Interdisziplinäres Teamprojekt

Parameterschätzung und Entwurf einer Modellprädiktiven Regelung eines autonomen Modellfahrzeuges

von

Hannes Heinemann
Matrikelnummer: 185102

André Pieper
Matrikelnummer: 184960

2. April 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 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 Zusamenarbeit erfolgt dabei durch die Fakultäten für Informatik, für Elektrotechnik und Informationstechnik und für Maschinenbau. Die Vorbereitung dieser Arbeit ist maßgeblich an das Forschungsprojekt [17] 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

1 Einleitung

Im Zuge der „digitalen Revolution“[1] die durch die rasante Entwicklung der Halbleitertechnologien (beschrieben durch das mooresche Gesetz[2]) 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“[10] sugeriert wird, sondern der Prozess ist schon so weit fortgeschritten, dass der autonome Straßenverkehr in naher Zukunft Realität wird. Dafür spricht zum einen, dass schon die ersten autonomen Fahrzeuge des „EUREKA-PROMETHEUS-Projekts“ [6] 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[8] und in Europa zumindest dafür neue Gesetze entworfen werden[5]. 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 Reglungskonzept 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 vorrangegangenen Arbeit [17] 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]

[1] 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

[2] 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.

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 [17, Seite 12] entnommen, welches dort 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 Direkte Messmethoden

Die direkte Bestimmung von Parametern erfolgt weitgehend durch die Messung der selbigen, oder deren einfache Berechnung durch weitere messbare Größen. Im Falle der Eingangsgröße u_1, dem Einschlagwinkel 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 nur 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.

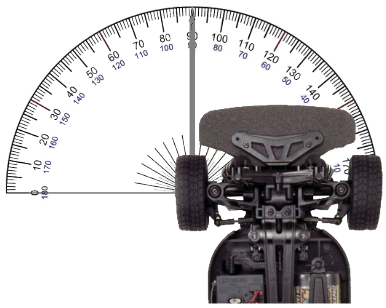

Abb. 2: Messung der Relation PWM-Stufe des Servos zum Einschlagwinkel der Räder (aus [16] und [14])

Als Ergebnis dieses Experimentes kam heraus, dass die Erhöhung um eine PWM-Stufe des Servomotors konstant eine Drehung der Räder um 0.0028 rad entspricht. Andere Parameter konnten durch Recherche ermittelt werden, da sie sehr gebräuchliche Kenngrößen darstellen, wie z.B. die Dichte der Luft. Desweiteren wurde das Trägheitsmoment durch die Arbeit [15] mithilfe eines CAD-Modells vom Fahrzeug berechnet. In der Tabelle 1 werden alle diese Größen diesbezüglich aufgeführt.

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

Tabelle 1: Direkt bestimmbare Parameter

2.2 Indirekte Messmethoden

Die verbliebenen zu ermittelnden Parameter erfordern weit umfangreichere Messungen, wie zum Beispiel die Parameter für den Rollwiderstand und die Schräglaufübersetzung 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 in der nächsten Passage wird erklärt, wie danach aus den gewonnen Messdaten die noch fehlenden Parametern bestimmt werden.

2.2.1 Geschwindigkeitsmessung

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 Magnete 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 sind 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.

2.2.2 Entwicklung eines Tracking-Systems zur Positionsmessung

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 Größe des Parcours war jedoch durch die räumlichen Gegebenheiten stark begrenzt, sodass der Aufbau lediglich eine Fläche von 5$\times$5 Metern misst. 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 [18]. Der Algorithmus dahinter funktioniert folgendermaßen:

1. Generierung der Kamera-Matrix K, oder auch Matrix der intrinsischen Parameter mithilfe der „Camera Calibration Toolbox for Matlab“ [9]. 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):

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

mit

$$n_h = 1, 2, 3, ..., \text{Höhe der Kameraauflösung}$$
$$n_b = 1, 2, 3, ..., \text{Breite der Kameraauflösung}$$

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 [9] berechnet.

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

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.

$$u = f_x \cdot x'' + c_x$$
$$v = f_y \cdot y'' + c_y$$

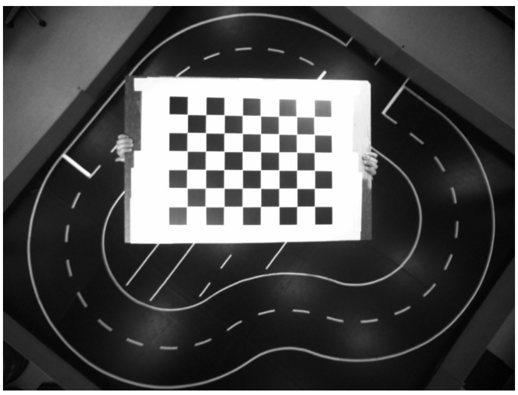

Abb. 3: Rohdaten der Kamera

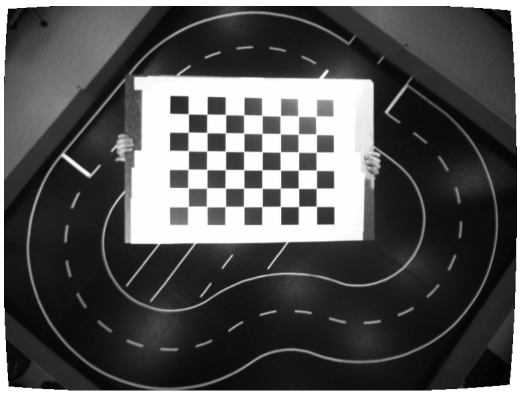

Abb. 4: Kamerabild nach der Entzerrung

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“ ([13] und [11]):

1. Generierung der extrinsischen Matrix T_{ext} mithilfe von [9]. 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.

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

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

3. Rotation um den Winkel α 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* [11], *maketform* und *imtransform* [12] das Bild im Anschluss perspektivisch entzerren.

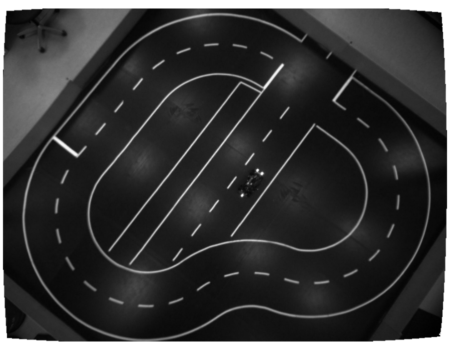

Abb. 5: Perspektivische Ansicht des Parcours

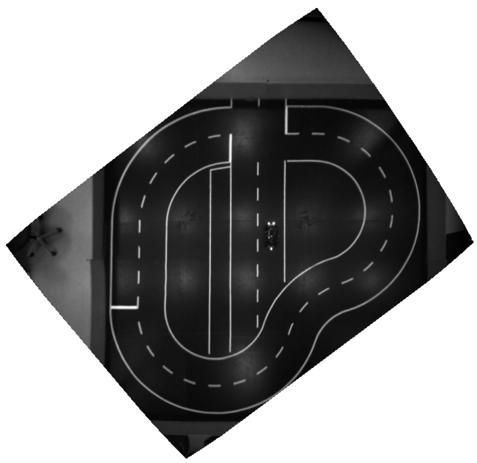

Abb. 6: Top-View Projektion des Parcours

Nach der Kalibrierung der Kamera stellt sich die Frage 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 können 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.

2.2.3 Synchronisierung interner und externen Messdaten

Nach der Aufnahme der externen und internen Messwerte, müssen diese für die Parameterschätzung 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 Sensoren der Geschwindigkeitsmessungen mit einem Takt von 100 Hz laufen, 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 Zustände.

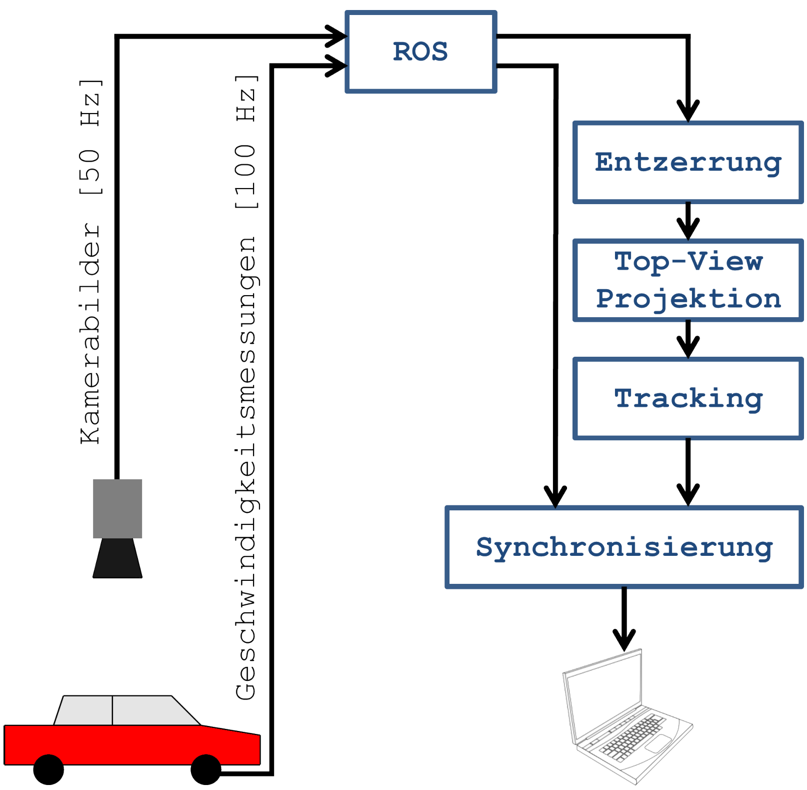

Abb. 7: Workflow des Tracking-Systems

2.3 Schätzverfahren

Unter der Verwendung der gemessenen Zustände $y_{mess} = \left\{x_1, x_2, \phi, \dot{\phi}, v\right\}$ mit x_1, x_2 für die Position, $\phi, \dot{\phi}$ für die Winkelauslenkung und die Winkelgeschwindigkeit, sowie 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 [17] und mit einem ODE Solver, basierend auf einem expliziten Runge-Kutta-Verfahren (*ode45*) in Matlab ermittelt und in dem Vektor $y_{sim} = \left\{x_{sim|1}, x_{sim|2}, \phi_{sim}, \dot{\phi}_{sim}, v_{sim}\right\}$ 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:

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

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 γ, dem Rollwiderstand f_R, der Schräglaufübersetzung C_α und der zeitlichen Verzögerung des Lenkeinschlages T_V zu bestimmen. Um die Dimension des Vektors zu reduzieren, wurde schon in [17] 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 Abbildung 8 wurde als Beispiel ein solcher Optimierungsaufruf durch den Partikel-Schwarm-Algorithmus dargestellt. Zu sehen ist der Verlauf der Kostenfunktion über jede neue Generation von Partikeln bei einer Geradeausfahrt für den Parameter f_R. Dabei wird sehr gut die Konvergenz vom Algorithmus ersichtlich.

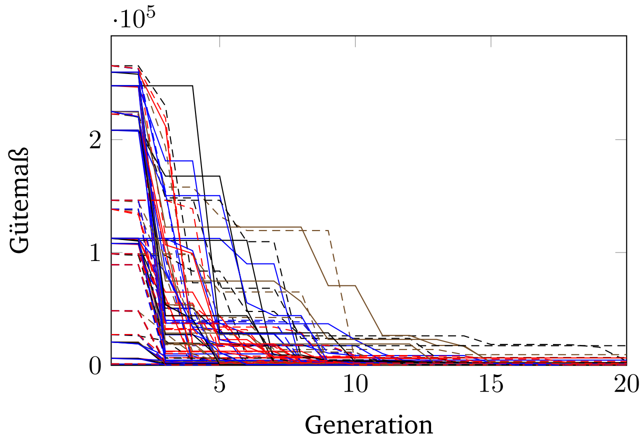

Abb. 8: Verlauf der Kostenfunktion für alle Partikel

2.4 Ergebnisse

In der Anwendung der Schätzverfahren revidierten sich einige der Annahmen aus [17] und es mussten teilweise neue getroffen werden, um bessere Ergebnisse zu erreichen. Zum Beispiel wurde die Annahme getroffen, dass die Schräglaufübersetzung C_α von Vorder- und Hinterräder gleich ist. Jedoch ergaben sich bei einer Trennung der Schräglaufübersetzung von Vorder- und Hinterräder eindeutig bessere Ergebnisse. Zudem musste der Eingang des Motordrehmomentes um ein weiterer Parameter η_M erweitert werden, um den Wirkungsgrad des Drehmomentes zu beschreiben, welches vom Motor erzeugt und 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	$f_R = 1,857$	[-]
Schräglaufübersetzung hinten	$C_{h,\alpha} = 0,93$	[-]
Schräglaufübersetzung 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

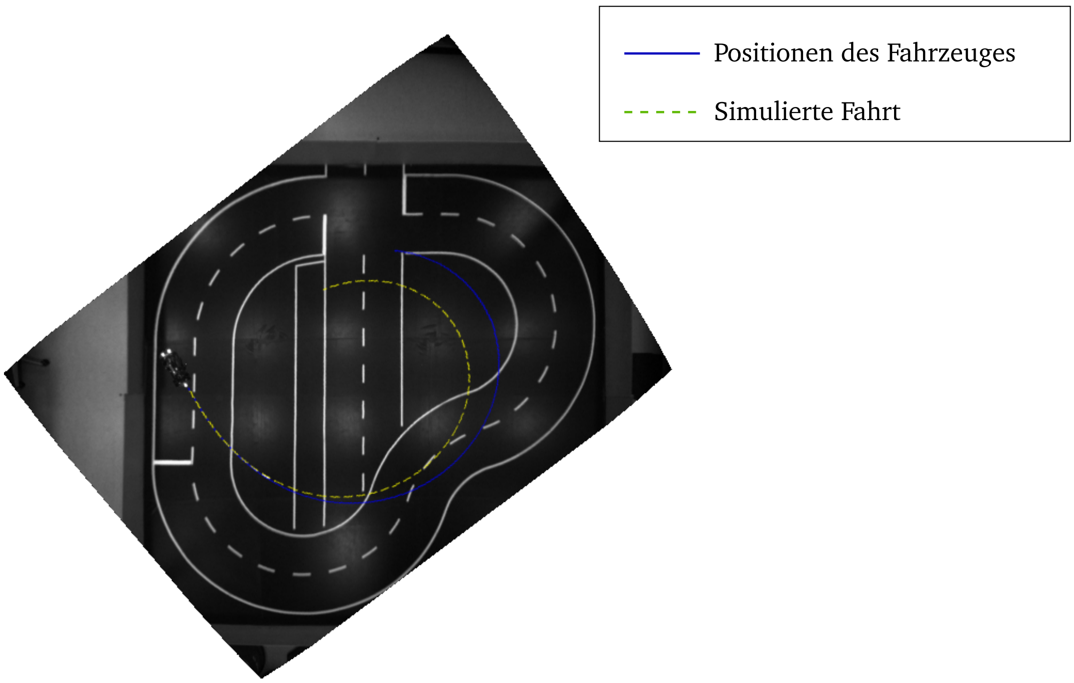

Abb. 9: Reale Messwerte und simulierte Zustände im direkten Vergleich

In Abbildung 9 wurden die Simulationsdaten mit den real gemessen Positionsdaten direkt gegenübergestellt. Die Fahrt einer Linkskurve wurde dabei mit einer Endgeschwindigkeit von $2\frac{\text{m}}{\text{s}}$ und einem konstanten Lenkeinschlag von ungefähr $15°$ durchgeführt. Die Ergebnisse scheinen auf den ersten Blick nicht zufriedenstellend zu sein, da nach ungefähr drei Metern die Abweichung beider Kurven größer als die Fahrspurbreite wird. Die Gründe sind zum einen, dass das Modell die Realität nur unzureichend beschreibt und zum anderen das bei einer numerischen Berechnung eines nichtlinearen Differentialgleichungssystems stets Berechnungsfehler gemacht werden. Jedoch unter Berücksichtigung der Tatsache, dass im autonomen Fahrmodus die Kamera maximal zwei Meter voraus blicken kann, relativiert sich die Güte des Modells zu einer sehr guten Vorhersage für einen begrenzten Horizont. Da in jedem Schritt das Fahrzeug wieder auf die Trajektorie platziert wird, summiert sich somit der gemachte Fehler nur bis zum nächsten Messzeitpunkt und wird danach wieder auf Null zurückgesetzt.

In einer zukünftigen Version vom „oTToCar“ ist die Spurerkennung nicht mehr allein durch die Kamera-Wahrnehmung gegeben. Zusätzlich soll sich das Auto in einer Weltkarte zurechtfinden und die Pfade entlang der Fahrbahn weit im Voraus berechnet werden. Für dieses Verfahren reicht das Modell und die Parameter jedoch nicht mehr aus und müsste weiter verbessert werden.

3 Fahrspurregelung mittels modellprädiktiver Regelung

In allen dynamischen Disziplinen des Carolo-Cups besteht die wichtige Teilaufgabe möglichst fehlerfrei einer Fahrbahn zu folgen. Um diese Aufgabe zu bewältigen, wurde ein modellprädiktiver Regelansatz gewählt.

Bei der modellprädiktiven Regelung (MPC) handelt es sich um eine Form der optimalen Regelung, bei der wiederholt eine Berechnung der optimalen Steuerung für ein System ausgehend von dessen aktuellem Zustand stattfindet. In vielen Bereichen finden MPCs immer häufiger Anwendung, da sie eine direkte Berücksichtigung von Beschränkungen erlauben und eine Form des strukturierten Reglerentwurfs ausgehend von der modellierten Systemdynamik darstellen. Dabei kann durch die geeignete Wahl der Kostenfunktion und deren Wichtungsparametern die Güte des Reglers gezielt beeinflusst werden. Allerdings ergeben sich auch Schwierigkeiten bei der Verwendung von MPCs. Zum einen ist die Konvergenz der Optimierung gegen einen optimalen Wert für die Optimierungsvariablen und die Stabilität des geschlossenen Kreises insbesondere bei nichtlinearen Systemmodellen oft nur schwierig nachweisbar und zum anderen stellt das wiederholte Lösen des meist hochdimensionalen Optimierungsproblems während der Laufzeit in genügend schneller Geschwindigkeit eine große Herausforderung dar.

3.1 Problemformulierung für die modellprädiktive Regelung

Im realen Anwendungsfall des oTToCAR-Projekts eignet sich eine Systemdarstellung in zeitdiskreter Form ([7]), bei der die Lösung des Optimierungsproblems weniger komplex ist und die ebenfalls zeitdiskreten Messwerte vom realen System weniger kompliziert integriert werden können. Demnach sind die diskretisierten Systemgleichungen wie folgt gegeben:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{f}\left(\boldsymbol{x}(k), \boldsymbol{u}(k)\right) \\ \boldsymbol{y}(k) &= \boldsymbol{g}\left(\boldsymbol{x}(k)\right) \end{aligned}$$

mit den nichtlinearen Funktionen $\boldsymbol{f}(\cdot)$ und $\boldsymbol{g}(\cdot)$, wobei

$$\begin{aligned} \boldsymbol{x}(k) &\in \mathcal{X} \subset \mathbb{R}^n \\ \boldsymbol{u}(k) &\in \mathcal{U} \subset \mathbb{R}^m \\ \boldsymbol{y}(k) &\in \mathcal{Y} \subset \mathbb{R}^r \end{aligned}$$

Ausgehend vom aktuellen Zustand $\boldsymbol{x}(k)$ des zu regelnden Systems, der wenn nicht messbar geschätzt werden muss, wird anhand des Systemmodells das zukünftige Systemverhalten

$$\boldsymbol{x_p} = \{\boldsymbol{x}(k+1), \ldots, \boldsymbol{x}(k+n_p)\}$$

bis zum Prädiktionshorizont n_p unter der Optimierung einer Sequenz von Eingängen

$$\boldsymbol{u} = \{\boldsymbol{u}(k), \ldots, \boldsymbol{u}(k+n_c-1)\}$$

bis zum Stellhorizont n_c vorhergesagt. Aus der gefundenen optimalen Eingangssequenz $\boldsymbol{u}^*$ wird der erste Eintrag $\boldsymbol{u}^*(k)$ auf das zu regelnde System angewandt. Im nächsten Zeitschritt kann der neue Zustand gemessen bzw. geschätzt werden und die Optimierung beginnt von neuem. Ziel dabei ist es einer Referenztrajektorie $\boldsymbol{x_r}$ zu folgen.

Für das an jedem Zeitschritt k zu lösende Minimierungsproblem wurde die benötigte Kostenfunktion J in quadratische Form mit $\boldsymbol{x_p}$ und $\boldsymbol{u}$ als Optimierungsvariablen aufgestellt:

$$\min_{\boldsymbol{x_p},\boldsymbol{u}} J := \sum_{i=k+1}^{k+n_p} \left[\boldsymbol{x}_p(i) - \boldsymbol{x}_r(i)\right]^T \boldsymbol{Q}_i \left[\boldsymbol{x}_p(i) - \boldsymbol{x}_r(i)\right] + \sum_{j=k}^{k+n_c-1} \boldsymbol{u}^T(j)\boldsymbol{R}_j\boldsymbol{u}(j)$$
$$s.t.\ \boldsymbol{x_p}(i+1) = \boldsymbol{f}\left(\boldsymbol{x_p}(i), \boldsymbol{u}(i)\right), \quad i = k, ..., k+n_c-1$$
$$\boldsymbol{x_p}(i+1) = \boldsymbol{f}\left(\boldsymbol{x_p}(i), \boldsymbol{u}(k+n_c-1)\right), \quad i = k+n_c, ..., k+n_p-1$$

Mit den Vektoren

$$\boldsymbol{x}_p(k) = \left[\boldsymbol{x}_p(k+1 \mid k), \ldots, \boldsymbol{x}_p(k+n_p \mid k)\right]^T$$
$$\boldsymbol{x}_r(k) = \left[\boldsymbol{x}_r(k+1), \ldots, \boldsymbol{x}_r(k+n_p)\right]^T$$
$$\boldsymbol{u}(k) = \left[\boldsymbol{u}(k), \ldots, \boldsymbol{u}(k+n_c-1)\right]^T$$

und den dazugehörigen positiv definiten Wichtungsmatrizen $\boldsymbol{Q}_i \in \mathbb{R}^{n\times n}$ $(i = 1, ..., n_p)$ und $\boldsymbol{R}_j \in \mathbb{R}^{m\times m}$ $(j = 0, ..., n_c - 1)$. Weiterhin lässt sich das Optimierungsproblem um einfache Beschränkungen der Eingänge

$$\boldsymbol{u}_{min} \leq \boldsymbol{u}(i) \leq \boldsymbol{u}_{max}, \quad i = k, ..., k+n_c-1$$

und Zustandsbeschränkungen der Form

$$\boldsymbol{A}\boldsymbol{x}_p(i) \leq \boldsymbol{b} \quad i = k+1, ..., k+n_p$$

erweitern.

3.2 Implementierung der Fahrspurregelung für das oTToCAR

Um die Theorie in die Praxis umzusetzen, muss der Algorithmus der MPC um die Erstellung einer Referenztrajektorie erweitert, sowie Vereinfachungen am Algorithmus vorgenommen werden. Denn der Schwachpunkt einer jeder modellprädiktiven Regelung ist die Laufzeit des Programms bei einer Online-Berechnung der nächsten Stellgröße. Somit muss ein Kompromiss zwischen Güte der Stellgröße und der Zeit zur Berechnung für diese gefunden werden.

3.2.1 Dimension der Optimierungsvariablen

Als Systemeingänge für das Fahrzeugmodell sind der Lenkeinschlag der Räder $u_1 = \delta$ und das Motordrehmoment $u_2 = M_A$ vorhanden, demnach ergibt sich der Stellgrößenvektor zum Zeitpunkt k zu $\boldsymbol{u}(k) = [u_1(k), u_2(k)]^T$. Da sich die prädizierten Zustände $\boldsymbol{x}_p$ aus den Gleichungsnebenbedingungen berechnen lassen, ergibt sich die Dimension der Optimierungsvariablen also zu $2 \cdot n_c$. Je größer man n_c wählt, desto besser lässt sich z.B. die Geschwindigkeit in Abhängigkeit zur Entfernung und Krümmung einer bevorstehenden Kurve begrenzen oder Schwierigkeiten beim Durchfahren von S-Kurven bewältigen. Allerdings nimmt dadurch auch die Dimension der Optimierungsvariablen zu, was dazu führt, dass die Grenzen der verfügbaren Rechenzeit zum Lösen des Optimierungsproblems während der Laufzeit schnell erreicht werden.

3.2.2 Referenztrajektorie

Mit einer auf dem Fahrzeug befestigten Kamera lässt sich die zu verfolgende Fahrbahn wahrnehmen. Diese Wahrnehmung liefert mit einer Rate von bis zu 50Hz eine Polylinie der erkannten Fahrbahnmarkierung, projiziert auf die rechte Fahrspur. Diese Polylinie kann je nach erkannter Fahrbahn aus einer unterschiedlichen Anzahl von Punkten mit variablem Abstand zu einander bestehen.

Um diese Polylinien als Referenztrajektorie für die MPC nutzen zu können, werden zu kurze Polylinien zunächst soweit extrapoliert, dass ein Vergleich zwischen Prädiktion zum Zeitschritt $k + np$ und Referenz in jedem Fall möglich ist. Anschließend werden die Koordinaten der Punkte auf der Polylinie als Funktion in Abhängigkeit der summierten Wegstrecke zwischen den Punkten (vom Auto ausgehend) hinterlegt, um diese

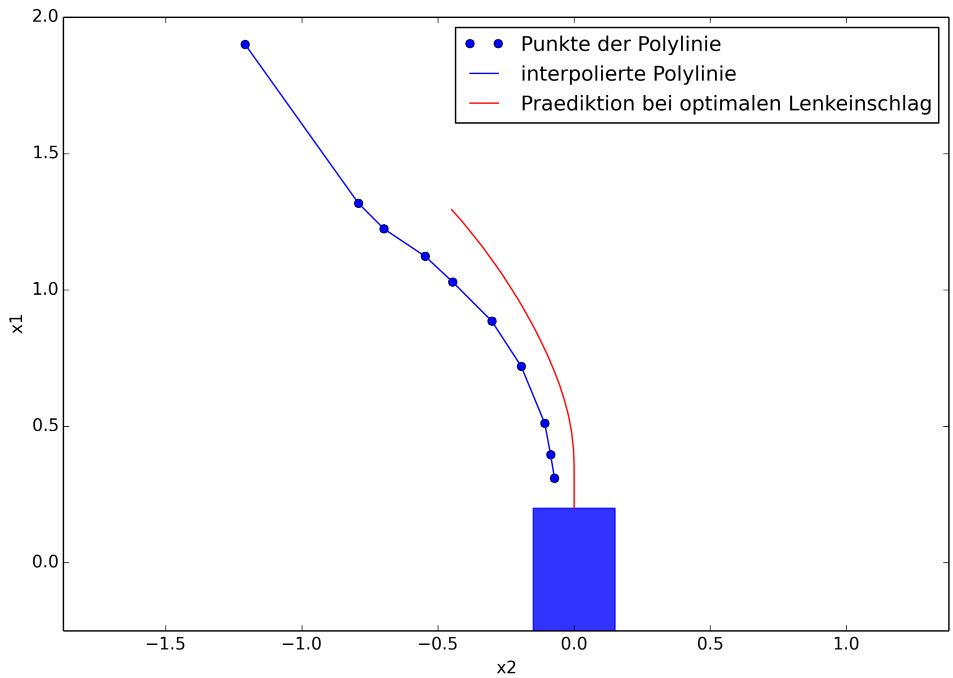

Abb. 10: Veranschaulichung der Referenztrajektorie und des prädizierten Verhaltens aufgetragen im Weltkoordinatensystem x_1 über x_2 in Meter

als Referenztrajektorie zu den dazugehörigen Werten aus der Prädiktion mit der zurückgelegten Wegstrecke $s_i = iv\Delta t$ interpolieren zu können. Dabei ist v die Geschwindigkeit der Fahrzeuges. Die Terme $[\boldsymbol{x}_p(i) - \boldsymbol{x}_r(i)]$ in der Kostenfunktion werden demnach durch $[\boldsymbol{x}_p(i) - \boldsymbol{x}_r(s_i)]$ ersetzt. In Abb. 10 ist ein Beispiel einer Referenztrajektorie dargestellt.

Das Modell liefert mit x_1 und x_2 die globale Position des Autos und den dazugehörigen Gierwinkel $x_3 = \psi$. Da bisher noch keine globale Karte implementiert ist, wird jedes Mal, wenn eine neue Polylinie übermittelt wird die jeweils lokal gültige Weltkarte auf das Fahrzeugkoordinatensystem zurückgesetzt. Es ist kurzzeitig möglich, bei Ausbleiben einer verlässlichen neuen Polylinie, sich weiter an der letzten Referenztrajektorie zu orientieren, in dem man den Zustand bezogen auf das letzte lokal gültige Weltkoordinatensystem mit Hilfe des Modells schätzt. Dies ist allerdings nur für sehr kurze Zeiten (einzelne ausbleibende Polylinien) verlässlich.

3.2.3 Reduzierung der Komplexität des Optimierungsproblems

Aufgrund zwischenzeitlich erschöpfter Rechenkapazitäten auf dem Fahrzeug während der Entwicklungsphase, wurde das zu lösende Optimierungsproblem stark vereinfacht. Dabei wurde darauf verzichtet die Motorsteuergröße mitzubetrachten und der Stellhorizont für den Lenkeinschlag auf $n_c = 1$ begrenzt. Aufgrund überraschend guter Ergebnisse wurde an dieser Vereinfachung auch im Carolo-Cup festgehalten.

Im nun eindimensionalen Optimierungsproblem lässt sich der Verlauf der Kostenfunktion in Abhängigkeit vom Lenkeinschlag visuell darstellen. Schnell wird dabei klar, dass die Kostenfunktion im Bereich der real möglichen Lenkeinschläge eine Parabelform aufweist. Der Scheitelpunkt der Parabel und damit eine sehr gute Approximation des optimalen Lenkeinschlags, lässt sich mit nur 3 Funktionsaufrufen bestimmten. Dieses Vorgehen führt zu immensen Rechenzeiteinsparungen.

Da die Stellgröße für das Motordrehmoment nicht optimiert wird, muss die Anpassung der Geschwindigkeit auf andere Weise vorgenommen werden, wenn das Fahrzeug auf den langen Geraden möglichst schnell

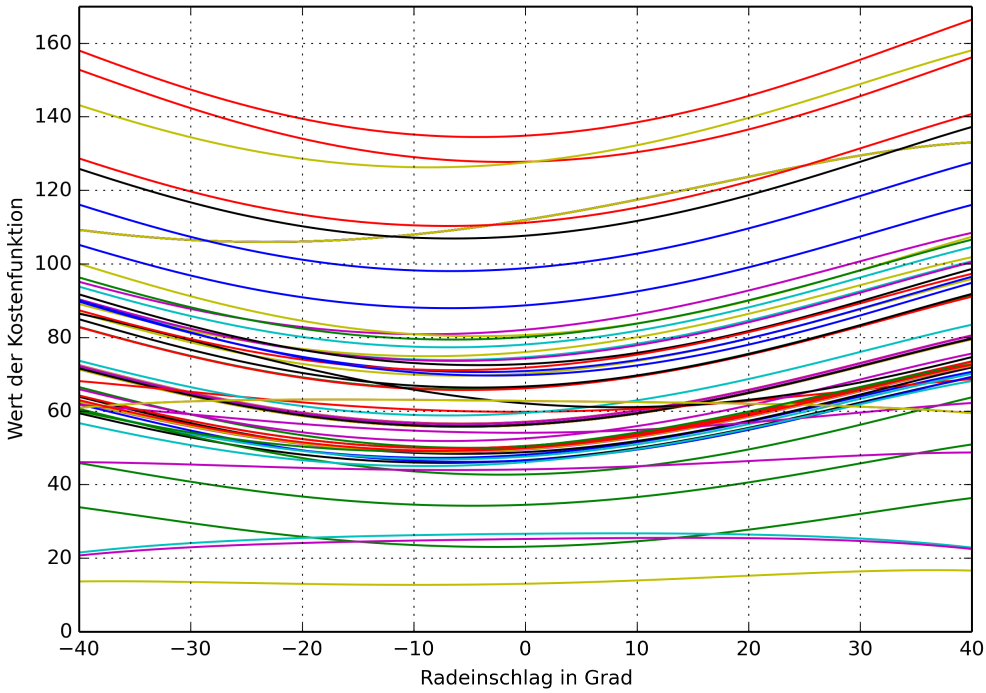

Abb. 11: Kostenfunktion in Abhängigkeit vom Radeinschlag in verschiedenen Fahrzeugpositionen

fahren soll. Die Vorgabe der Geschwindigkeit erfolgt dabei aus einer einfachen Abhängigkeit von der mittleren prädizierten Gierrate $\dot{\psi}$ und wird von einem unterlagertem Geschwindigkeitsregler sichergestellt.

3.2.4 Stabilität

Um die Garantie zu haben, dass der Algorithmus immer gegen ein Optimum konvergiert, wurde die im implementierten Fall skalare Kostenfunktionen während Fahrt auf dem Parcours in möglichst vielen denkbaren Positionen aufgenommen und überprüft, ob sich Fälle ergeben in denen die Kostenfunktion im relevanten Bereich der möglichen Lenkeinschläge nicht konvex ist. Abb. 11 zeigt das Ergebnis dieser Untersuchung. Es ist zu erkennen, dass die Kostenfunktion nur in wenigen Ausnahmefällen einer nach unten geöffneten Parabel ähnelt. Der hierbei berechnete Lenkeinschlag wird nicht auf das System gegeben, stattdessen wird an der vorherigen berechneten Referenztrajektorie weiter gefahren und auf die nächste Polylinie von der Wahrnehmung gewartet. Dank der relativ hohen Wiederholrate des Reglers stellen diese vereinzelten Ausfälle kein Problem dar.

3.3 Spurwechsel

Im Rundkurs mit Hindernissen beim Carolo-Cup ist es erforderlich Hindernissen auf der Fahrbahn auszuweichen. Dazu wird die erkannte Polylinie auf die linke Fahrspur verschoben, sobald ein Hindernis detektiert wird (wie in Abb. 12 dargestellt). Die MPC soll die optimale Trajektorie zu den verschobenen Punkten finden, sodass keine gesonderte Fallbetrachtung mit alternativen Reglereinstellungen vorgenommen werden muss. Ein Problem stellt dabei allerdings die Wahrnehmung dar, da bei zu starkem Einlenken die Fahrbahn aus dem Sichtfeld der Kamera verschwindet. Alternativ eine langsamere Spurwechseltrajektorie vorzugeben bei der die Fahrbahn weiterhin wahrgenommen wird, ist nicht denkbar, da Hindernisse nach einer Kurve oft erst spät erkannt werden und deshalb Spurwechsel auf möglichst kürzester Distanz nötig sind. Da das Verfolgen der zuletzt gesehenen Polylinie wegen Modellungenauigkeiten und begrenzter Länge der Polylinie nur kurzzeitig ausreichend gut funktioniert, schlägt die implementierte Strategie immer noch häufig fehl. Dazu wird allerdings, wie schon oben erwähnt, zum nächsten Wettkampf eine globale Karte implementiert, in der die komplette Referenztrajektorie

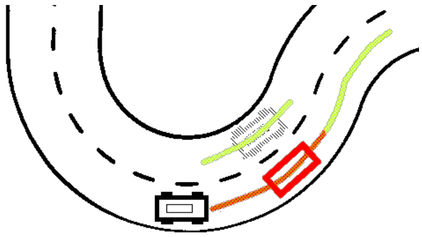

Abb. 12: Verschieben der Referenztrajektorie auf die linke Fahrspur bei erkanntem Hindernis

eingetragen ist und die tatsächliche Position des Fahrzeugs im Weltkoordinatensystem anhand der erkannten Strecke besser geschätzt werden kann.

3.4 Dynamic Reconfigure

In der unmittelbaren Vorbereitung auf den Wettkampf hat sich herausgestellt, dass die kurzen Testzeiten auf der Originalstrecke gut ausgenutzt werden müssen. Deswegen musste bereits im Vorfeld Überlegungen angestellt werden, welche Parameter entscheidenden Einfluss auf die Güte der Regelung haben, um diese gezielt zu beeinflussen. Zur Einstellung der Parameter liefert das genutzte Kommunikations-Framework ROS mit dem Dynamic Reconfigure Paket die ideale Funktion. So kann online während der Fahrt auf dem Parcours auf die wichtigsten Parameter wie die maximale Geschwindigkeit v_{max}, der Koeffizient zur Drosslung der Geschwindigkeit in den Kurven, sowie die Einträge der Wichtungsmatrizen Einfluss genommen werden. Der direkte Eindruck von Änderungen im Verhalten des Fahrzeuges auf der Strecke erleichtert das Tunen des Reglers erheblich.

3.5 Zukünftig geplante Verbesserungen des Algorithmus

Im nächsten Schritt der Entwicklung sollen, sofern dies möglich ist, Schritt für Schritt die angewandten Vereinfachungen revidiert werden. So verspricht die Erhöhung des Stellhorizontes n_c genauere Informationen von der Prädiktion über die zukünftige Bewegung des Fahrzeuges. Damit sollte es möglich sein schon eher vor Kurven zu bremsen und ebenfalls frühzeitiger aus Kurven heraus beschleunigen zu können, um eine im Mittel höhere Geschwindigkeit zu erreichen. Auch die Einbindung des Motordrehmomentes in die Optimierung, sollte eine gezieltere Anpassung der Geschwindigkeit begünstigen.

Andere Teile des oTToCAR-Teams beschäftigen sich außerdem mit der Erstellung einer globalen Karte zur Vorgabe als Referenztrajektorie und Möglichkeit einer genaueren Schätzung des aktuellen Zustands. Diese macht weitere bereits angedeutete Anpassung des Regelalgorithmus möglich, was das Fahrverhalten zudem robuster gegenüber Fehlwahrnehmungen der Fahrbahn gestaltet.

Mit der bereits erfolgreichen Teilnahme am Carolo-Cup 2015 im Rücken und dem weiteren dargelegten Entwicklungspotential steht das oTToCAR-Team dem nächsten Wettkampf im Jahr 2016 zuversichtlich gegenüber.

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

[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] Die Zeit: Dobrindt entwickelt Regeln für Roboterautos. http://www.zeit.de/mobilitaet/2015-02/alexander-dobrindt-selbstfahrende-autos-strassenverkehr

[6] H. Weise: Das intelligent geführte Auto. In: Die Zeit http://www.zeit.de/1992/35/das-intelligent-gefuehrte-auto

[7] J. Adamy: Nichtlineare Regelungen. Springer-Verlag, 2009

[8] 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/

[9] J.-Y. Bouguet: Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

[10] M. Brambilla (Regie): Demolition Man. Film (1993). http://www.imdb.com/title/tt0106697/

[11] P. Kovesi: MATLAB and Octave Functions for Computer Vision and Image Processing. http://www.csse.uwa.edu.au/~pk/research/matlabfns/

[12] 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

[13] 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

[14] Scientif38: Winkelmesser. http://commons.wikimedia.org/wiki/File:Protractor_Rapporteur_Degree_V1.jpg

[15] T. Rose ; M. Treseler: Entwicklung einer Karosserie für das oTToCar.

[16] Tamiya Inc.: Tamiya TT-01 Type-E (TT-01E) Chassis. http://www.rcscrapyard.net/de/tamiya-tt-01-type-e.htm

[17] V. Wiedmeier ; A. Himmel: oTToCAR, Interdisziplinäres Teamprojekt.

[18] 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