

Mehrstufiges Planungskonzept für pilotierte Parkhausfunktionen

Dipl.-Ing. **D. Lenz**, fortiss GmbH, München;
P. Minnerup, M.Sc., fortiss GmbH, München;
Dipl.-Ing. **C. Chen**, fortiss GmbH, München;
Dipl.-Ing. **E. Roth**, Audi Electronics Venture, Gaimersheim;

Kurzfassung

In diesem Beitrag adressieren wir die Bahn- und Trajektorienplanung eines vollautonomen Fahrzeuges, das sich innerhalb eines Parkhauses bewegt und selbständig ein- und ausparkt. Wir schlagen hierzu eine hierarchische Struktur vor, die das gesamte Planungsproblem in auf Steuergeräteebene beherrschbare Teilschritte zerlegt. Dabei wird vor allem Wert gelegt auf eine möglichst geringe Kommunikation mit dem Parkhaus sowie dem Fahrzeug als letztendliche Entscheidungsinstanz im abgesicherten Bereich des Parkhauses. Erste Ergebnisse in einer Simulationsumgebung werden vorgestellt.

1. Einleitung

Im Bestreben mehr Komfort, Sicherheit und Zeitersparnis für den Autofahrer zu erreichen sind Fahrerassistenzsysteme vor allem im Bereich der Teil- und Vollautomatisierung ein wesentlicher Faktor. Dabei sollen ermüdende und zeitraubende Tätigkeiten vom Fahrzeug teilweise oder vollständig übernommen werden. Ein typischer Anwendungsfall stellt das pilotierte Fahren im Parkhaus in Form des *Parkhauspiloten* dar. Hierbei stellt der Nutzer sein Fahrzeug in einer Übergabezone am Eingang eines Parkhauses ab. Anschließend fährt das Fahrzeug selbstständig bis zu einem Stellplatz und parkt dort ein. Bei Abholung fährt das Fahrzeug wiederum selbstständig bis zur Übergabezone zurück, an der es der Fahrer wieder in Empfang nimmt.

Ein wichtiger Aspekt eines solchen Fahrerassistenzsystems ist eine sichere und robuste Bahnplanung. Bereits auf der CES 2013 haben wir einen Prototyp des Parkhauspiloten demonstriert. Anhand des Parkhaus-Gebäudeplans werden ein statisches Umfeldmodell und eine Referenzbahn generiert. Ein Parkhaus-Server speichert diese Informationen und berechnet auf Anfrage eine Bahn für die gesamte Fahrt, einschließlich Transit und Ein- oder Ausparkvorgang. Der Fahrzeug-Rechner steuert das Fahrzeug um der Bahn zu folgen und meldet dem Parkhaus, wenn unerwartet ein Hindernis im Fahrweg auftaucht. In diesem Fall wird das Fahrzeug angehalten und erst nach Erhalt einer neuen Bahn vom Parkhaus-Server

kann die Fahrt weiter fortgesetzt werden. Im Gegensatz zu dem hier vorgestellten Ansatz legte dieses System mehr Intelligenz auf die Infrastruktureseite. Dadurch ist mehr Aufwand bei der Aufrüstung, Instandhaltung und Sicherstellung der Kompatibilität mit heutigen Parkhaus-Infrastrukturen erforderlich.

2. Stand der Technik

Viele der zur DARPA-Urban Challenge 2007 entwickelten Forschungsprototypen haben die Fähigkeit in einer semistrukturierten Umgebung zu navigieren. In der Arbeit von Rainer Kümmerle, et al [1] wird die Möglichkeit gezeigt, dass ein Fahrzeug mit umfangreicher Sensorausstattung, insbesondere dem Velodyne Laserscanner, vollautomatisiert in ein mehrstufiges Parkhaus fährt. Jedoch ist das gesamte Sensoren- und Rechner-System nicht geeignet für eine serienreife Lösung.

Ein anderes Projekt, das V-Charge-Forschungsprojekt [2] konzentriert sich auf die Umsetzung eines autonomen Ladesystems für Fahrzeuge mit seriennaher Sensorausstattung. Ähnlich wie bei dem hier vorgestellten Ansatz, wird das Straßennetz in Form eines gerichteten Graphen repräsentiert auf dessen Basis mit Hilfe des A-Stern-Algorithmus eine globale Bahn geplant wird. Das Ergebnis der Planung ist eine Folge von Segmenten, welche in einem nachfolgenden Schritt weiter optimiert und als Referenzbahn für die Trajektorien-Planung benutzt, die schließlich auf dynamische Hindernisse reagiert. Einen ähnlichen Ansatz, findet sich auch in der Arbeit von Löper et al. [5] wieder. Dort wird auf Basis einer Gesamtkarte und Straßengraph eine Route geplant und durch Aufteilung in Manöver wie *Spur-Folgen*, *Abbiegen*, *Stehen*, etc. eine Verteilung der Planungskomplexität sowie eine Wiederverwendung von Planungsalgorithmen zwischen verschiedenen Manövern erreicht.

Neben Forschungsarbeiten hat auch die Automobilindustrie verschiedene hochautomatisierte Fahrzeuge vorgestellt, die selbst in einer Parkhaus-Umgebung ein- und ausparken können, unter anderem, Volvo und Nissan. Das Zusammenspiel mit der Infrastruktur ist dort aber wenig adressiert.

Keiner der vorgestellten Ansätze adressiert dabei hinreichend die Speicher-, Kommunikation und Rechenleistungseinschränkungen von einem serientauglichen System und lässt somit Fragen der Skalierbarkeit für beliebig große und komplexe Parkhäuser offen.

3. Problembeschreibung

Das mehrstufige Planungskonzept für pilotierte Parkhausfunktionen orientiert sich im Hinblick einer praxistauglichen Umsetzung an nachfolgenden Randbedingungen: pilotiert befahrbare

Parkhäuser sollen gegenüber existierenden Parkhausneubauten nur einen geringen Mehraufwand bzgl. infrastrukturseitiger Überwachungssensorik zur Detektion von Fahrzeug- und Objektbewegungen, z.B. mittels Videokameras, erfordern. Der bidirektionale Datenaustausch zwischen Parkhaus und Fahrzeugen soll überall im Parkhaus mittels drahtloser Car2-X Technologie möglich sein. Verbindungsabbrüche im Sekundenbereich dürfen die pilotierte Funktionsausführung nicht beeinträchtigen. Die pilotierten Fahrzeuge müssen mit entsprechender Umfeld-, Eigenbewegungs- und Lokalisierungssensorik sowie Auswerte- und Entscheidungsalgorithmik ausgestattet sein, so dass diese im pilotierten Betrieb jederzeit unabhängig von der Parkhausinfrastruktur auf statische und dynamische Hindernisse durch Ausweichen, Bremsen oder Nothalt reagieren können.

Vor dem Einfahren eines Fahrzeugs in das Parkhaus übermittelt das Parkhaus-Leitsystem auf Anfrage des Fahrzeugs die Daten eines freien Zielparkplatzes an das Fahrzeug und stellt eine Karte des Parkhauses bereit. Die Karte muss vollständig oder in Teilen abrufbar sein, da davon ausgegangen werden muss, dass die Übertragungsbandbreite begrenzt ist bzw. nicht ausreichend Speicherplatz im Fahrzeug für eine vollständige Zwischenspeicherung vorhanden ist. Die abrufbaren Kartendaten sollen unabhängig vom Fahrzeugtyp sein und mindestens Informationen zum Fahrspurnetz, Lage, Geometrie und Belegung der Stellplätze, sowie Positionsdaten von markanten natürlichen oder künstlichen Referenzobjekten (Markern) für die Lokalisierung enthalten. Auf Basis der Umfeldsensor- und Eigenbewegungsdaten des Fahrzeugs, sowie der vom Parkhaus empfangenen Daten soll das Fahrzeug selbständig in der Lage sein die optimale Route und evtl. mögliche Ausweichrouten von der aktuellen Position zum zugewiesenen Stellplatz zu berechnen.

4. Implementierung

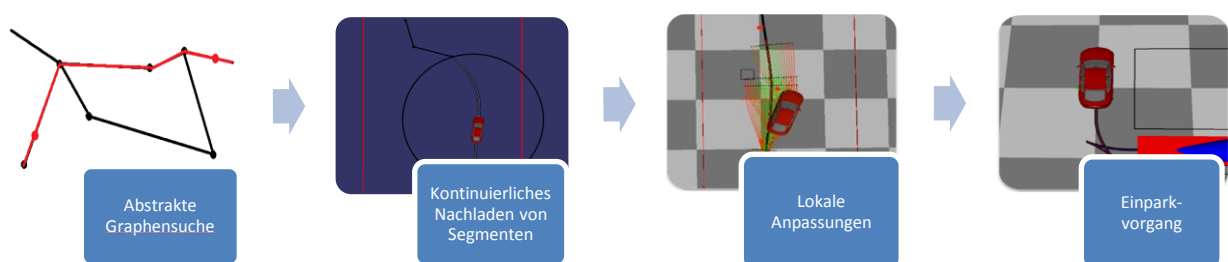


Abbildung 1: Mehrstufiges Planungskonzept

Da die Karte des Parkhauses in ihrer Gesamtheit aufgrund von fahrzeugseitigen Speicherressourcenbeschränkungen unter Umständen nicht auf das Fahrzeug geladen werden kann, wird ein mehrstufiges Planungskonzept vorgeschlagen, um die

Ressourcenbeschränkung zu umgehen. Siehe **Error! Reference source not found.**

Zunächst wird die Gesamtstrecke im Parkhaus vom Eingang bis in die Umgebung des Zielparkplatzes per A-Stern-Graphensuche geplant. Anschließend werden kontinuierlich detailliertere Daten der als nächstes zu fahrenden Streckensegmente geladen. Die darauf basierende Referenzbahn wird mit Hilfe der Informationen der fahrzeugeigenen Sensorik lokal angepasst. In der letzten Stufe wird die Bahn zum Einparken mit einem vom Parkhaus unabhängigen Algorithmus geplant. Im Folgenden werden diese Schritte näher beschrieben.

4.1 Abstrakte Graphensuche

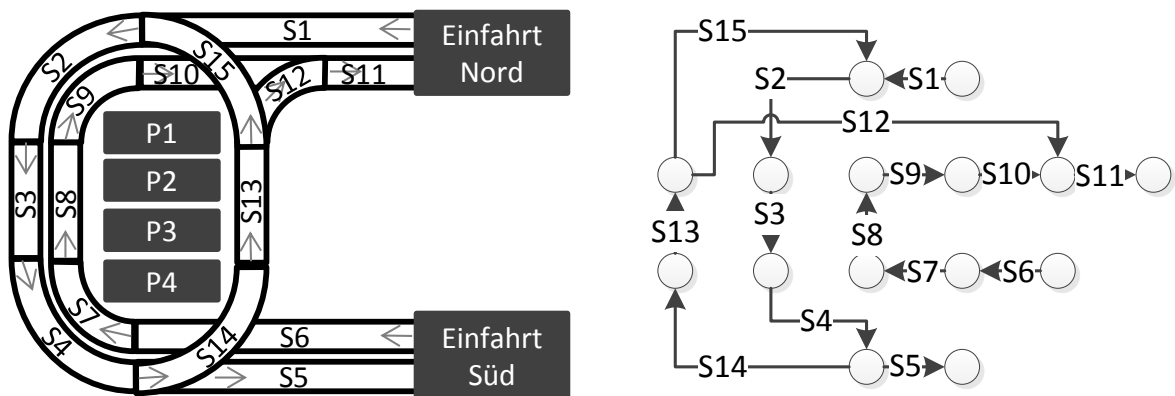


Abbildung 2: Schematische Darstellung eines Parkhauses und abstrakte Repräsentation als gerichteter Graph

Um die Transitstrecke von der Einfahrt des Parkhauses bis zum Parkplatz zu planen, benötigt der Planungsalgorithmus eine abstrakte und speichereffiziente Repräsentation des Parkhauses. Die Repräsentation muss bereits alle erforderlichen Informationen enthalten, damit der Planungsalgorithmus sich für eine Transitstrecke entscheiden kann. **Error! Reference source not found.** zeigt die schematische Darstellung eines kleinen Parkhauses. Es gibt zwei Ein- und Ausfahrten, vier Parkplätze (P1-P4), sowie 15 Spursegmente (S1-S15). Die Umgebung eines Parkplatzes kann aus mehreren Segmenten bestehen. Parkplatz P3 ist beispielsweise von den Segmenten S3, S8 und S13 aus zu erreichen. Für die Transifahrt vom Parkplatz zum Ausgang sind dadurch sowohl mehrere Startsegmente möglich, als auch mehrere Zielsegmente, da es mehr als eine Ausfahrt gibt. Die Planung der Transifahrt ist also eine Graphensuche von N Startsegmenten zu M Zielsegmenten. Als Eingabe für diese Suche ist ein Suchgraph nötig, der die Topologie der Segmente einschließlich der Länge der einzelnen Streckenabschnitte abbildet. Dies reicht als Eingabe für eine Graphensuche mit Hilfe des A-Stern-Algorithmus. Um die euklidische Distanz als Heuristik verwenden zu können, enthält der Suchgraph außerdem die

geometrische Position der Segmentenden. Alle oben beschriebenen Eingabedaten benötigen wenig Speicher bzw. Übertragungsbandbreite. Um den Speicherplatz noch weiter einzuschränken ist es möglich einen Dijkstra-Algorithmus zu verwenden und die geometrischen Informationen wegzulassen. Das Ergebnis der Graphensuche ist eine Liste von Segmentnummern.

4.2. Kontinuierliches Nachladen von Segmenten

Nach der Graphensuche lädt der Parkhauspilot kontinuierlich die detaillierteren Daten der gerade benötigten Segmente nach. Dazu gehören Meta-Daten die zur Lokalisierung im Parkhaus dienen, sowie die für das Planungsmodul benötigte geometrische Form der Segmente und die Karte mit bekannten statischen Hindernissen. Die Geometrie der Segmente ist abschnittsweise als parametrisierte Polynome 5. Ordnung hinterlegt und sind wie folgt definiert:

$$P(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = \sum_{i=0}^5 \begin{pmatrix} c_{x,i} \\ c_{y,i} \end{pmatrix} s^i, \quad s \in [0 \dots L]$$

wobei L die Länge des Segments und $c_{x,i}, c_{y,i}$ die Koeffizienten des Polynoms beschreiben.

Zum Erstellen der Polynome wird ein Karteneditor verwendet. Durch die quintischen Polynome kann der Editor garantieren, dass die Krümmung der Referenzbahn auch an den Stützstellen stetig ist. Dadurch ist die Referenzbahn komfortabel abfahrbar.

Um für die unterlagerte lokale Planungsschicht eine von den eingesetzten Bahnplanungsalgorithmen unabhängige Schnittstelle zu erreichen werden die Polynome bei der Zusammenführung in gesampelte und linear interpolierte Kurven umgewandelt. Dabei werden neben der geometrischen Position $P(s)$ auch die Orientierung der Tangente θ und die Krümmung σ berechnet um numerische Ungenauigkeiten und stetige Neuberechnung zu vermeiden:

$$\theta(s) = \text{atan2}(y'(s), x'(s)) \quad \text{mit} \quad \begin{pmatrix} x'(s) \\ y'(s) \end{pmatrix} = \sum_{i=1}^5 i \begin{pmatrix} c_{x,i} \\ c_{y,i} \end{pmatrix} s^{i-1}$$

$$\sigma(s) = \frac{x'y'' - y'x''}{\sqrt{x'+y'}} \quad \text{mit} \quad \begin{pmatrix} x''(s) \\ y''(s) \end{pmatrix} = \sum_{i=2}^5 i(i-1) \begin{pmatrix} c_{x,i} \\ c_{y,i} \end{pmatrix} s^{i-2}$$

4.3. Ein- und Ausparkvorgang

Je nach gewünschtem Manöver wird eine Ein- oder Ausparkbahn mit der einer Transifahrt verknüpft. Die kombinierte Ablauf wird durch eine Ablaufsteuerung angestoßen. Im letzten Segment der Transifahrt in das Parkhaus hinein bringt die Ablaufsteuerung das Fahrzeug in der Umgebung des Zielparkplatzes in den Stillstand um den Einparkvorgang zu starten. Der

Einparkvorgang kann von einem vom Transitmodul unabhängigen Planungsmodul ausgeführt werden. Dadurch kann der Funktionsverantwortliche sowohl die bereits in Serienfahrzeugen verbauten Einparkassistenten als Vergleichsbasis verwenden, als auch bestehende Planungsmodul für z.B. die Freiraumplanung wieder verwenden. Im hier vorgestellten Projekt verwenden wir eine Implementierung eines auf CC-Steer [4] basierenden A-Stern-Planungsalgorithmus. Die Kostenfunktion für den Planer wurde dazu so angepasst, dass das Fahrzeug mit möglichst wenigen Zügen und minimaler Wegstrecke in die Parklücke einfährt.

Das Ausparken funktioniert umgekehrt. Die Transittfahrt-Planung legt zuerst die Transitsegmente fest und im Anschluss versucht der A-Stern-Planer eine Bahn vom Parkplatz zu einem günstigen Punkt auf dem nahlegenden Transitsegment zu finden. Wenn ein bereits vorhandener Auspark-Assistent verwendet wird, kann das Fahrzeug zunächst selbständig Ausparken, und dann die Transittfahrt fortsetzen.

4.4. Lokale Anpassungen

Der lokale Planer verwendet einen Algorithmus auf Basis des Ansatzes von Moritz Werling [3] um Trajektorien basierend auf einer Referenzbahn zu generieren. Mit Hilfe der so berechneten Trajektorien ist das Fahrzeug in der Lage, sowohl statischen Hindernissen als auch dynamischen Hindernissen auszuweichen, bzw. vor ihnen stehen zu bleiben. Häufig auftretende statische Hindernisse im Parkhaus sind ungünstig geparkte Fahrzeuge oder fehlerhaft kartierte Säulen. Dynamische Hindernisse können andere Fahrzeuge oder Fußgänger sein.

Verschiedene Arten von Trajektorien werden nach unterschiedlichen Zielsetzungen gebaut:

Fahr-Trajektorien sind dafür zuständig, dass das Fahrzeug der Referenzbahn mit einer bestimmten Geschwindigkeit folgt.

Anhalte-Trajektorien sind für eine genaue Zielbremsung parametrisiert.

Sicherheits-Trajektorien bringen das Fahrzeug jederzeit zum Stillstand, falls keine gültigen Trajektorien gefunden wurden.

Zwischen den Trajektorienarten kann in jedem Rechenzyklus umgeschaltet werden. Ein Ausweichen von Hindernissen wird durch Zulassen von Querabweichungen der Fahrtrajektorie realisiert. Durch die Sicherheitstrajektorien wird sichergestellt, dass vor statischen und dynamischen Hindernissen abgebremst wird, wenn diesen nicht ausgewichen werden kann.

5. Ergebnisse

Die oben aufgeführten Algorithmen wurden als mehrere Softwarekomponenten (Filter) für ADTF in C/C++ implementiert und in einer Simulationsumgebung getestet. Die Simulation läuft ebenfalls im ADTF Framework, sie bedient sich dabei Modulen, welche eine Umgebungserfassung, eine vordefinierte Parkhaus-Karte sowie die Ablauf-Logik repräsentieren.

Eine vereinfachte Dynamiksimulation und Fahrzeugansteuerung kümmern sich um die Ausführung der geplanten Trajektorie. Einige Screenshots der Simulation werden in Abbildung 3 dargestellt.

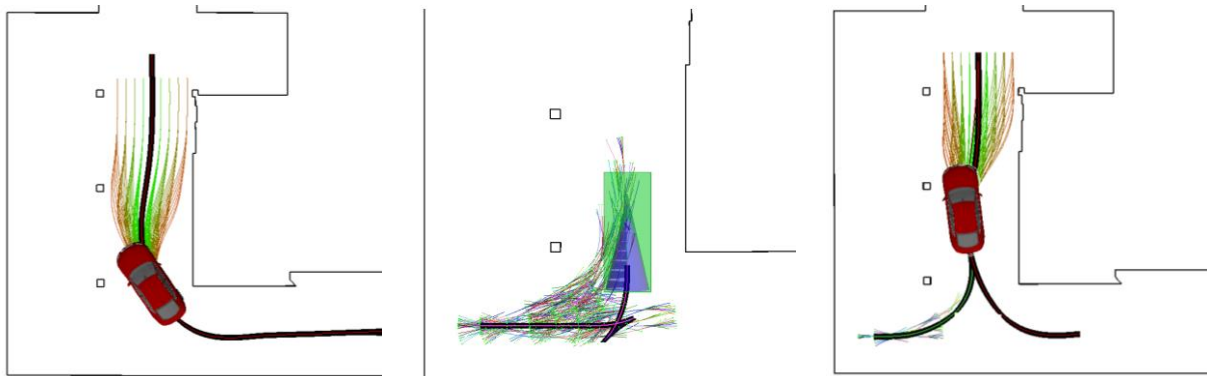


Abbildung 3: Simulationsergebnisse: eine Transitfahrt mit abstrakter Graphensuche und Nachladen der RoadGraph Segmente (links); das Ein- oder Ausparken mit dem A-Stern Algorithmus (middle); das kombinierte Manöver vom Ausparken und Transit (rechts).

Die dünnen schwarzen Polygonlinien sind die vom Fahrzeug erkannten Hindernisse in der Umgebung. Die dicken Polynomlinien sind die geplanten Referenzbahnen, die von unterschiedlichen Planungsalgorithmen berechnet und zusammengefügt werden können. Die generierten Trajektorien werden als Trajektorienfächer um die Referenzbahn herum dargestellt. Die Kosten der Trajektorien werden durch den Farbverlauf/Graustufen dargestellt. Mittels des Trajektorienfächers hat das Fahrzeug die Möglichkeit seine lokale Bewegung an Hindernisse anzupassen und gleichzeitig noch der Referenzbahn zu folgen. In Abbildung 3 Mitte sieht man den Suchgraphen der vom A-Stern Planer generiert wird und die Visualisierung der günstigsten gefundene Verbindung als Referenzbahn in Form einer dicke schwarze Linie.

In der Simulation hat dieses mehrstufige Planungskonzept gezeigt, dass es das gesamte Manöver im Parkhaus erfolgreich planen und durchführen kann. Durch den immer feiner werdenden Abstraktionsgrad spielt die Größe des Parkhauses keine Rolle (nur bei der anfänglichen Graphensuche) und es können einfachere Algorithmen für jede Stufe

verwendet werden. Dadurch können komplexere wie zum Beispiel mehrstöckige Parkhäuser adressiert werden. Gleichzeitig bietet das Konzept die Flexibilität, spezielle Planer für spezifische Problemstellungen zu nutzen. Dies entspricht einer guten Wiederverwendbarkeit in der Entwicklung von FAS- oder pilotierten Fahrfunktionen.

6. Zusammenfassung und Ausblick

In diesem Beitrag haben wir unser mehrstufiges Konzept für Planungen in einem Parkhaus vorgestellt und in einer Simulationsumgebung gezeigt wie ein funktionierendes Gesamtsystem aussehen kann. Durch die Aufteilung ist der Kommunikations-, Speicher- und Planungsaufwand auf wesentliche Teile des Parkhauses beschränkt, die durch eine Graphensuche identifiziert werden. Dadurch und durch die Möglichkeit der sequentiellen Nachladung von Detail-Informationen ist diese Lösung skalierbar im Hinblick auf die Größe und Komplexität eines Parkhauses.

Aktuell wird das Konzept an einem realen Parkhaus evaluiert, vor allem im Hinblick auf lokale Trajektorienanpassungen bzgl. anderer Verkehrsteilnehmer und nicht kartierten statischen Hindernissen. Die kooperative Planung für den simultanen Betrieb mehrerer autonomer Fahrzeuge auf Basis eines dezentralen Planungssystems stellt eine weitere Forschungsfrage für die Zukunft dar.

Referenzen

- [1] Rainer Kümmerle, Dirk Hänel, Dmitri Dolgov, Sebastian Thrun, and Wolfram Burgard, „Autonomous Driving in a Multi-level Parking Structure,” IEEE International Conference on Robotics and Automation, 2009, pp. 3395–3400.
- [2] Paul Furgale, Ulrich Schwesinger, Martin Rufli, Wojciech Derendarz, et. Al. „Toward Automated Driving in Cities using Close-to-Market Sensors, an Overview of the V-Charge Project“, IEEE Intelligent Vehicles Symposium (IV), 2013, pp.809-816
- [3] Moritz Werling, Sören Kammel, Julius Ziegler, and Lutz Gröll, “Optimal Trajectories for Time-critical Street Scenarios Using Discretized Terminal Manifolds,” The International Journal of Robotics Research, vol. 31, no. 3, pp. 346–359, Mar. 2012.
- [4] Thierry Fraichard and Alexis Scheuer, “From Reeds and Shepp’s to Continuous-Curvature Paths,” IEEE Transactions on Robotics, vol. 20, no. 6, pp. 1025–1035, Dec. 2004.
- [5] C. Loper, C. Brunken, G. Thomaidis, S. Lapoehn, P. P. Fouopi, H. Mosebach, and F. Koster, “Automated valet parking as part of an integrated travel assistance,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 2341–2348.