
- [15] V. WIEDMEIER ; A. HIMMEL: oTToCAR, Interdisziplinäres Teamprojekt.
- [16] Z. ZHANG: A Flexible New Technique for Camera Calibration / Microsoft Research, Microsoft Corporation. <http://research.microsoft.com/en-us/um/people/zhang/Papers/TR98-71.pdf>. – Forschungsbericht