

Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen

Tobias Homolla^{*†}, Grischa Gottschalg^{*‡}, Hermann Winner[§]

Zusammenfassung: Automatisiert gesteuerte, mobile Systeme verfügen über eine planende und eine ausführende Instanz, die sowohl integriert als auch unabhängig voneinander vorliegen können. Beide benötigen Informationen über die aktuelle Pose des Systems. Um ein unerwünschtes Systemverhalten zu verhindern, ist es notwendig, dass beide Instanzen identische (konsistente) Informationen über die aktuelle Pose des Systems erhalten. Am Beispiel eines automatisiert gesteuerten Fahrzeugs wird ein Verfahren vorgeschlagen, das ermöglicht, für Planung und Regelung inkonsistente Lokalisierungsdaten zu verwenden. Dazu wird der Offset der ermittelten Posen bestimmt, überwacht und korrigiert. Außerdem werden die Effekte von Sprüngen der Pose im Rahmen der Sensordatenfusion auf die Bewegungsregelung unterdrückt und ein Beitrag zur Selbstwahrnehmung des Systems geleistet.

Schlüsselwörter: Automatisiertes Fahren, Lokalisierung, Sensordatenfusion, Regelung

1 Einführung

Automatisierte mobile Systeme (z.B. automatisierte Kraftfahrzeuge) sind von großem Forschungsinteresse und werden in zahlreichen Projekten in Wissenschaft und Industrie erforscht. Für die automatisierte Fahrzeugführung spielt unter anderem die Rückführungsstrategie der fahrdynamischen Zustandsgrößen im Rahmen der Bewegungsregelung eine wichtige Rolle [1].

1.1 Rückführungsstrategien

In vielen Projekten aus der Forschung und Industrie wird entsprechend der Klassifizierung aus [1] eine High-Level-Stabilisierung verwendet, wie bspw. beim Fahrzeug *BOSS*, Gewinner der *DARPA Urban Challenge 2007* [2]. High-Level-Stabilisierung bedeutet, dass die Trajektorie in jedem Berechnungsschritt der Planung neu berechnet wird, wohingegen bei

*Die markierten Autoren sind Erstautoren mit gleichen Beiträgen.

†Tobias Homolla ist Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, (e-mail: homolla@fzd.tu-darmstadt.de).

‡Grischa Gottschalg ist Wissenschaftlicher Mitarbeiter am Fachgebiet Physikalische Geodäsie und Satellitengeodäsie der TU Darmstadt, Franziska-Braun-Str. 7, 64287 Darmstadt (e-mail: gottschalg@psg.tu-darmstadt.de).

§Hermann Winner ist Professor und Leiter des Fachgebiets Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail: winner@fzd.tu-darmstadt.de)



der Low-Level-Stabilisierung die Trajektorie fest im Raum liegt. Bi-Level-Stabilisierung basiert auf der Low-Level-Stabilisierung und berechnet nur bei Überschreitung von bestimmten Kriterien eine neue Trajektorie, so dass die Vorteile beider Rückführungsstrategien kombiniert werden können.[1]

1.2 Forschungsprojekt UNICAR*agil*

Im Forschungsprojekt UNICAR*agil* arbeitet ein Konsortium aus acht deutschen Universitäten und acht Industriepartnern an disruptiven, modularen Strukturen für agile, automatisierte Fahrzeugkonzepte [3]. Dabei sind einzelne Funktionen als abgeschlossene Software-Dienste mit definierten Schnittstellen konzipiert, die im Rahmen einer Systemarchitektur sinnvoll miteinander kombiniert werden müssen. Dies schließt den Entwurf einer Rückführungsstrategie der fahrdynamischen Zustandssgrößen mit ein.

1.3 Struktur des Beitrags

Bei der Verwendung von mehreren Lokalisierungsfunktionen in einem automatisierten Fahrzeug, wie es in UNICAR*agil* der Fall ist, ergeben sich zahlreiche Möglichkeiten sowie Herausforderungen im Bezug auf die Gestaltung der Fahrzeugführung. In Kapitel 2 wird die in [3] für UNICAR*agil* definierte Systemarchitektur erläutert, bevor in Kapitel 3 auf die mit dieser Architektur verbundenen Herausforderungen eingegangen wird. In Kapitel 4 werden überarbeitete Architekturen vorgestellt und mit Hilfe der definierten Anforderungen eine Lösung ausgewählt, die in Kapitel 5 erläutert wird. Abschließend folgt in Kapitel 6 ein Fazit und Ausblick.

2 Initiale Systemarchitektur

In [3] wird die zu Beginn des Projekts UNICAR*agil* angedachte Systemarchitektur beschrieben, die als Basis für die weiteren Betrachtungen genutzt wird. Das Fahrzeug verfügt dabei über eine serviceorientierte Systemarchitektur mit drei makroskopischen Systemebenen: Großhirn, Stammhirn und Rückenmark. Im Großhirn findet die Umfeldwahrnehmung sowie die Verhaltens- und Trajektorienplanung statt. Im ausfallsicher gestalteten Stammhirn ist die Fahrdynamik- und Trajektorienregelung (FTR) und das sog. Sichere Anhalten verortet, während die Aktorik des Fahrzeugs dem Rückenmark zugeordnet ist. Durch die vorhandene Einzelradaktorik verfügt das Fahrzeug über den zusätzlichen, von der Bewegungsrichtung unabhängigen Freiheitsgrad des Gierwinkels.

Das Fahrzeug verfügt über zwei unabhängige Systeme zur Ermittlung der derzeitigen Pose (Position + Ausrichtung). Im Großhirn ist die kamera- und kartenbasierte Lokalisierung (KKLo) angesiedelt, die mittels eines Abgleichs von Landmarken mit einer digitalen Karte die Pose des Systems bestimmt. Im Stammhirn befindet sich die Fahrdynamikzustandsschätzung (FZS), die Informationen aus Navigationssatelliten, Odometriesensorik sowie einer inertialen Messeinheit nutzt, um ebenfalls eine Pose zu schätzen.

Durch die serviceorientierte Softwarearchitektur verfügt das System für die Fahrzeugführung über eine Wirkkette hintereinandergeschalteter Dienste. Aufbauend auf der Wahrnehmung des Umfelds durch entsprechende Sensorik wird das Verhalten des Fahrzeugs und anschließend eine Soll-Trajektorie geplant. Diese Soll-Trajektorie besteht aus

einer zeitgestempelten Abfolge von Posen und ihrer beiden zeitlichen Ableitungen und wird nur bei einer Änderung des dynamischen Umfelds, nicht jedoch bei einer Regelabweichung angepasst. Die nachgelagerte FTR führt einen Soll-Ist-Vergleich durch, für den die Soll-Trajektorie und der Ist-Zustand des Fahrzeugs aus der FZS verwendet werden, und generiert Stellbefehle für die Aktorik des Fahrzeugs. Bezogen auf die Bewegungsregelung des Fahrzeugs ist die Trajektorienplanung daher im Rahmen der initialen Architektur als Steuerung (Open-Loop) ausgelegt, die Aufgaben der Trajektorienplanung und -regelung sind entkoppelt. Jeder Schritt dieser Wirkkette ist latenzbehaftet. Daher besteht ein zeitlicher Verzug zwischen dem Messzeitpunkt der Wahrnehmung des Umfelds bis zur Umsetzung einer Reaktion durch die Aktorik.

Eine Besonderheit des UNICAR*agil*-Konzepts stellt die Funktion Sicheres Anhalten dar. Dieser separate Betriebsmodus steht als Rückfallebene zur Verfügung, um bei einem Ausfall des Großhirns das Fahrzeug in einen sicheren Zustand zu überführen. Dazu wird von der Trajektorienplanung zusätzlich zur Soll-Trajektorie in jedem Zeitschritt eine Notbahn geplant, die aus einer Abfolge von Posen und Soll-Geschwindigkeiten besteht. Der Dienst Sicheres Anhalten überprüft diese Notbahn mithilfe separater Umfeldsensorik auf Belegung, reduziert wenn nötig die Geschwindigkeit und transformiert die Bahn damit in eine Trajektorie, die als Eingang für die FTR dient. Für die FTR ist daher keine Kenntnis des Betriebsmodus notwendig, da eine identische Schnittstelle vorliegt. Abbildung 1 stellt die relevante Systemarchitektur dar, wie sie in [3] vorgesehen ist.

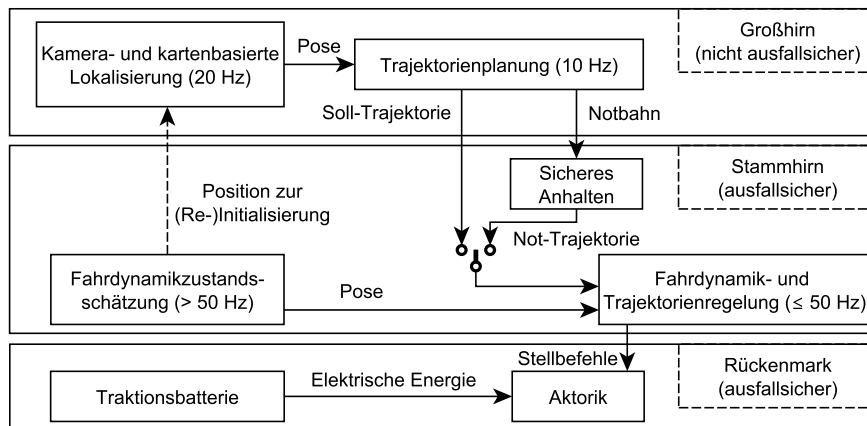


Abbildung 1: Wirkkette der Bewegungsregelung im Projekt UNICAR*agil* (gemäß [3])

3 Problemstellung

Aus der in [3] vorgestellten Architektur ergeben sich mehrere Herausforderungen, die für eine Umsetzung zu bestehen sind. Zunächst wird die Abbildung 1 betrachtet. Die planerische und die ausführende Ebene greifen auf zwei unterschiedliche Lokalisierungsfunktionen zurück. Die Soll-Trajektorie wird auf Basis der Pose der KKLo in der Trajektorienplanung ermittelt, wohingegen die Ist-Werte des Fahrerdynamikzustands für die FTR in der FZS berechnet werden. Innerhalb der FTR wird ein Soll-Ist-Vergleich durchgeführt. Inkonsistenzen in den Lokalisierungsinformationen der zwei Lokalisierungsfunktionen können daher zu scheinbaren Regelabweichungen führen, die nicht zu Regeleingriffen führen dürfen,

da sie nicht auf einer realen Abweichung zwischen den Soll- und Ist-Werten beruhen. Eine solche Inkonsistenz tritt bspw. durch eine Ungenauigkeit der Karte der KKLo [4] oder bei einem Drift in der Position der FZS aufgrund einer Abschattung des globalen Navigationssatellitensystems (GNSS) auf. Es ergibt sich daher die Notwendigkeit, die Systemarchitektur aus Abbildung 1 zu überarbeiten, um solche unerwünschten Regeleingriffe zu verhindern.

Außerdem ist in Abbildung 1 zu erkennen, dass die FTR mit einer Taktrate von bis zu 50 Hz arbeitet. Daraus ergibt sich als weitere Anforderung an eine neue Systemarchitektur, dass die für die Regelung benötigten Lokalisierungsinformationen mindestens mit der Taktrate der FTR zur Verfügung stehen. Des Weiteren folgt eine Anforderung aus den im vorherigen Abschnitt erläuterten Betriebsmodi in UNICAR*agil* und der Annahme, dass nur Einzelfehler betrachtet werden. Die Fahrzeugführung muss sowohl im Regelbetrieb als auch bei Einzelfehlern im Modus Sicheres Anhalten sichergestellt werden. Bei der Überarbeitung der Systemarchitektur sind Schleifenbildungen in der Informationsverarbeitung zu vermeiden, da diese im Falle nicht berücksichtigter Korrelationen zu Fehlfunktionen bis hin zur Instabilität des Gesamtsystems führen können.¹ Im Projekt UNICAR*agil* benötigt die KKLo für (Re-)Initialisierungen eine absolute Position, die durch die FZS zur Verfügung gestellt wird. Ein Beispiel für eine Schleifenbildung in der Informationsverarbeitung wäre eine Verarbeitung der Pose aus der KKLo im Filter der FZS, woraus eine Korrelation der beiden Posen folgt. Bei der anschließenden Verarbeitung der Posen aus der FZS in der KKLo zur (Re-)Initialisierung würde eine Schleife in der Informationsverarbeitung entstehen.

Zusammenfassend ergeben sich vier Anforderungen an die Systemarchitektur, die in Tabelle 1 dargestellt sind.

Tabelle 1: Anforderungen an die Systemarchitektur

Nr	Anforderung
A1	Regeleingriffe durch systematischen Offset der Ist-Posen der beiden Lokalisierungsfunktionen werden verhindert.
A2	Die benötigten Größen des Fahrdynamikzustands aus der Lokalisierung stehen mindestens mit der Datenrate der FTR zur Verfügung.
A3	Die automatisierte Fahrzeugführung ist sowohl im Regelbetrieb als auch im Betriebsmodus Sicheres Anhalten sichergestellt.
A4	Schleifenbildung in der Informationsverarbeitung wird vermieden.

4 Überarbeitung der Systemarchitektur

Im vorangegangenen Kapitel wurde aufgezeigt, weshalb die in [3] präsentierte Systemarchitektur nicht ohne Modifikation umgesetzt werden kann. Des Weiteren wurden Anforderungen abgeleitet, die bei der Wahl einer finalen Systemarchitektur zu erfüllen sind. Innerhalb dieses Kapitels werden zunächst verschiedene Lösungsvarianten (LV) für die Systemarchitektur des Fahrzeugs aufgezeigt. Dabei werden nur LV betrachtet, die sich

¹Bei einer Schleifenbildung bzw. Rückkopplung ist eine positive Kopplung möglich. Diese führt zur Selbstverstärkung im System, welche die Instabilität des Regelkreises zur Folge haben kann [5, S. 56].

signifikant voneinander unterscheiden. Durch Abgleich mit den zuvor ermittelten Anforderungen findet anschließend eine Auswahl einer Systemarchitektur statt.

4.1 Mögliche Architekturen

4.1.1 High-Level-Stabilisierung

Die Bewegungsregelung kann als High-Level-Stabilisierung konzipiert werden (Abbildung 2A). Dabei werden die von der Trajektorienplanung bzw. dem Sicheren Anhalten berechneten Geschwindigkeits- und Beschleunigungsvorgaben in Abhängigkeit des Fahrzeugzustands ($x(t)$) angepasst, was einer Posenregelung entspricht. Die unterlagerte FTR im Stammhirn ist als Geschwindigkeits- und Beschleunigungsregelung ausgelegt. Die Trajektorienplanung und die Posenregelung des Fahrzeugs finden in diesem Fall auf Basis der von der KKLo bestimmten Pose statt. Die Ausgangsgrößen der FZS werden ausschließlich zur Geschwindigkeits- und Beschleunigungsregelung innerhalb der FTR genutzt, um Stellgrößen für die Aktorik ($u(t)$) zu generieren.

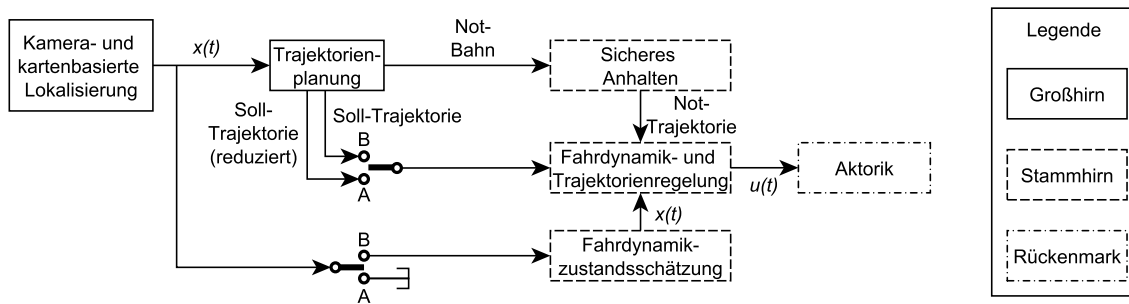


Abbildung 2: Systemarchitekturen für LV1 (A) und LV2 (B)

4.1.2 Low-Level-Stabilisierung mit Integration der KKLo-Pose in FZS-Filter

LV2 stellt die Einbeziehung der von der KKLo ermittelten Pose in das stochastische Filter der FZS dar. Die auf diese Weise zusätzlich verfügbare Pose wird als Stützinformation in der Sensordatenfusion der FZS verwendet. Durch eine entsprechende Gewichtung der Eingangsdaten können Inkonsistenzen zwischen den beiden Posen im System reduziert werden. Die resultierende Architektur entspricht Abbildung 2B. Im Unterschied zu LV1 liegt eine Low-Level-Stabilisierung vor, so dass die Posenregelung innerhalb der FTR stattfindet. Die Trajektorienplanung bzw. das Sichere Anhalten passt die Soll-Trajektorie in diesem Fall nicht in Abhängigkeit des Fahrzeugzustands an.

4.1.3 Low-Level-Stabilisierung mit Posen-Offsetkorrektur

LV3 ist eine Architektur mit Low-Level-Stabilisierung. Die Trajektorienplanung nutzt dabei die KKLo-Pose, die FTR die Pose der FZS (vgl. Abbildung 3). Um dabei die in Kapitel 3 dargelegten Probleme zu vermeiden, ist bei dieser Architektur ein zusätzlicher Berechnungsschritt im Form einer Offsetkorrektur notwendig. Dabei werden zunächst die Abweichungen zwischen den von den beiden Lokalisierungsfunktionen ermittelten Posen berechnet. Anschließend werden entweder die Ist-Posen der FZS (Variante A) oder die

in der Soll-Trajektorie bzw. Notbahn enthaltenen Posen (Variante B) um diesen Offset korrigiert.

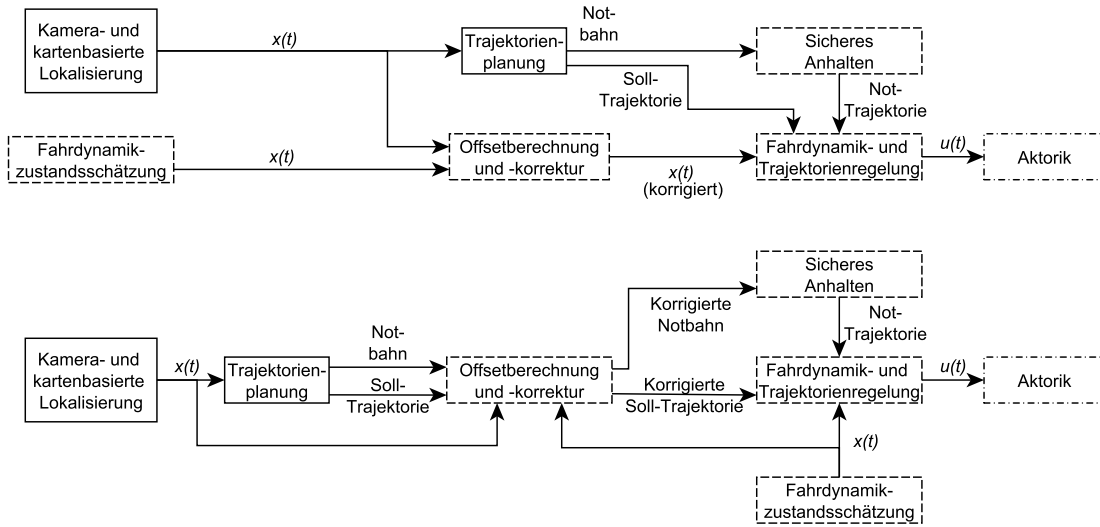


Abbildung 3: Systemarchitektur für LV3 mit den beiden Varianten A (oben) und B (unten)

4.1.4 Einheitliche Lokalisierungs-Pipeline

LV4 umfasst eine Architektur mit Low-Level-Stabilisierung und den Aufbau einer gemeinsamen Lokalisierungs-Pipeline mit der gesamten vorhandenen Lokalisierungssensorik (KKLo und FZS). Dieser Ansatz ist in [6] beschrieben.² Die so ermittelte Pose wird sowohl für die Trajektorienplanung als auch für die FTR genutzt (vgl. Abbildung 4).

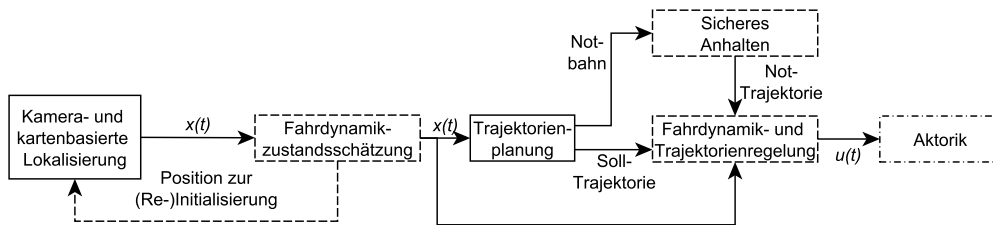


Abbildung 4: Systemarchitektur für LV4

4.1.5 Separater Posenregler im Großhirn

LV5 basiert auf einer Architektur mit Low-Level-Stabilisierung (vgl. Abbildung 5). Die Trajektorienplanung im Großhirn plant die Trajektorie auf Basis der KKLo-Pose. Die von der Trajektorienplanung unabhängige Posenregelung befindet sich dabei ebenfalls im Großhirn und nutzt für den Soll-Ist-Vergleich ebenfalls die KKLo-Pose. Die FTR im Stammhirn reduziert sich auf die Funktion einer Geschwindigkeits- und Beschleunigungsregelung.

²Bei dieser Lösung können die gegenseitigen Korrelationen berücksichtigt werden, da der Quellcode aller Teilfunktionen bei der Entwicklung der Pipeline zur Verfügung steht.

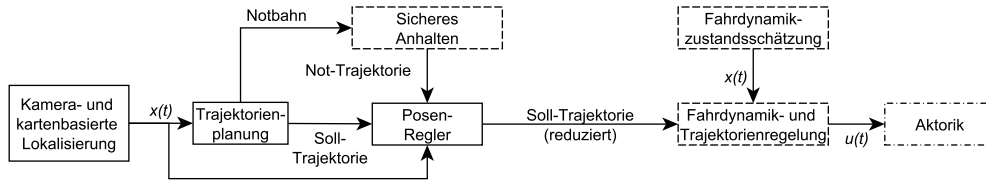


Abbildung 5: Systemarchitektur für LV5

4.1.6 Low-Level-Stabilisierung mit Soll-Ist-Vergleich im Großhirn

LV6 nutzt ebenfalls eine Low-Level-Stabilisierung, verlagert jedoch den Soll-Ist-Vergleich für die Posenregelung in das Großhirn. Dabei wird ausschließlich die KKLo-Pose genutzt. Die Regelabweichung wird an das Stammhirn übertragen, in dem der eigentliche Posenregler angesiedelt ist. Diese Architektur bietet den Vorteil, dass keine Inkonsistenzen beim Soll-Ist-Vergleich der Posen entstehen kann, da ausschließlich die von der KKLo ermittelte Pose verwendet wird. Die Architektur entspricht LV5, wenn der Posenregler durch einen Dienst zur Bestimmung der Posenabweichung ersetzt wird und statt der Pose aus der FZS die KKLo Pose verwendet wird (siehe Abbildung 6).

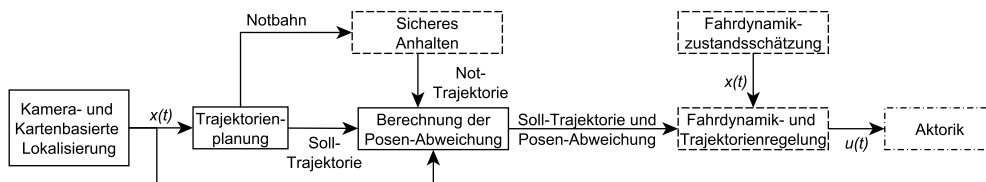


Abbildung 6: Systemarchitektur für LV6

4.2 Auswahl eines Verfahrens

Auf Basis der definierten Anforderungen und des aufgezeigten Lösungsraums wird nachfolgend die Entscheidung für eine Systemarchitektur getroffen.

LV1 steht im Widerspruch zu Anforderung A3, nach der die FTR in allen Betriebsmodi in der Lage sein muss, die Fahrzeugführung zu gewährleisten. Im Betriebsmodus Sicheres Anhalten muss als Worst-Case-Szenario von einem vollständigen Ausfall des Großhirns ausgegangen werden. Bei einer High-Level-Stabilisierung mit Posenregelung im Großhirn ist ein Abfahren der vorab berechneten Notbahn mit dem bestehenden Konzept für das Sichere Anhalten nicht möglich. Aus demselben Grund sind LV5 und LV6 zu verwerfen.

LV2 ist für die vorliegende Systemarchitektur nicht geeignet, da aufgrund der nicht bekannten Korrelationen die Gefahr von Schleifenbildungen (positive Kopplung) besteht³ und eine Stabilität des Gesamtsystems daher u.U. nicht garantiert werden kann (vgl. Abschnitt 3). Dieselbe Gefahr besteht bei LV4, da die Position der FZS zur Initialisierung der KKLo benötigt wird. LV3 Variante B verhindert die rechtzeitige Unterdrückung

³Eine Berücksichtigung aller Korrelationen zwischen den Modulen ist nicht möglich, da die einzelnen Module von verschiedenen Partnern erarbeitet werden und der zugehörige Quellcode nicht vollständig zur Verfügung steht.

von Sprüngen der FZS im Betriebsmodus Sicheres Anhalten, da der Dienst Sicheres Anhalten mit einer niedrigeren Frequenz als die FTR ausgeführt wird und die korrigierte Trajektorie deshalb zeitverzögert an die FTR weiterleitet, was zu einer unerwünschten Regelabweichung führt. Dies entspricht einem Verstoß gegen Anforderung A3.

Aufgrund der nicht erfüllten Anforderungen der LV1-2, LV3B und LV4-6 stellt einzig die Low-Level-Stabilisierung mit Posen-Offsetkorrektur in Variante A (LV3A) eine geeignete Systemarchitektur dar und wird daher weiter verfolgt.

Tabelle 2: Übersicht der LV und der ermittelten Ausschlusskriterien

Nr.	Beschreibung	Verstoß gegen
LV1	High-Level-Stabilisierung	A3
LV2	Low-Level-Stabilisierung mit Einbezug der KKLo-Pose in FZS	A4
LV3A	Low-Level-Stabilisierung mit Posen-Offsetkorrektur	Keine
LV3B	Low-Level-Stabilisierung mit Posen-Offsetkorrektur	A3
LV4	Einheitliche Lokalisierungs-Pipeline	A4
LV5	Low-Level-Stabilisierung mit Posenregler im Großhirn	A3
LV6	Low-Level-Stabilisierung mit Soll-Ist-Vergleich im Großhirn	A3

5 Offsetkorrektur

Die im vorausgegangenen Kapitel ausgewählte Offsetkorrektur lässt sich in drei Unterfunktionen einteilen: die Grundfunktionalität zur Bestimmung des Posenoffsets und zur Umrechnung der Soll-Trajektorie aus der Trajektorienplanung und der Ist-Pose aus der FZS in ein gemeinsames Bezugssystem, eine Zusatzfunktion zur Vermeidung von unerwünschten Effekten von Sprüngen der Ist-Pose aus der FZS auf die FTR und eine Zusatzfunktion als Beitrag zur Selbstbeobachtung.

5.1 Funktionalität

5.1.1 Aufbau

Wie in Abbildung 3 zu erkennen ist, wird die Offsetkorrektur zwischen der planerischen und der ausführenden Ebene verschaltet und besteht aus der Berechnung und der Korrektur des Offsets. Die Soll-Trajektorie wird auf Basis der Ist-Pose aus der KKLo geplant. Die j -te Soll-Trajektorie T_j enthält Soll-Posenabfolgen $P_{j,i}$ mit Zeitstempeln $t_{j,i}$, deren zeitlichen Ableitungen $\dot{P}_{j,i}$ und $\ddot{P}_{j,i}$ sowie die Ist-Pose $P_{j,U}$ aus der das Umfeld (U) betrachtenden KKLo, auf der die Trajektorie geplant wurde, mit dem Zeitstempel $t_{j,U}$ an dem diese Pose gültig war:

$$T_j = \left\{ (t_{j,U}, P_{j,U}) ; \left(t_{j,i}, P_{j,i}, \dot{P}_{j,i}, \ddot{P}_{j,i} \right)_{i=1,\dots,n} \right\} \quad (1)$$

Dabei steht P_j für eine kontinuierliche Posenabfolge und $P_{j,i}$ für eine diskretisierte Posenabfolge (gilt ebenso für die Ableitungen). Die Ist-Pose $P_{j,U}$ aus Gleichung (1) wird an die Offsetberechnung übermittelt.

Außerdem erhält die Offsetberechnung die Ist-Pose der FZS $P_F(t_k)$ zum aktuellen Zeitpunkt t_k . Der bestimmte Offset wird zusammen mit den Größen $P_F(t_k)$, $\dot{P}_F(t_k)$, $\ddot{P}_F(t_k)$ der FZS an die Offsetkorrektur gesendet. Die korrigierten Größen $P_{F,\text{korr}}(t_k)$, $\dot{P}_{F,\text{korr}}(t_k)$, $\ddot{P}_{F,\text{korr}}(t_k)$ werden von der Offsetkorrektur an die FTR übermittelt. Diese berechnet daraus zusammen mit der Soll-Trajektorie und den anderen Fahrdynamikgrößen aus der FZS die Stellgrößen für die Aktorik des Fahrzeugs.

5.1.2 Grundfunktionalität

Die Funktion der Offsetberechnung und -korrektur lässt sich folgendermaßen zusammenfassen: Die empfangenen Ist-Posen der FZS $P_F(t_k)$ werden zusammen mit den zugehörigen Zeitstempeln t_k in einem Ringspeicher vorgehalten. Zunächst wird im Rahmen der Offsetberechnung ein korrespondierendes Posen-Paar aus Ist-Pose der FZS $P_F(t_{j,U})$ und der Ist-Pose der KKL0 $P_{j,U}$, auf deren Basis die aktuelle Soll-Trajektorie berechnet wurde, ermittelt. Im Regelfall existiert keine Ist-Pose der FZS $P_F(t_k)$ mit $t_k = t_{j,U}$ und es wird daher zwischen den zwei nächstgelegenen Zeitpunkten interpoliert. Anschließend wird die systematische Abweichung zwischen den beiden Posen („Offset“) bestimmt:

$$\Delta P_{j,U} = P_F(t_{j,U}) - P_{j,U} \quad (2)$$

Die Abweichung zwischen den beiden Posen $\Delta P_{j,U}$ setzt sich aus der Abweichung in der Position und der Abweichung in der Orientierung zusammen, wobei die Matrix $\Delta \mathbf{R}_{j,U}$ die Rotation um die Abweichungen in der Orientierung beschreibt.

Nachfolgend werden im Rahmen der Offsetkorrektur die Ist-Posen $P_F(t_k)$ um den ermittelten Offset der Posen sowie die Ist-Geschwindigkeiten $\dot{P}_F(t_k)$ und Ist-Beschleunigungen $\ddot{P}_F(t_k)$ um den Offset in der Orientierung korrigiert:

$$P_{F,\text{korr}}(t_k) = P_F(t_k) - \Delta P_{j,U} \quad (3)$$

$$\dot{P}_{F,\text{korr}}(t_k) = \Delta \mathbf{R}_{j,U} \dot{P}_F(t_k) \quad (4)$$

$$\ddot{P}_{F,\text{korr}}(t_k) = \Delta \mathbf{R}_{j,U} \ddot{P}_F(t_k) \quad (5)$$

Auf diese Art und Weise führen Abweichungen der beiden Lokalisierungsfunktionen nicht zu einer Regelabweichung und damit zu Stellbefehlen für die Aktorik. Des Weiteren wird die Konsistenz der Geschwindigkeits- und Beschleunigungsvorgaben sichergestellt.

5.1.3 Zusatzfunktionen

Wie zu Beginn des fünften Abschnitts aufgezeigt wurde, enthält die Offsetkorrektur zwei Zusatzfunktionalitäten.

Die erste Zusatzfunktion unterdrückt unerwünschte Effekte auf die FTR, die durch Sprünge in der Ist-Pose der FZS entstehen würden. Diese Sprünge in der Ist-Pose der FZS treten nur bei definierten Schritten im Verarbeitungsablauf, wie bspw. beim Messupdate im Kalman-Filter oder einer (Re-)Initialisierung des Filters auf. Bei diesen Verarbeitungsschritten gibt es neben dem Sprung in der Ist-Pose auch einen Sprung in der zugehörigen Standardabweichung. Daher können die Sprünge bei gleichzeitiger Betrachtung der Verläufe von Ist-Pose und dessen Standardabweichung erkannt werden. Um die rechtzeitige Erkennung der Sprünge vor einer Weitergabe an die FTR sicherzustellen, läuft die Offsetkorrektur mit der gleichen Datenrate wie die FTR. Wenn ein Sprung erkannt

wird, wird dessen Höhe in der Offsetkorrektur ermittelt und der Einfluss des Sprungs auf die FTR kompensiert. Dazu wird die Veränderung der Pose der vergangenen zwei Zeitschritte verwendet, um die erwartete Posenänderung aufgrund der Fahrzeugbewegung im aktuellen Zeitschritt zu präzisieren. Der verbleibende Anteil der Posenänderung wird auf den Sprung zurückgeführt und herausgerechnet.

Die zweite Zusatzfunktion leistet einen Beitrag zur Selbstwahrnehmung, indem zuvor definierte Kriterien überprüft werden. Diese können bspw. Grenzen für den Posen-Offset oder dessen zeitliche Änderung enthalten. Der aktuelle Posen-Offset sowie ggf. die Nichteinhaltung der Kriterien werden an die Selbstbeobachtung des Gesamtsystems zur Funktionsüberwachung gemeldet.

5.2 Programmablauf

Die Offsetberechnung und -korrektur läuft in mehreren Schritten ab, wobei die Offsetberechnung mit der Datenrate der Trajektorienplanung und die Offsetkorrektur mit der Datenrate der FTR läuft. Die Offsetberechnung wird demnach nur ausgeführt, wenn eine neue Soll-Trajektorie vorliegt, wohingegen die Offsetkorrektur immer vor einem Berechnungsschritt der FTR ausgeführt wird. Im ersten Schritt der Offsetkorrektur wird der berechnete Offset, wie in Abschnitt 5.1.2 dargelegt wird, mit den Größen aus der FZS verrechnet. Falls ein Sprung in der Ist-Pose der FZS erkannt wird, wird dieser im zweiten Schritt kompensiert. Im dritten Schritt wird die Überprüfung der definierten Kriterien durchgeführt (vgl. Abschnitt 5.1.3).

5.3 Implementierung und Testkonzept

Das vorgestellte Verfahren wird im weiteren Verlauf des Projekts UNICAR*agil* im entwickelten Forschungsfahrzeug implementiert. Der erste Schritt der Implementierung ist eine Simulation als Proof-of-Concept. Als Testszenario wird die Durchfahrt eines Tunnels gewählt. Dieses Szenario ist für einen Proof-of-Concept geeignet, da im Tunnel ein signifikanter Unterschied zwischen den ermittelten Posen der beiden Lokalisierungsfunktionen erwartet wird. Während die Genauigkeit der FZS im Tunnel durch die Abschattung des Satellitenempfangs stark beeinträchtigt ist, trifft dies auf die Genauigkeit der KKLo nicht zu, da die KKLo auf Landmarken basiert. Ohne Satellitenempfang sind in der FZS keine GNSS-Messupdates möglich. Da Inertialsensorik und Odometrie keine absolute Position messen, wächst der Positionsfehler mit der Zeit an (Drift).

Als Test werden die definierten Anforderungen an die Positionierungsgenauigkeit des Fahrzeugs herangezogen, wobei diese Genauigkeit den Unterschied zwischen Soll-Pose und in der Simulation bekannter Ist-Pose (ground truth) beschreibt. Durch den im Tunnel auftretenden Drift in der durch die FZS geschätzten Pose wird ein Einhalten der Anforderungen im Tunnel verhindert. Im Rahmen des Proof-of-Concepts wird gezeigt, dass die Anforderungen in mehreren Simulationen eingehalten werden können. Wird ein statistisch signifikantes Ergebnis erreicht, gilt der Test als bestanden.

Des Weiteren ist es möglich, anhand des Szenarios die erste Zusatzfunktion des Verfahrens zu testen, die den Einfluss von Sprüngen in der Position der FZS unterdrückt. Ein solcher Sprung tritt bspw. auf, wenn die im Tunnel angewachsenen Fehler in der geschätzten Pose der FZS nach Ausfahrt aus dem Tunnel durch ein GNSS-Messupdate korrigiert

werden. Durch die nach einem solchen Messupdate signifikant gesunkene geschätzte Standardabweichung der durch die FZS ermittelten Position ist das Erkennen des Sprungs möglich und der nächste Korrekturschritt wird frühzeitig durchgeführt. Auf diese Weise wird das Entstehen einer unerwünschten Regelabweichung auch hier verhindert.

6 Zusammenfassung und Ausblick

Im Rahmen dieses Beitrags wurden verschiedene Systemarchitekturen für die Bewegungsregelung serviceorientierter, mobiler technischer Systeme abgeleitet und diskutiert. Anhand generierter Anforderungen an das System wurde gezeigt, weshalb etablierte Architekturen nicht geeignet sind, um die mit der Nutzung unterschiedlicher Lokalisierungsfunktionen für Planung und Regelung verbundenen Herausforderungen zu bestehen. Anschließend konnte gezeigt werden, wie eine neuartige Architektur mit integrierter Posen-Offsetkorrektur eine geeignete Lösung der Problematik darstellt. Die Offsetkorrektur erlaubt es, einen Einfluss inkonsistenter Lokalisierungsdaten auf die Fahrzeugführung zu verhindern und eröffnet daher neue Gestaltungsspielräume bei der Entwicklung modularer Systeme. Des Weiteren ist es mit dem vorgeschlagenen Verfahren möglich, den Einfluss von Sprüngen in der Lokalisierung auf die Regelung zu reduzieren und die Selbstwahrnehmung des Systems zu unterstützen.

Die Anwendung des präsentierten Verfahrens ist nicht auf automatisierte Kraftfahrzeuge beschränkt, sondern kann immer dort erfolgen, wo ein mobiles System mit getrennter planerischer und ausführender Ebene sowie mit mehr als einer Lokalisierungsfunktion zum Einsatz kommt.

Danksagung

Diese Forschungsarbeiten wurden im Rahmen des Projekts „UNICAR^{agil}“ durchgeführt (FKZ 16EMO0286). Wir bedanken uns für die finanzielle Unterstützung des Projekts durch das Bundesministerium für Bildung und Forschung (BMBF).

Literaturverzeichnis

- [1] Moritz Werling. *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien: Zugl.: Karlsruhe, KIT, Diss., 2010*. Print on demand. Vol. 34. Schriftenreihe des Instituts für Angewandte Informatik - Automatisierungstechnik, Universität Karlsruhe (TH). Hannover and Karlsruhe: Technische Informationsbibliothek u. Universitätsbibliothek and KIT Scientific Publishing, 2011. ISBN: 978-3-86644-631-1. DOI: 10.5445/KSP/1000021738. URL: <https://edocs.tib.eu/files/e01fn12/685609162.pdf>.
- [2] Chris Urmson, Joshua Anhalt, et al. “Autonomous driving in urban environments: Boss and the Urban Challenge”. In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466. ISSN: 15564959. DOI: 10.1002/rob.20255.

- [3] Timo Woopen, Bastian Lampe, et al. “UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts”. In: *27th Aachen Colloquium*. Aachen, Germany, Oct. 2018.
- [4] Clément Fouque and Philippe Bonnifait. “Navigable map-aided differential odometry to enhance GNSS in adverse conditions”. In: *Accurate Localization for Land Transportation*. Paris, France, 2009, p. 5. URL: <https://hal.archives-ouvertes.fr/hal-00445307>.
- [5] Bernard P. Zeigler, Herbert Prähofer, and Tag Gon Kim. *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems*. 2. ed., transferred to digital print. Amsterdam and Heidelberg: Academic Press, 2000. ISBN: 0-12-778455-1.
- [6] Chassis Systems Control, Robert Bosch GmbH. *Safe GNSS/Inertial Positioning by VMPS for Highly Automated Driving*. 2018. URL: <https://www.bosch.co.jp/aee2018/pdf/bosch-automotive-engineering-exposition-2018-automated-cc-vehicle-motion-and-position-sensor-vmmps-02.pdf>.