
PARKRAUM OPTIMAL NUTZEN MIT AUTONOM KOOPERIERENDEN, VERNETZTEN FAHRZEUGEN

Februar 2021

Lukas Haverbeck, Frederik Schittny, Lukas Grave
Unter Betreuung von Hendrik Büdding
Annette-von-Droste-Hülshoff-Gymnasium Münster

Kurzfassung

Durch ineffizientes Parken bleiben große Teile urbanen Parkraums ungenutzt, sodass sich Fahrzeiten und -strecken bei der Suche nach einem Parkplatz erhöhen. Vernetzte, autonom interagierende Fahrzeuge können zur Lösung dieses Problems beitragen. Dazu setzen sie eine möglichst effiziente Anordnung der Fahrzeuge selbstständig um, indem sie ihr Verhalten untereinander koordinieren, zueinander aufrücken und so den ungenutzten Platz zwischen den Fahrzeugen verringern. An der Optimierung öffentlicher Flächen in Innenstädten und der Minimierung externer Kosten, z. B. in Form von Emissionen, besteht ein gesellschaftliches und wirtschaftliches Interesse. Dazu ist es unter anderem notwendig, Parkflächen effizienter zu nutzen. Die Ausgestaltung der Arbeit beschäftigt sich primär mit der Konzeptionierung und Implementierung eines dezentralen Kommunikationsmodells zum Austausch von Intentionen und Umgebungsinformationen der Fahrzeuge zur Realisierung der notwendigen Kommunikationsabläufe.

Inhaltsverzeichnis

1	Einleitung	4
2	Begriffsklärung	5
3	Theoretische Modellierung	5
3.1	Problem- und Lösungsspezifikation	5
	Problem 1: Lineares Wachstum der ungenutzten Fläche auf Parkstreifen	5
	Problem 2: Suboptimale Abstände zwischen den Akteuren	5
	Lösung: Dynamisches Parken durch ein kooperatives, intelligentes Parksystem (C-IPS) . .	6
3.2	Beschreibung des Zustands des Parkstreifens	6
3.3	V2V Kommunikation zur Ermittlung der Formation	7
	V2V Kommunikation für C-ITS: DSRC / ITS G5 vs. C-V2X	8
	Bestimmung und Synchronisation einer Formation	8
3.4	Steuerung kooperativer Abläufe	9
	Priorisierung simultaner Intentionen verschiedener Akteuren derselben Formation	10
	Kooperation beim Verlassen der Formation	10
	Kooperation bei der dynamischen Minimierung ungenutzter Abstände	10
4	Implementierung von Testfahrzeugen	11
4.1	Entwurf und Konstruktion von Modellfahrzeugen	11
	Antrieb	11
	Ansteuerung des Motors	12
	Sensorik und eindeutige Identifizierung	13
4.2	Implementation der C-IPS Software	13
	V2V Kommunikation	13
	Formation	14
	Kooperation	14
5	Ergebnis	14
5.1	Einschätzung der potenziellen Effizienzsteigerung	15
5.2	Vergleich von Modell und Testfahrzeugen	15
6	Ergebnisdiskussion	16
6.1	Nutzen von C-IPS	16
6.2	Realisierbarkeit von C-IPS	16
6.3	Ausblick	17
7	Reflexion	17
8	Unterstützungsleistungen und Danksagung	18
A	Informationen zur Implementation	19
B	Verwendete Bauteile	19
D	Quellen- und Literaturverzeichnis	20

1 Einleitung

Unzureichend verfügbarer Parkraum stellt in fast allen deutschen Großstädten ein signifikantes Verkehrsproblem dar. Eine Studie des privaten Mobilitätsunternehmens INRIX aus dem Jahr 2017 zeigt, dass der Mangel an Parkraum in Deutschland jährlich volkswirtschaftliche Kosten in Höhe von 40,4 Milliarden Euro verursacht [CP17], die sich vor allem auf den erhöhten Kraftstoffverbrauch, die bei der Parkplatzsuche verschwendete Zeit und den zusätzlichen Ausstoß von Abgasen bei der Parkplatzsuche zurückführen lassen [INR17]. Eine Untersuchung der Parkraumnutzung in London, deren Ergebnisse sich aber in ihren Grundaussagen auch auf deutsche Großstädte übertragen lassen, zeigt außerdem, dass ein Großteil aller Parkvorgänge in einem geschäftlichen oder kommerziellen Zusammenhang steht [BL12, S. 34] und dass die Anreise zur Arbeit der häufigste Grund für das Suchen eines Parkplatzes ist [BL12, S. VII], weshalb mangelnder Parkraum zu Stoßzeiten zur Überlastung des Personen- und Güterverkehrs beiträgt.

“Der Parksuchverkehr macht allein 30 bis 40 Prozent des innenstädtischen [sic] Gesamtverkehrs aus. Dabei braucht ein Autofahrer durchschnittlich zehn Minuten für die Parkplatzsuche und legt dabei 4,5 Kilometer zurück – eine vermeidbare Belastung für Mensch und Umwelt.” [SH20]

In den Innenstädten wächst unter anderem deswegen das gesellschaftliche Interesse an der Reduzierung der Präsenz von PKWs im Stadtbild [WK17]. Neben der Ersparnis externer Kosten durch die Reduzierung des Verkehrsaufkommens würde eine effizientere Nutzung bestehender Parkkapazitäten in Städten die Möglichkeit bieten, frei werdende Flächen anderweitig zu nutzen. Dabei bieten vor allem Parkstreifen Optimierungspotenzial, da mit zunehmender Besiedlungsdichte der Anteil der auf der Straße geparkten Fahrzeuge zunimmt [BL12, S. 28] und sich in Deutschland ca. 70% der Parkplätze am Straßenrand befinden [SH20]. Konventionelles Parken auf Parkstreifen birgt vornehmlich zwei Probleme:

1. Jedes parkende Fahrzeug hält nach vorne und hinten mindestens so viel Abstand, wie es selbst benötigt, um ausparken zu können. Dadurch wächst der nicht unmittelbar zum Parken genutzte Platz auf einem Parkstreifen linear mit der Anzahl der Fahrzeuge. Davon wird aber nur ein deutlich geringerer Betrag tatsächlich benötigt.
2. Fahrzeuge parken häufig in zu großen Parklücken, in dem Sinne, dass sie mehr Abstand halten als sie zum Ausparken benötigen.

Simultan zur zunehmenden Urbanisierung und der damit einhergehenden Verschärfung der Parksituation in vielen Städten lässt sich in der Automobilindustrie ein Trend hin zu autonomen Fahrsystemen und zu einer größeren Vernetztheit von Fahrzeugen feststellen. Da die Vernetzung partiell autonom agierender Fahrzeuge großes Potenzial für die sicherere und effizientere Gestaltung des Verkehrs bietet, werden kooperative, intelligente Transportsysteme (kurz C-ITS für “Cooperative, Intelligent Transport Systems”) intensiv erforscht [K S17] und in ersten Bereichen bereits angewendet [Con20]. Unser Ziel ist es, ein System zu entwickeln, das die effiziente Nutzung von Parkstreifen ermöglicht, indem es Fortschritte bei der Vernetzung von Fahrzeugen ausnutzt.

In dieser Arbeit formalisieren wir Ineffizienzen bei der Nutzung von Parkstreifen und entwickeln darauf basierend ein Modell eines kooperativen, intelligenten Parksystems (kurz C-IPS für “Cooperative Intelligent Parking System”) als Anwendung autonomer und vernetzter Mobilität, das eine wesentlich effizientere Nutzung von Parkstreifen ermöglicht. Um die Umsetzbarkeit des theoretischen Modells zu untersuchen, konstruieren wir Testfahrzeuge, für die wir die C-IPS Software implementieren. Wir evaluieren außerdem das Potenzial von C-IPS im Hinblick auf technische Realisierbarkeit unter anderem im Bezug auf infrastrukturelle Anforderungen, Skalierbarkeit und Kosten.

2 Begriffsklärung

Da sich diese Arbeit mit der effizienteren Nutzung von Parkstreifen beschäftigt, bezeichnet der Ausdruck “Parkplatz” im Folgenden stets einen Platz auf einem Parkstreifen, auf dem ein Fahrzeug steht oder stehen könnte. Dabei wird angenommen, dass alle auf einem Parkstreifen parkenden Fahrzeuge gleich ausgerichtet sind. Ein “Akteur” ist ein auf einem Parkstreifen stehendes oder ausparkendes Fahrzeug, nicht jedoch ein einparkendes Fahrzeug. Ein einparkendes Fahrzeug wird zum Akteur, sobald der Einparkvorgang abgeschlossen ist.

3 Theoretische Modellierung

In diesem Abschnitt wird C-IPS als System zur effizienten Nutzung von Parkstreifen etabliert. Dafür werden formale Ausdrücke bestimmt, die das Maß der Ineffizienz beim Parken auf Parkstreifen beschreiben. Die in der Einleitung angedeuteten Teilprobleme werden dabei zunächst separat betrachtet und dann zusammengeführt. Aus der Problemspezifikation leiten wir ein Lösungskonzept ab, indem wir charakteristische Systemanforderungen zur Minimierung der zuvor etablierten Problemgröße aufstellen. Der Rest des Kapitels befasst sich damit, dieses Lösungskonzept zu konkretisieren. Dafür wird der Zustands eines Parkstreifens, die Kommunikation zwischen Akteuren und die autonome Kooperation von Akteuren als grundlegende Bestandteile von C-IPS formalisiert.

3.1 Problem- und Lösungsspezifikation

Problem 1: Lineares Wachstum der ungenutzten Fläche auf Parkstreifen

Man betrachte einen Parkstreifen mit den Akteuren $a_1, a_2, \dots, a_n$, wobei $n \in \mathbb{N}, n \geq 3$ die Anzahl der Akteure auf dem Parkstreifen sei. Für alle $i \in \mathbb{N}, i < n$ stehe a_i vor a_{i+1} . Damit alle Akteure genügend Platz zwischen einander haben, um jederzeit ausparken zu können, halten sie feste Abstände $d_j, j \in \mathbb{N}, j \leq n+1$, wobei d_j für $2 \leq j \leq n$ der Abstand zwischen a_j und a_{j-1} sei, d_1 der Abstand vor a_1 sei und d_{n+1} der Abstand hinter a_n sei. Vereinfachend wird angenommen, dass $d_1, \dots, d_{n+1}$ genau so groß sind, dass jeder Akteur problemlos ausparken kann. Es ergibt sich $D_{ges} := \sum_{k=1}^{n+1} d_k$ als nötiger Gesamtabstand, den die Akteure auf dem Parkstreifen insgesamt nach vorne und hinten halten. Sei ferner $d := \frac{D_{ges}}{n+1}$ der durchschnittlich nötige Abstand der Akteure nach vorne bzw. hinten.

Man bemerke, dass sich an der Definition von D_{ges} unmittelbar ablesen lässt, dass der nötige Gesamtabstand annähernd proportional zu n wächst, da aufgrund ähnlicher Fahrzeugbeschaffenheiten $\forall l \in \mathbb{N}, l \leq n+1 : d_l \approx d$ angenommen werden kann. Folglich wächst die nicht unmittelbar zum Parken genutzte oder verfügbare Gesamtstrecke auf einem Parkstreifen linear mit der Anzahl der Akteure. Theoretisch ermöglicht dies allen Akteuren, gleichzeitig auszuparken (wenngleich dafür ein weitgehend synchrones Fahrverhalten nötig wäre). Jedoch kommt es in der Realität nur selten vor, dass auf einem Parkstreifen mehrere Fahrzeuge gleichzeitig ausparken. Geht man davon aus, dass zu jedem Zeitpunkt maximal ein beliebiger, also insbesondere der Akteur a_{max} , der am meisten Platz zum Ausparken benötigt, den Parkstreifen verlässt, muss zu jedem Zeitpunkt höchstens ein konstanter Abstand δ_{max} um das ausparkende Fahrzeug herum gehalten werden. Zwischen den nicht ausparkenden Akteuren muss dann lediglich ein fester Sicherheitsabstand λ mit $\lambda \ll d$ bestehen, der sehr gering angenommen werden kann, da dieser lediglich Zusammenstöße der Fahrzeuge verhindern soll und nicht zum Ausparken genutzt wird. Der gesamte Sicherheitsabstand auf dem Parkstreifen sei $\Lambda := n \cdot \lambda$.

Eine Ineffizienz bei der herkömmlichen Nutzung von Parkstreifen besteht also darin, dass der nicht unmittelbar zum Parken genutzte Gesamtabstand, den die Akteure halten, um ausparken zu können, linear mit der Anzahl der Akteure wächst, obwohl unter der Voraussetzung, dass stets maximal ein Akteur auf einmal ausparkt, lediglich ein konstanter Abstand zum Ausparken benötigt wird. Diese Ineffizienz lässt sich näherungsweise als $\Delta_1 = D_{ges} - (\delta_{max} + \Lambda) = (n+1) \cdot d - \delta_{max} - n \cdot \lambda$ ausdrücken. Wegen $\lambda \ll d$ nimmt Δ_1 auf Parkstreifen, auf denen viele Fahrzeuge hintereinander parken, signifikante Beträge an.

Problem 2: Suboptimale Abstände zwischen den Akteuren

In dem vorangegangenen Abschnitt wurde $D_{ges} := \sum_{k=1}^{n+1} d_k$ als Quantifizierung des auf einem Parkstreifen nicht unmittelbar für parkende Fahrzeuge genutzten oder verfügbaren Gesamtabstands etabliert.

Dabei wurden die Abstände $d_1, \dots, d_{n+1}$ jedoch idealisiert als minimal in dem Sinne angenommen, dass alle Abstände $d_k, d_{k+1}, k \in \mathbb{N}, k \leq n$ gerade ausreichend dafür sind, dass a_k den Parkstreifen problemlos verlassen kann. Wie Abbildung 1 aber zeigt, ist diese Annahme in der Realität nicht immer zutreffend, da häufig Fahrzeuge in Parklücken parken, in denen zuvor längere Fahrzeuge standen. Vor und hinter jedem Akteur ergeben sich demnach häufig Abstände, die nicht zum Ausparken benötigt werden.



Abbildung 1: Exemplarische Darstellung der ineffizienten Nutzung eines Parkstreifens

Folglich gibt es auf einem Parkstreifen neben $d_1, \dots, d_{n+1}$ weitere Abstände $d'_1, \dots, d'_{n+1}$, wobei $d'_j, j \in \mathbb{N}, 2 \leq j \leq n$ der Abstand zwischen a_j und a_{j-1} sei, den weder a_j noch a_{j-1} zum Ausparken benötigt, d'_1 der Abstand vor a_1 sei, den a_1 nicht zum Ausparken benötigt und d'_{n+1} der Abstand hinter a_n sei, den a_n nicht zum Ausparken benötigt. Eine weitere Ineffizienz bei der herkömmlichen Nutzung von Parkstreifen ergibt sich also dadurch, dass die Abstände, die die Akteure auf einem Parkstreifen halten, größer sein können als zum Ausparken benötigt. Diese Ineffizienz ergibt sich als $\Delta_2 := \sum_{k=1}^{n+1} d'_k$.

Im Folgenden bezeichnet $\Delta := \Delta_1 + \Delta_2$ die gesamte, in dieser Arbeit behandelte, Ineffizienz bei der Nutzung von Parkstreifen.

Lösung: Dynamisches Parken durch ein kooperatives, intelligentes Parksystem (C-IPS)

Erklärtes Ziel des Modells ist es, eine effiziente Anordnung der Fahrzeuge auf einem Parkstreifen zu erreichen, durch die $\Delta = \Delta_1 + \Delta_2 = 0$ wird. Dafür muss $D_{ges} = \delta_{max} + \Lambda$ sein. Das liegt daran, dass der Gesamtabstand dann gerade so groß ist, dass Sicherheitsabstände gehalten werden können und zusätzlich auf den gesamten Parkstreifen verteilt ausreichend Platz zur Verfügung steht, dass jederzeit genau ein beliebiger Akteur ausparken kann. Um diese Anordnung umzusetzen, entwickeln wir ein kooperatives, intelligentes Parksystem (C-IPS), in dem die Akteure miteinander kommunizieren und autonom miteinander interagieren. Dadurch ist es den Fahrzeugen möglich, Informationen über sich und ihre Anordnung auszutauschen und so δ_{max} zu bestimmen. Außerdem kann ein ausparkendes Fahrzeug auf diese Weise seine Intention auszuparken kommunizieren, sodass die anderen Akteure um den ausparkenden Akteur herum den benötigten Abstand $\delta \leq \delta_{max}$ schaffen können, indem sie autonom zueinander aufrücken. In diesem System gilt $\Delta_1 = \delta_{max} + \Lambda - (\delta_{max} + \Lambda) = 0$. Der Abstand $D_{ges} = \delta_{max} + \Lambda$ muss gehalten werden, damit auch wenn a_{max} ausparkt, genügend Platz dafür zur Verfügung steht. Folglich wird in diesem System nicht mehr Abstand gehalten als unbedingt nötig, weshalb auch $\Delta_2 = 0$ ist. Wegen $\Delta_1 = \Delta_2 = 0$ wird damit die gesamte Ineffizienz $\Delta = 0$.

3.2 Beschreibung des Zustands des Parkstreifens

Da nicht angenommen werden kann, dass jedes Fahrzeug auf einem Parkstreifen C-IPS implementiert, bildet eine zusammenhängende Reihe von C-IPS Fahrzeugen eine Formation, deren Akteure autonom miteinander interagieren. Formal beschreibt eine Formation den Zustand eines Parkstreifens, den jeder Akteur kennen muss, um seine eigene Position sowie zu einem beliebigen, ausparkenden Akteur dessen Position und den zum Ausparken benötigten Abstand ermitteln zu können.

Eine Formation $F := (a_1, \dots, a_m), m \in \mathbb{N}, m \geq 3$ sei gemäß der obigen Anforderungen ein m -Tupel von Akteuren, in dem jeder Akteur $a_x = (\delta_x, id_x)$ durch ein 2-Tupel modelliert wird, das aus seinem zum Ausparken benötigten Abstand δ_x und seinem eindeutigen Bezeichner id_x besteht. Jeder

Akteur a_x implementiert C-IPS und es gilt entweder $a_x = a_1$ oder a_x steht hinter a_{x-1} , sodass die Reihenfolge der Akteure in F der Reihenfolge der Akteure auf dem Parkstreifen entspricht. Im Folgenden werden ausschließlich Formationen betrachtet, bei denen m unter den zuvor genannten Einschränkungen maximal ist. Dabei kann es auf einem Parkstreifen mehrere, voneinander unabhängige Formationen geben, was z. B. dann der Fall ist, wenn mehrere zusammenhängende Reihen aus C-IPS Fahrzeugen von nicht ausgerüsteten Fahrzeugen unterbrochen werden. Die Möglichkeit, mehrere Formationen auf einem Parkstreifen zu bilden, hat den Vorteil, dass C-IPS funktioniert, sobald mindestens 3 C-IPS Fahrzeuge hintereinander stehen.

Wichtig ist zu bemerken, dass eine Formation eine Momentaufnahme des Zustands eines Parkstreifens darstellt. Daher muss die Formation z. B. dann angepasst werden, wenn Fahrzeuge ein- oder ausparken.

Vor der ersten Ermittlung der Formation kennt jeder Akteur lediglich Informationen über sich selbst und sein unmittelbar durch Sensoren erfassbares Umfeld, aus denen sich die Formation nicht direkt ableiten lässt. Eine Formation lässt sich aber aus den Informationen aller ihrer Akteure zusammensetzen. Daher ist es notwendig, dass die Akteure miteinander kommunizieren, um so ihre Informationen untereinander austauschen zu können.

3.3 V2V Kommunikation zur Ermittlung der Formation

Kooperative, intelligente Transportsysteme (C-ITS) sind Anwendungen von vernetzten Fahrzeugen (kurz V2V für "Vehicle to Vehicle"), die auf einem Frequenzband von 5,9 GHz miteinander kommunizieren und so die Verkehrssicherheit und -effizienz erhöhen sollen [K S17, S. 1f.]. C-ITS ermöglicht Fahrzeugen das Austauschen von Informationen sowie insbesondere Intentionen und somit das Abstimmen ihres Verhaltens untereinander ("[...] The V2X sensor can [...] receive information and the intention of other objects, and the vehicle can then adapt its behavior to this received data." [K S17, S. 90]). Neben der Erhöhung der Verkehrssicherheit ist auch die effizientere Nutzung des Parkraums Gegenstand der Forschung zur Verkehrsvernetzung [NP19, S. 70173]. Das von uns entwickelte kooperative, intelligente Parksysteem C-IPS ist eine Ausprägung von C-ITS mit Fokus auf die effiziente Nutzung von Parkstreifen.

Bereits seit den 1990er Jahren sind C-ITS und V2V Kommunikation Gegenstand grundlegender sowie angewandter Forschung, die aufgrund ihres Potenzials zur Erhöhung von Verkehrssicherheit und -effizienz, unter anderem von der Europäischen Kommission [K S17, S. 95] und Herstellern aus der Wirtschaft [Con20, S. 2] unterstützt und vorangetrieben wird. Schwerpunktmäßig befasst sich die Forschung unter anderem mit der Entwicklung marktfähiger Anwendungen von C-ITS [F A11] und unterschiedlichen Möglichkeiten zur Implementierung von V2V Kommunikation [NP19].

Neben der Kommunikation zwischen Fahrzeugen (V2V) involviert C-ITS unter anderem auch die Kommunikation von Fahrzeugen mit Infrastruktur (V2I). Die folgenden Anforderungen an die Kommunikation zwischen C-IPS Akteuren verdeutlichen jedoch, dass für die Implementation von C-IPS lediglich V2V Kommunikation erforderlich ist.

- Die Akteure haben die Möglichkeit, dynamische aber stabile Ad-hoc-Netzwerke zu bilden, um innerhalb einer veränderlichen Formation Informationen austauschen zu können.
- Die Kommunikation zwischen den Akteuren soll keine externe Infrastruktur erfordern, um die Umsetzung von C-IPS im Hinblick auf ökonomische Faktoren zu vereinfachen.
- Die Übertragung von Informationen erfolgt mit einer zeitlichen Verzögerung, die maximal in einer Größenordnung von 100 ms liegt, was aufgrund des geringen Payloads der Nachrichten möglich ist. Dies ist notwendig, um die Reaktionszeiten der Akteure gering zu halten.
- Um sicherzustellen, dass die gesendeten Daten von allen Akteuren auf die gleiche Weise interpretiert werden können, werden die Informationen einheitlich kodiert.
- Damit nur Informationen aus vertrauenswürdigen Quellen verarbeitet werden, authentifiziert sich jeder Absender einer Nachricht innerhalb des Netzwerks.

V2V Kommunikation für C-ITS: DSRC / ITS G5 vs. C-V2X

Es gibt zwei wesentliche Klassen von Technologien, die V2V Kommunikation implementieren. Das sind zum einen die auf WLAN basierenden Standards DSRC (kurz für “Dedicated Short Range Communications”) sowie ITS G5 (kurz für “Intelligent Transport System on 5 GHz”) und zum anderen das auf zellulärer Technologie beruhende C-V2X (kurz für “Cellular Vehicle to Everything”). Sowohl die EU als auch die USA vergeben weitgehend gleiche Spektren auf dem 5,9 GHz Frequenzband für die Erforschung und Bereitstellung von V2V Kommunikation [K S17, S. 90 und D9, S. 6], die von DSRC und ITS G5 zur direkten Kommunikation zwischen Fahrzeugen verwendet werden. C-V2X kann ebenfalls auf diesen Spektren kommunizieren, zusätzlich aber auch das Mobilfunknetz nutzen (“[...] C-V2X can operate in the 5.9 GHz band as well as in the cellular operators’ licensed carrier”, [NP19, S. 70169]).

DSRC und ITS G5 beruhen auf dem WLAN Standard IEEE¹ 802.11p [NP19, S. 70170] [J H14, S. 6], welcher eine Abwandlung des Standards IEEE 802.11a für herkömmliches WLAN ist [NP19, S. 70170]. Der speziell für Anwendungen mit hoher Mobilität entwickelte Standard IEEE 802.11p ermöglicht einen besonders schnellen Aufbau von Verbindungen in sich dynamisch verändernden Ad-hoc-Netzwerken [NP19, S. 70170] [J H14, S. 71]. Sowohl für DSRC als auch für ITS G5 existieren Normierungen für den Aufbau von Nachrichten [J H14, S. 73] [EN14, S. 11], die z. B. Informationen über Geschwindigkeit, GPS-Standort und Fahrzeuglänge enthalten können [J H14, S. 75] [EN14, S. 23]. Beide Technologien definieren Bereiche im zugeteilten Frequenzspektrum, die nicht nur für sicherheitsbezogene Dienste, sondern auch für Anwendungen zur Steigerung der Effizienz im Verkehr, wie durch C-IPS, genutzt werden können [J H14, S. 93] [TS12, S. 13]. Das von 3GPP² entwickelte, LTE basierte C-V2X ermöglicht ähnlich wie DSRC und ITS G5 direkte V2V Kommunikation ohne externe Infrastruktur und darüber hinaus Kommunikation über das Mobilfunknetz [NP19, S. 70169], was unter anderem den Anschluss der Fahrzeuge an das Internet und den Austausch von Daten über große Distanzen ermöglicht. Alle drei Technologien authentifizieren die Kommunikationsteilnehmer mithilfe öffentlicher Schlüssel und wechselnder privater Schlüssel, also asymmetrischer Verschlüsselung, die die Verarbeitung möglicherweise manipulierter Informationen verhindert [K S17, S. 91].

Die vorgestellten Möglichkeiten zur Realisierung von V2V Kommunikation ermöglichen eine stabile Kommunikation mit Latenzen von weniger als 100 ms innerhalb eines Radius von über 100 m [NP19, S. 70173] [Ass18, S. 10, 109] und werden mit Fokus auf die Verringerung von Latenzen bei hohen Geschwindigkeiten weiterentwickelt [NP19]. Sie erfüllen aber alle bereits heute die zuvor beschriebenen V2V Anforderungen für C-IPS. Welche der Technologien am besten für die Implementierung von C-IPS geeignet ist, hängt aufgrund mangelnder Interkompatibilität [NP19, S. 70181] eher davon ab, welche der Technologien in Zukunft am weitesten verbreitet sein wird. Dies ist allerdings noch nicht absehbar, da zwar C-V2X in Studien Leistungsvorteile gegenüber DSRC und ITS G5 zeigt [Ass18], dafür aber ITS G5 bereits in ersten Modellen serienmäßig verbaut wird [Ass18, S. 1].

Bestimmung und Synchronisation einer Formation

Unabhängig von der Wahl der konkreten Technologie, kann die V2V Kommunikationsschnittstelle genutzt werden, um den im Folgenden beschriebenen Algorithmus zu implementieren, der es den Akteuren ermöglicht, die Formation, in der sie sich befinden, zu bestimmen und zu synchronisieren. Als Voraussetzung wird angenommen, dass jeder C-IPS Akteur technisch dazu in der Lage ist, das vor ihm parkende Fahrzeug zu erkennen und eindeutig zu identifizieren, falls es sich ebenfalls um einen C-IPS Akteur handelt.

In der ersten Phase des Algorithmus wird die Formation $F_m := (a_1, \dots, a_m)$ bestimmt, indem sie vom ersten Akteur a_1 bis zum letzten Akteur a_m sukzessive aufgebaut wird. Dafür versucht jeder Akteur, einen C-IPS Akteur vor sich zu identifizieren, was nur im Falle von a_1 nicht möglich ist. a_1 bestimmt sich auf diese Weise eindeutig als ersten Akteur der Formation und konstruiert die Formation $F_1 := (a_1)$, in der nur a_1 selbst enthalten ist. Über die V2V Schnittstelle sendet a_1 die unvollständige Formation F_1 an alle C-IPS Akteure in der Umgebung. Jeder Akteur a_x überprüft beim Erhalt einer unvollständigen Formation F_k , in der er nicht selbst enthalten ist, ob es sich beim letzten Akteur der Formation um seinen Vordermann a_{x-1} handelt. Nur dann reagiert der Akteur a_x , indem er gegenüber

¹ Das Institute of Electrical and Electronics Engineers (IEEE) befasst sich hauptsächlich mit der Standardisierung von Technik.

² Das 3rd Generation Partnership Project (3GPP) befasst sich mit der Standardisierung von Mobilfunk.

a_k den Erhalt der Formation F_k bestätigt und die unvollständige Formation $F_{k+1} = F_x := (a_1, \dots, a_x)$ weitersendet. Im Beispiel von F_1 reagiert also ausschließlich der Akteur a_2 auf F_1 , indem er den Erhalt der Formation bestätigt und $F_2 := (a_1, a_2)$ an die anderen Akteure weitersendet. Nach diesem Prinzip wird die Formation nach und nach zusammengesetzt. Da der letzte Akteur a_m nach Versenden der Formation F_m innerhalb eines, im Hinblick auf die Latenzen im Netzwerk angemessenen, Zeitintervalls keine Bestätigung erhält, steht a_m als letzter Akteur der Formation fest. In diesem Sinne ist F_m nach m Schritten als vollständige Formation bestimmt.

Da zu diesem Zeitpunkt lediglich a_m über F_m verfügt, wird in der zweiten Phase des Algorithmus F_m unter den Akteuren $a_1, \dots, a_m$ als vollständige Formation synchronisiert. a_m sendet F_m als vollständige Formation an die C-IPS Akteure in der Umgebung. Auf den Erhalt der vollständigen Formation F_m , reagiert ein Akteur a_x nur dann, wenn F_m von a_{x+1} gesendet wurde, indem er F_m validiert. F_m wird von a_x genau dann als valide betrachtet, wenn a_x in F_m enthalten ist und F_m aktueller ist als die aktuelle Formation von a_x . Ist F_m valide, übernimmt der Akteur a_x F_m als seine aktuelle Formation und sendet außerdem F_m als vollständige Formation weiter, sodass nach und nach alle Akteure $a_1, \dots, a_m$ F_m als aktuelle und vollständige Formation erhalten.

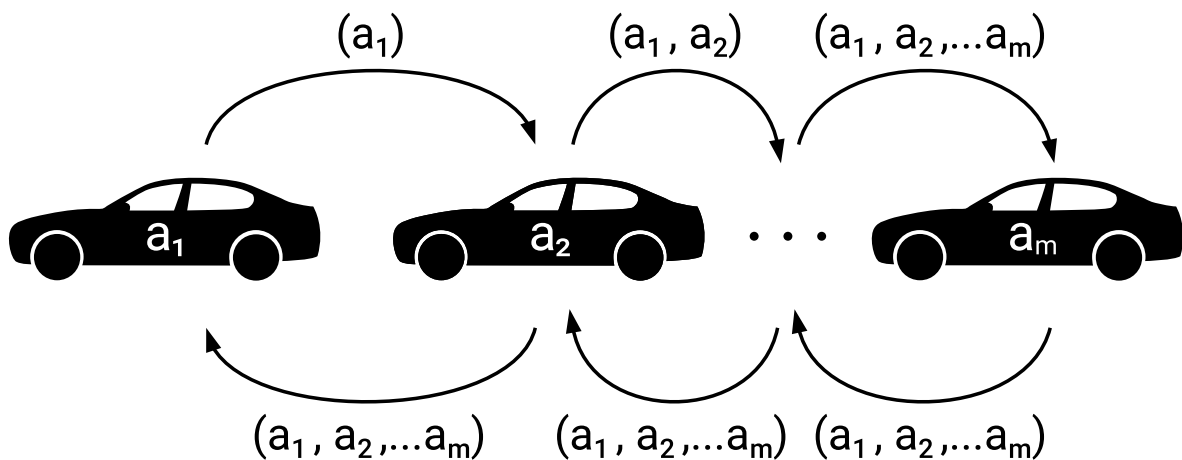


Abbildung 2: Visuelle Darstellung der Synchronisation einer Formation

Da sowohl bei der Bestimmung als auch bei der Synchronisation der Formation ausschließlich solche Akteure aufeinander reagieren, die in einer Formation unmittelbar hintereinander stehen, kommt es zu keinen Konflikten im Falle mehrerer, eigenständiger Formationen in näherer Umgebung. Damit die Akteure stets auf ihre aktuelle Formation zugreifen können, muss dieser Algorithmus immer dann ausgeführt werden, wenn sich die Formation auf dem Parkstreifen verändert, was in der Regel dann der Fall ist, wenn Fahrzeuge ein- oder ausparken.

3.4 Steuerung kooperativer Abläufe

C-IPS sieht vor, dass zu jedem Zeitpunkt auf eine vollständige Formation verteilt genau so viel Abstand gehalten wird, dass ein beliebiges Fahrzeug ausparken kann. Das bedeutet allerdings, dass sich um ein Fahrzeug herum nicht unbedingt so viel Platz befindet, dass es sofort ausparken kann. C-IPS löst dieses Problem durch autonome Interaktion der Akteure. Immer wenn ein Fahrzeug ausparkt, rücken die anderen Fahrzeuge autonom nach vorne bzw. hinten, sodass um das ausparkende Fahrzeug herum genügend Platz zum Ausparken entsteht. In jeder anderen Situation rücken die C-IPS Akteure autonom so zueinander auf, dass zwischen ihnen der minimal nötige Abstand gehalten wird. Daraus ergeben sich die folgenden Anforderungen an die Interaktion der C-IPS Akteure.

- Der Gesamtabstand zwischen den Akteuren ist möglichst minimal.
- Deswegen verlässt zu jedem Zeitpunkt maximal ein Akteur seine Formation.
- Trotzdem kann jedes Fahrzeug den Parkstreifen verlassen.

Im Folgenden wird vorausgesetzt, dass die C-IPS Akteure über eine vollständige, korrekte Formation verfügen und technisch dazu in der Lage sind, den nach vorne bzw. hinten verfügbaren Abstand zu messen, sich autonom zu bewegen und wie zuvor beschrieben zu kommunizieren.

Die Steuerung der interaktiven Abläufe unterscheidet übergeordnet zwei Fälle abhängig davon, ob es einen Akteur gibt, der die Intention hat auszuparken.

Priorisierung simultaner Intentionen verschiedener Akteuren derselben Formation

Da mit C-IPS minimale Abstände eingehalten werden, steht zu jedem Zeitpunkt nur so viel Abstand zwischen den Fahrzeugen zur Verfügung, dass maximal ein Akteur ausparken kann. Deswegen muss das Verhalten der Akteure für den (eher selten eintretenden) Fall spezifiziert werden, dass mehrere Akteure gleichzeitig vorhaben auszuparken. C-IPS priorisiert daher gleichzeitig auftretende Vorhaben auszuparken nach dem Prinzip *First Come First Serve*. Die Fahrzeuge parken also in der Reihenfolge aus, in der sie ihre Intention dazu im System melden. Dazu sendet jeder Akteur bei der Bestimmung der Formation, falls vorhanden, seine Intention auszuparken, falls es keine ältere Intention innerhalb der Formation gibt. Sobald die Formation vollständig aufgebaut ist, ist durch dieses Verfahren eindeutig bestimmt, welcher Akteur zuerst die Intention auszuparken angemeldet hat. Die Synchronisation der vollständigen Formation nach vorne berechtigt dann genau diesen Akteur dazu auszuparken. Sobald dieser den Parkstreifen, und damit die Formation, verlassen hat wird der Prozess zusammen mit der Aktualisierung der Formation wiederholt. Das bedeutet insbesondere, dass die Formation auch immer dann aktualisiert werden muss, wenn sich die Intention eines ihrer Akteure verändert.

Kooperation beim Verlassen der Formation

Zu jedem Zeitpunkt, zu dem genau ein Akteur der Formation ausparkt, kommuniziert dieser zunächst seine Intention auszuparken. Die anderen Akteure reagieren darauf, indem sie anhand der Formation ermitteln, ob sie sich vor oder hinter dem ausparkenden Akteur befinden. Basierend darauf rücken die Akteure nach vorne bzw. nach hinten, bis um den ausparkenden Akteur herum ausreichend Platz verfügbar ist, sodass dieser die Formation verlassen kann. Das Ausparken ist dabei nicht notwendigerweise ein autonomer Prozess und nicht Teil von C-IPS und es ist nicht entscheidend, ob dafür ein Parkassistent verwendet wird oder der Fahrer selbst ausparkt.

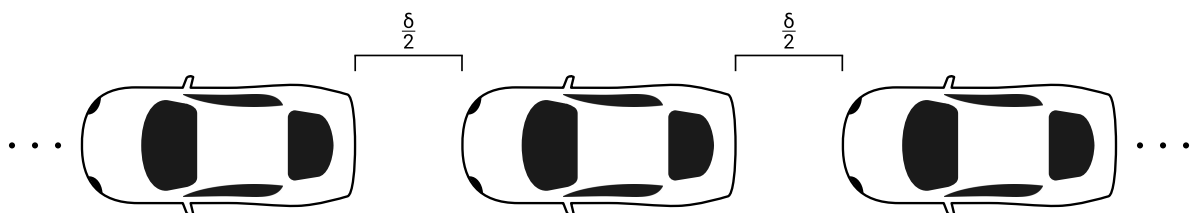


Abbildung 3: Darstellung der Situation nach dem kooperativen Aufrücken

Um sicherzustellen, dass stets maximal ein Akteur der Formation ausparkt, kommuniziert ein Akteur erst dann seine Absicht auszuparken, wenn der vorherige Ausparkvorgang abgeschlossen ist. Dafür kommuniziert jeder Akteur unmittelbar nach Verlassen der Formation, dass der Ausparkvorgang abgeschlossen wurde.

Damit für jeden Akteur ausreichend Platz geschaffen werden kann, muss auf die Formation verteilt der in der Lösungsspezifikation beschriebene Gesamtabstand $D_{ges} = \delta_{max} + \Lambda$ gehalten werden. Im Folgenden wird der Mechanismus beschrieben, der dies sicherstellt.

Kooperation bei der dynamischen Minimierung ungenutzter Abstände

Zu jedem Zeitpunkt, zu dem kein Akteur der Formation ausparkt, wird gewährleistet, dass die Akteure innerhalb der Formation insgesamt den Abstand $D_{ges} = \delta_{max} + \Lambda$ halten. Auf diese Weise steht, auf die Formation verteilt, genau so viel Platz zur Verfügung, dass unter Wahrung von Sicherheitsabständen

zwischen den Fahrzeugen, einem beliebigen Akteur genügend Platz zum Ausparken geschaffen werden kann. Dafür berechnen alle Akteure unabhängig voneinander anhand ihrer Formation F den Abstand $D_{ges} = \delta_{max} + n \cdot \lambda$ durch $\delta_{max} = \max\{\delta_x | (\delta_x, id_x) \in F\}$ und $n = |F|$. Eine Möglichkeit, diesen Abstand auf der Formation zu verteilen ist, vor jedem Akteur a_i den Abstand $d_i = \frac{\delta_{max}}{n} + \lambda$ zu halten. Es ist nicht konzeptionell notwendig, die Abstände genau auf diese Weise zu verteilen. Diese Anordnung bietet jedoch den Vorteil, dass zwischen allen Akteuren mit annähernd gleichem Aufwand der zum Ausparken notwendige Platz geschaffen werden kann. Wenn sich die Formation der Akteure aufgrund ein- oder ausparkender Fahrzeuge verändert, wird D_{ges} neu berechnet, sodass die Akteure ggf. veränderte Abstände zueinander autonom einnehmen. Ein C-IPS Akteur kann sich also in eine bestehende Formation einordnen, indem er vor oder hinter dieser Formation einparkt.

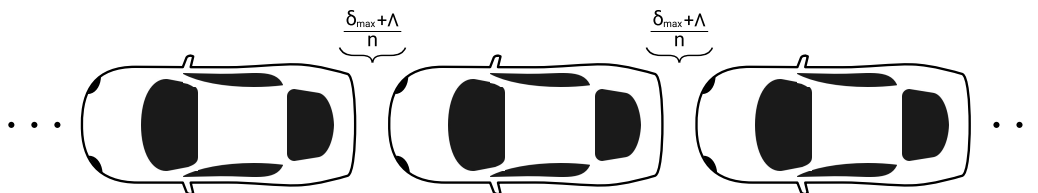


Abbildung 4: Gleichmäßige Verteilung der Abstände zwischen den C-IPS Akteuren

4 Implementierung von Testfahrzeugen

In diesem Abschnitt wird zur Untersuchung der technischen Realisierbarkeit von C-IPS auf die praktische Entwicklung von Testfahrzeugen eingegangen. Dafür wurden mehrere Modellfahrzeuge konstruiert, die grundlegende C-IPS Anforderungen erfüllen und für die außerdem eine mögliche Implementation der C-IPS Software entwickelt wurde. Sowohl die Konstruktion der Modellfahrzeuge als auch der Entwurf und die Programmierung der C-IPS Software sind in ihren Details umfangreich und werden daher lediglich in Grundzügen geschildert.

4.1 Entwurf und Konstruktion von Modellfahrzeugen

Für die Implementation von C-IPS, ergeben sich aus der theoretischen Modellierung technische Anforderungen, die insbesondere den Antrieb, die Sensorik und die Steuerungseinheit eines Modellfahrzeugs betreffen.

- Das Fahrzeug kann mittels Software bei geringen Geschwindigkeiten präzise lenken und fahren, damit es auf einem Parkstreifen autonom und sicher manövrieren kann.
- Damit die Akteure sich gegenseitig identifizieren und Abstände messen können, verfügen die Fahrzeuge über entsprechende Sensoren.
- Die Steuerungseinheit ermöglicht es, parallel Sensordaten zu verarbeiten, Prozesse auszuführen und eine V2V Schnittstelle zu nutzen.

Antrieb

Das verwendete Grundmodell eines Elektrofahrzeugs im Maßstab eins zu zehn [RC] ist im Lieferzustand für die Verwendung eines herkömmlichen Gleichstrommotors sowie eine Antriebsübersetzung bestehend aus einem Stirnrad und einem Antriebsritzel des Standards 48P³ ausgelegt.

Diese Grundausstattung wurde ersetzt, da ein Gleichstrommotor entsprechend seines Aufbaus bei geringen Stromflüssen nur eine ungenaue Geschwindigkeitskontrolle zulässt, die es erschwert, Fahraufgaben wie das Ein- und Ausparken zu bewältigen, da diese hohe Präzision beim Fahren mit geringen

³ 48P und 32P beschreiben verschiedene Standards zum Aufbau von Getrieberädern, die sich auf den Zahnabstand beziehen. Dabei hat 32P größere Zähne, die weiter auseinanderstehen.

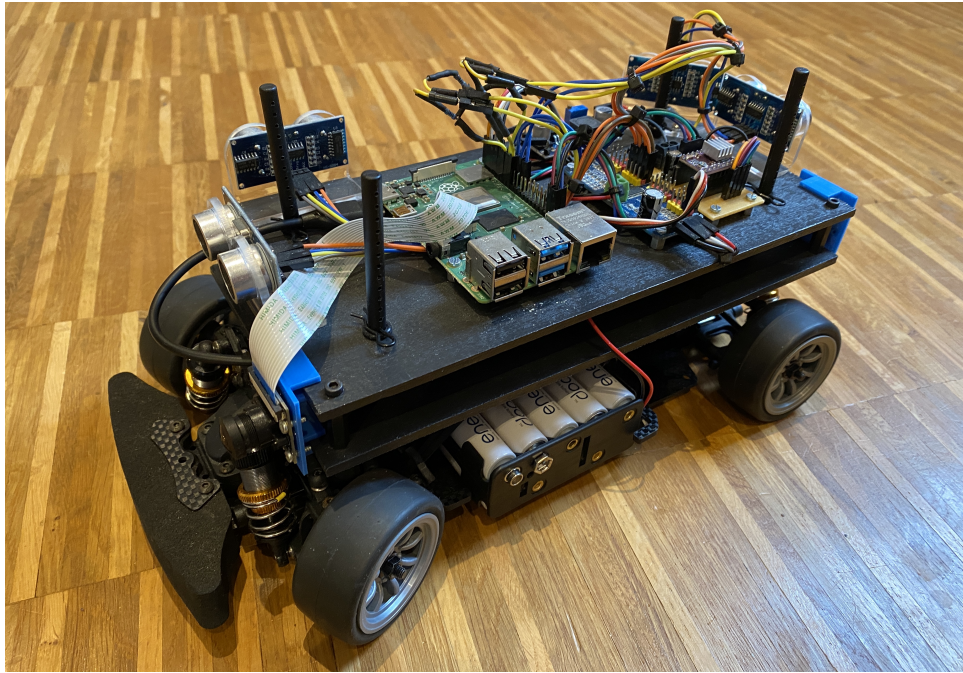


Abbildung 5: Bild von einem der drei Modellfahrzeuge

Geschwindigkeiten voraussetzen.

Beim Umbau des Modells kam ein Schrittmotor vom Typ Nema17 [Ste] zum Einsatz. Die Verwendung eines Schrittmotors erlaubt dem Fahrzeug dabei auch sehr geringe Geschwindigkeiten und kleine Strecken mit großer Präzision zu fahren, da die kleinste fahrbare Einheit mit der verwendeten Kombination aus Motor und Übersetzung wenige Millimeter beträgt. Um die vom Schrittmotor ausgeübte Kraft richtig übersetzen zu können, wurden die Antriebsräder mit Einheiten ersetzt, die dem Standard 32P entsprechen. Diese kommen normalerweise in größeren Modellen mit dem Maßstab eins zu zwölf zum Einsatz und haben größere und damit stabilere Zähne, die auch für die Übertragung großer Kräfte geeignet sind.

Ansteuerung des Motors

Um die verschiedenen Fahraufgaben, welche das System voraussetzt, bewältigen zu können, verwenden die entwickelten Testfahrzeuge je einen Motor für die Lenkung und einen weiteren Motor, der das Fahrzeug vorwärts und rückwärts bewegt.

Die zentrale Steuereinheit ist dabei ein Einplatinencomputer der Bauart Raspberry Pi [Foub]. Dieser Computer wurde aufgrund seiner Fähigkeit gewählt, sowohl die Kommunikation über WLAN, die Datenverarbeitung als auch die Ansteuerung der Sensoren und Motoren bewältigen zu können.

Die Lenkung basiert dabei auf einem Servomotor mit angebauter Achse. Da dieser mittels Pulsweitenmodulation⁴ (PWM) angesteuert wird, ist ein vom Raspberry Pi externer Servocontroller [Ada] verbaut. Dieser hat eine höhere Präzision bei der Ausgabe von PWM-Signalen als die im Raspberry Pi integrierten GPIO-Schnittstellen⁵ und verfügt zudem über die Möglichkeit, eine externe Stromquelle anzuschließen, um den Raspberry Pi vor zu hohen Stromflüssen zu schützen. Da der Motor nicht direkt mit den Rädern verbunden ist, wurde anhand von Messdaten eine Funktion aufgestellt, die den Einschlagwinkel der Räder in einen PWM-Wert umrechnet.

Der Antrieb des Fahrzeugs erfolgt durch einen Schrittmotor. Für die Ansteuerung dieses Mo-

⁴ Pulsweitenmodulation (PWM) ist ein System, mit dem die Spannung eines Ausgangs durch schnelles Ein- und Ausschalten künstlich zwischen dem Maximalwert und null Volt verstellbar ist.

⁵ General Purpose In / Out (GPIO) ist eine Schnittstelle, die mittels Software analoge und digitale Signale ausgeben und lesen kann.

tortyps wird eine zusätzliche Steuereinheit [Pol] verwendet. Die Steuerung des Motors erfolgt über einen Schrittpin, der den Motor um $\frac{1}{200}$ einer Umdrehung dreht sowie über einen Richtungspin. Auch hier wurde aufgrund der Übersetzung mittels Zahnrädern und Achsen eine neue Funktion ermittelt, welche die Anzahl an Schritten berechnet, die für eine bestimmte Distanz notwendig sind.

Sensorik und eindeutige Identifizierung

Um Abstände messen und Akteure identifizieren zu können, werden Abstandssensoren und eine Kamera verwendet.

Mithilfe von Ultraschallsensoren [Spa], die in ähnlicher Form auch für Parkassistenzsysteme genutzt werden, wird die Distanz zwischen dem Fahrzeug und einem festen Körper ermittelt, indem der jeweilige Sensor ein Ultraschallsignal aussendet. Basierend auf der Zeitdifferenz zwischen dem Aussenden und dem Empfang des reflektierten Signals wird die dabei zurückgelegte Strecke berechnet. Am Fahrzeug sind Ultraschallsensoren verbaut, die die Distanz nach vorne, hinten und rechts ermitteln.

Die Kamera [Foua] ist so am Fahrzeug befestigt, dass sie den Bereich der Straße nach vorne erfasst und ist über einen CSI-2⁶ Anschluss direkt mit dem Raspberry Pi verbunden. Zudem ist am Heck des Fahrzeugs ein QR-Code angebracht, der die ID des jeweiligen Akteurs kodiert. Mithilfe der Kamera kann ein Akteur daher ein vor ihm parkendes Fahrzeug eindeutig identifizieren.

4.2 Implementation der C-IPS Software

Die Umsetzung der C-IPS Software beruht auf dem zuvor entwickelten theoretischen Modell und ist öffentlich auf GitHub [A5] zugänglich. Das Programm ist in Python3 implementiert und beruht auf einem objektorientierten Entwurf. Dabei orientiert sich die Klassenstruktur an der Unterteilung von C-IPS in die grundlegenden Bestandteile der V2V Kommunikation, der Formation und der Kooperation. Die Implementation der entsprechenden Klassen wird im Folgenden erläutert. Dabei wird nicht weiter auf die darüber hinausgehenden Programmteile, wie etwa zur Ansteuerung der Hardware, eingegangen, da sie zwar notwendig, aber nicht wesentlich für C-IPS sind.

V2V Kommunikation

C-IPS Akteure stellen sicher, über Ad-hoc-Netzwerke miteinander verbunden zu sein, indem sie sich stets mit dem aktuellsten C-IPS Netzwerk in ihrer Umgebung verbinden oder eines erstellen, falls keines verfügbar ist.

```
1      program etabliere_netzwerk:
2          if Netzwerk vorhanden:
3              stelle Verbindung zu Netzwerk her
4          else:
5              erstelle Netzwerk
```

Pseudocode: Aufbau eines Ad-hoc-Netzwerks

Die Klasse *Communication* implementiert die Kommunikationsschnittstelle, die von anderen Programmteilen zum Austausch von Informationen zwischen den Akteuren genutzt wird. Sie stellt insbesondere die Methode *send* zur Verfügung, die eine Nachricht innerhalb des Netzwerks versendet. Dafür wird eine Instanz der Klasse *Message* erzeugt, die den Empfänger, den Inhalt und den Zeitpunkt einer Nachricht speichert und im JSON⁷ Format kodiert werden kann. Diese JSON-Repräsentation wird an alle Akteure des Netzwerks versandt, die die Nachricht dann dekodieren und auswerten. Zum Austausch von Nachrichten betreibt jeder Akteur einen Webserver, der Daten innerhalb des lokalen Netzwerks über HTTP⁸ versendet und empfängt.

⁶ Das Camera Serial Interface (CSI) ist eine Schnittstelle zwischen einer Kamera und einem Prozessor.

⁷ Die JavaScript Object Notation (JSON) ist ein Datenformat zum Datenaustausch zwischen Anwendungen.

⁸ Das Hypertext Transfer Protocol (HTTP) ist ein Protokoll zur Übertragung von Daten in einem Rechnernetz.

Formation

Die Klasse *Formation* stellt eine Formation in Form einer Liste dar, die die IDs der zugehörigen Akteure enthält. Zusätzlich wird die maximale Länge der Fahrzeuge gespeichert, aus der der Abstand berechnet werden kann, der maximal zum Ausparken benötigt wird. Neben der Modellierung einer Formation stellt die Klasse außerdem Methoden zur Synchronisation der Formation unter den Akteuren zur Verfügung. Der Programmablauf untergliedert sich, entsprechend der Modellierung, in zwei Phasen, in denen die Formation zuerst sukzessive nach hinten aufgebaut wird und dann als vollständige Formation wieder nach vorne gesendet wird. Dies geschieht durch die Methoden *send_backward_pass*, *receive_backward_pass*, *send_confirmation*, *send_forward_pass* und *receive_forward_pass*. Die Synchronisation wird durch die Methode *update* initiiert, indem regelmäßig, in einem Intervall von einer Sekunde, geprüft wird, ob der jeweilige Akteur der erste in der Formation ist. In diesem Fall beginnt der Ablauf zur Synchronisation der Formation. Auf diese Weise spiegelt die Formation stets den Zustand des Parkstreifens zu Beginn des letzten Intervalls wieder.

```
1      program baue_formation_auf:
2          use Kamera as scanner:
3          if not (scanner identifiziert Akteur):
4              formation := [eigene_id, eigene_laenge]
5              sende(formation)
```

Pseudocode: Ausgangspunkt der Synchronisation der Formation

Der Empfänger einer vollständigen Formation überschreibt seine aktuelle Formation, falls er selbst darin enthalten ist und der Zeitstempel aktueller ist als der seiner gespeicherten Formation.

Kooperation

Die Interaktion der Akteure wird durch die Klasse *ActionManager* gesteuert, welche die Intentionen der Akteure innerhalb der Formation protokolliert und darauf basierend das Verhalten der Akteure aufeinander abstimmt. Dafür wird ein Einigungsprozess implementiert, der bestimmt, welcher Akteur zu einem konkreten Zeitpunkt den Parkstreifen verlassen darf und so sicherstellt dass, entsprechend der Systemvoraussetzung, stets maximal ein Fahrzeug auf einmal ausparkt. Dafür sendet ein ausparkender Akteur mithilfe der Methode *send_global_action* seine Intention auszaparken, die von den anderen Akteuren der Formation erhalten und in der Methode *receive_global_action* verifiziert wird. Die Verifizierung stellt sicher, dass jeder Akteur die älteste, gesendete Intention speichert. Um dabei die Latenzen bei der Kommunikation zu berücksichtigen, prüft der ausparkende Akteur nach Senden seiner Intention, ob seine Intention innerhalb eines Zeitintervalls durch eine andere überschrieben wird. Ist das nicht der Fall, wurde seine Intention damit von jedem Akteur als die am höchsten priorisierte bestätigt. Die Methode *act* definiert das Folgeverhalten aller Akteure: Der ausparkende Akteur wartet darauf, dass die anderen Akteure um ihn herum genügend Platz schaffen, indem sie nach vorne bzw. hinten rücken bis sie nur noch Sicherheitsabstände halten. Dafür wird die Methode *create_space* der Steuerungsklasse *Driver* verwendet, die auf die Formation zurückgreift, um zu bestimmen, in welche Richtung der jeweilige Akteur aufrücken muss, um Platz zu schaffen. Der ausparkende Akteur verlässt den Parkstreifen und bestätigt danach den Abschluss des kooperativen Prozesses mithilfe der Methoden *send_completion*. Nach Erhalt der Bestätigung wird durch die Methode *receive_completion* die letzte gespeicherte Intention gelöscht. Neben der Methode *create_space* implementiert die Steuerung auch die Methode *move_up* zur dynamischen Minimierung ungenutzter Abstände. Dafür wird mit der Methode *calc_gap* der aktuellen Formation der zu haltende Abstand zwischen zwei Akteuren berechnet und autonom eingenommen.

5 Ergebnis

Die Testfahrzeuge wurden entwickelt, um die Umsetzbarkeit von C-IPS in der Praxis zu untersuchen. Im Folgenden wird die potenzielle Effizienzsteigerung durch C-IPS als Modell beschrieben und die entwickelten Testfahrzeuge mit diesem Modell verglichen. Insbesondere werden Unterschiede zwischen Modell und Testfahrzeugen sowie wesentliche Implementationsentscheidungen erläutert, um darzulegen, zu welchem Grad die Testfahrzeuge der C-IPS Spezifikation entsprechen.

5.1 Einschätzung der potenziellen Effizienzsteigerung

Nach der Problemspezifikation kann der Platz S_1 auf einem Parkstreifen, der von herkömmlichen Akteuren mit der durchschnittlichen Länge l und dem durchschnittlichen Abstand d zum Parken benötigt wird, in Abhängigkeit von der Anzahl n der Fahrzeuge durch $S_1(n) = n \cdot (l + d) + d$ beschrieben werden. Bei Anwendung von C-IPS wird der Platz $S_2(n) = n \cdot (l + \lambda) + \delta_{max}$ benötigt.

$$\begin{aligned} S_1(n) &= S_2(n+1) \\ \Leftrightarrow n \cdot (l + d) + d &= (n+1) \cdot (l + \lambda) + \delta_{max} \\ \Leftrightarrow n \cdot d &= l + (n+1) \cdot \lambda + \delta_{max} \\ \Leftrightarrow n \cdot (d - \lambda) &= l + \lambda + \delta_{max} \\ \Leftrightarrow n &= \frac{l + \lambda + \delta_{max}}{d - \lambda} \end{aligned}$$

Geht man von einer mittleren Länge der Autos von $l = 4,4m$, einem Sicherheitsabstand von $\lambda = 0,05m$, einem mittleren Abstand von $d = 0,85m$ und einem maximal benötigten Abstand von $\delta_{max} = 1m$ aus, so lässt sich n bestimmen als:

$$n = \frac{4,4m + 0,05m + 1m}{0,85m - 0,05m} = \frac{5,45m}{0,8m} = 6,8125$$

Demnach können im Durchschnitt auf einem Parkstreifen, auf dem normalerweise sieben Fahrzeuge parken können, acht C-IPS Fahrzeuge parken. Das entspricht einer Effizienzsteigerung von etwa 14,29%.

5.2 Vergleich von Modell und Testfahrzeugen

Der Entwurf der Modellfahrzeuge orientiert sich in relevanten Aspekten an dem Aufbau von PKWs. Die Ultraschallsensoren sowie die Kamera sind wesentliche Bestandteile der Konstruktion und werden in ähnlicher Form für Fahrerassistenzsysteme verwendet. So verfügen z. B. bereits viele Fahrzeuge über Parkassistenzsysteme, die Ultraschallsensoren sowie Kameras einsetzen und deren technische Anforderungen daher mit denen von C-IPS vergleichbar sind. Antrieb und Steuerung unterscheiden sich zwar wesentlich von der Umsetzung in PKWs, jedoch ist im Hinblick auf C-IPS der Aufbau von Motor und Getriebe nicht ausschlaggebend, solange die Steuerung und die Sensoren über Software adressierbar sind. Dies ist in den meisten modernen Fahrzeugen möglich, da viele neue Modelle bereits autonome Assistenzsysteme verbauen oder zumindest mithilfe des OBD⁹ Ports sowie des CAN Bus¹⁰ autonom gesteuert werden können. Der größte Unterschied zwischen Modellierung und Umsetzung der Modellfahrzeuge besteht in der Umsetzung der V2V Kommunikationsschnittstelle auf Hardwareebene. Da DSRC, ITS G5 und C-V2X noch erforscht werden, ist entsprechende Hardware noch nicht auf dem Massenmarkt vorhanden, geringfügig dokumentiert und daher nur unter hohem Aufwand zu integrieren. Daher wurde ein Chip verwendet, der anstatt des IEEE 802.11p Standards für V2V Direktkommunikation auf den gewöhnlichen WLAN Standards IEEE 802.11ac bzw. IEEE 802.11n basiert. Folglich orientiert sich die Modellumsetzung eher an DSRC bzw. ITS G5, da die Akteure direkt miteinander kommunizieren anstatt Daten über ein externes Netzwerk auszutauschen.

Aufgrund dieser Hardwarelimitierung wurde für die Implementation der V2V Kommunikationsschnittstelle ein Verfahren zur Datenübertragung mittels Webserver und HTTP gewählt. Der Modellaufbau zeigt also, dass auch herkömmliches WLAN für die Implementation von C-IPS geeignet ist. Aufgrund der absehbar zunehmenden Verbreitung von DSRC, ITS G5 oder C-V2X, wäre der Einsatz einer dedizierten V2V Technologie allerdings dennoch prinzipiell sinnvoll.

Eine weitere Implementationsentscheidung besteht darin, die eindeutige Identifizierung über einen QR Code am Heck eines Akteurs zu realisieren, der eine zufällige aber eindeutige ID kodiert, die durch einen anderen Akteur mithilfe der Kamera ausgelesen werden kann. Diese Umsetzung ist weniger fehleranfällig als das Auslesen eines Nummernschildes, da QR Codes, anders als Nummernschilder, dazu konzipiert sind, automatisiert und optisch ausgewertet zu werden.

Da die Modellfahrzeuge baugleich sind, umfasst die Implementation der Formation statt aller Abstände nur den maximalen Abstand, den ein Fahrzeug zum Ausparken benötigt. Zudem wird die Formation fortlaufend aktualisiert. Auf diese Weise wird verhindert, dass bei unvorhergesehenen

⁹ Über den On-Board-Diagnose (OBD) Port können Fahrzeugdaten ausgelesen und Anweisungen an das Fahrzeug übermittelt werden.

¹⁰ Der Controller Area Network (CAN) Bus dient unter anderem der Vernetzung von Steuergeräten und Sensoren.

Veränderungen der Formation (z. B. durch den Ausfall eines Akteurs) bei der Kooperation der Akteure eine falsche Formation verwendet wird.

Eine wichtige Implementationsentscheidung, die bewusst nicht durch C-IPS spezifiziert wird, ist die Strategie, den Abstand der Akteure gleichmäßig auf die gesamte Formation zu verteilen. Auf diese Weise werden alle Akteure insofern gleichberechtigt, als dass um jedes Fahrzeug herum mit ähnlichem Zeitaufwand Platz geschaffen werden kann. Grundsätzlich wären aber auch Strategien denkbar, die z. B. Heuristiken nutzen, um abzuschätzen, in welcher Reihenfolge die Akteure den Parkstreifen verlassen werden, wodurch sich der Zeitaufwand und die Bewegung der Akteure minimieren ließen.

Dass die modellhafte Umsetzung der Testfahrzeuge nicht wesentlich von der C-IPS Spezifikation abweicht zeigt, dass C-IPS prinzipiell technisch umsetzbar ist. Die Orientierung an verbreiteter Hardware und bestehenden Assistenzsystemen beim Entwurf der Implementation legen außerdem nahe, dass C-IPS auch im größeren Maßstab realisierbar ist.

6 Ergebnisdiskussion

In diesem Kapitel werden die zuvor beschriebenen Ergebnisse ausgewertet, indem der Nutzen und die Realisierbarkeit von C-IPS bewertet wird. Außerdem werden weitergehende Potenziale beschrieben, die über das im Rahmen dieser Arbeit modellierte System hinausgehen und nächste Entwicklungsschritte genannt.

6.1 Nutzen von C-IPS

Wie Kapitel 5.1 zeigt, kann durch Anwendung von C-IPS eine erhebliche Effizienzsteigerung bei der Nutzung von Parkstreifen erreicht werden, wodurch das Verkehrsaufkommen insbesondere in Innenstädten wesentlich verringert und damit externe Kosten reduziert würden.

Um die volle Effizienzsteigerung zu erreichen, ist allerdings ein hoher Anteil von C-IPS Akteuren im Straßenverkehr notwendig, da der Nutzen von C-IPS mit der Anzahl der Fahrzeuge wächst, die es implementieren, weil erst mehrere hintereinander parkende C-IPS Akteure eine deutliche Effizienzsteigerung bewirken. Daher bietet C-IPS insbesondere in ersten Anwendungen des Systems dann einen Vorteil, wenn es für spezifische Fahrzeuge genutzt wird, die häufig in Reihen geparkt werden. Daher umfassen denkbare Erstanwendungen unter anderem LKWs oder Lieferfahrzeuge. Um den Nutzen für Einzelanwender zu steigern, ist die Integration von C-IPS in andere Assistenzsysteme, wie z. B. autonome Park- und Rufsysteme, sinnvoll, da durch die Bündelung von Systemen Synergien entstünden, die zusätzlich die Attraktivität von C-IPS für erste Anwender erhöhen würden.

Unabhängig von der möglichen Effizienzsteigerung bietet C-IPS das Potenzial zu einem sicheren ersten Schritt für autonome Mobilität, da C-IPS als Parksysteem bei geringen Geschwindigkeiten in einer beschränkten, sicheren Umgebung unüberwacht autonom operiert. Das bietet die Möglichkeit, anhand eines sicheren Systems, grundlegende juristische und gesellschaftliche Problemstellungen bezüglich autonomer Mobilität zu klären, deren Lösungen auf weniger beschränkte Systeme, wie z. B. autonomes Fahren auf Autobahnen, übertragen werden können. Auf diese Weise können rechtliche Fragen sowie gesellschaftliche Akzeptanz untersucht bzw. erprobt werden, was C-IPS als geeignetes Einstiegsszenario für fortgeschrittene autonome Mobilität qualifiziert.

6.2 Realisierbarkeit von C-IPS

Da die Testfahrzeuge so entworfen wurden, dass die verwendete Hardware skalierbar ist und viele Bauteile durch die zunehmende Verbreitung von Assistenzsystemen bereits in vielen Fahrzeugen verbaut sind, ist es möglich, C-IPS mit vertretbarem Aufwand in realen Anwendungen zu implementieren. Daher kann für bestehende Fahrzeuge die für C-IPS benötigte Hardware nachgerüstet und die Software mithilfe des OBD Ports und des CAN Bus implementiert und installiert werden. Die dabei anfallenden Kosten können als verhältnismäßig gering angenommen werden. Die größte Hürde besteht dabei in der V2V Kommunikation, da noch nicht absehbar ist, welcher Standard sich in Forschung und Praxis durchsetzen wird und die für V2V Kommunikation benötigte Hardware dementsprechend wenig verbreitet ist.

Da zur Umsetzung von C-IPS lediglich die Fahrzeuge selbst ausgestattet werden müssen, ist ein Umbau der Infrastruktur auf Parkplätzen nicht erforderlich. Um die größtmögliche Effizienzsteigerung erreichen, muss C-IPS aber von verschiedenen Anbietern, wie etwa den Automobilherstellern oder Drittanbietern, implementiert und nachgerüstet werden können. Damit Fahrzeuge verschiedener Anbieter kompatibel sind und miteinander kooperieren können, müssen verschiedene Implementationen einem einheitlichen C-IPS Standard entsprechen. Die Tatsache, dass die diskutierten V2V Technologien DSRC und C-V2X nicht miteinander kompatibel sind [NP19, S. 70181], verdeutlicht diese Problematik.

Bevor allerdings C-IPS Anwendungen in der Praxis denkbar sind, ist eine intensive Untersuchung möglicher Sicherheitsrisiken und die Entwicklung von Sicherheitsmechanismen notwendig. Dies umfasst zum einen die Sicherheit des Netzwerks, in dem die Akteure miteinander kommunizieren, sowie die Validierung externer Informationen durch Plausibilitätsprüfungen. Zum anderen ist die Entwicklung von Sicherheitsmechanismen erforderlich, die gewährleisten, dass C-IPS auch bei aktiver Manipulation, insbesondere bei der Verarbeitung falscher Informationen, kein Sicherheitsrisiko darstellt.

6.3 Ausblick

Da die Modellfahrzeuge bisher ausschließlich Ultraschallsensoren zur Erfassung des Parkstreifens nutzen, sollen zukünftig für eine realistischere Modellierung der Parksituation auch etwa das Kamerabild des Akteurs verstärkt ausgewertet werden. Das ermöglicht es, weniger idealisierte Grundvoraussetzungen an die Beschaffenheit eines Parkstreifens zu stellen, da basierend auf einer visuellen Auswertung der Umgebung etwa auch gekrümmte Parkstreifen genutzt werden könnten. Außerdem würde ein sichereres Verhalten der Akteure ermöglicht, da auf diese Weise auch Hindernisse und andere Verkehrsteilnehmer erkannt werden können, die nur mit Abstandssensoren nicht zuverlässig erfasst werden können.

Um die Umsetzbarkeit von C-IPS im größeren Maßstab zu untersuchen, bestünde der nächste Schritt darin, das System für größere Fahrzeuge zu implementieren und zu testen. In diesem Kontext sind insbesondere auch weitere Fortschritte bei der Erforschung und Anwendung von V2V Technologie von Relevanz, da sich dadurch potenziell Möglichkeiten zur Integration von C-IPS in andere V2V Anwendungen erschließen lassen.

Eine weitere Möglichkeit, Parkraum effizienter zu nutzen, besteht darin, die durch C-IPS gesammelten Daten zum Parkverhalten anonymisiert auszuwerten und dafür zu verwenden, die Abstände in einer Formation besser zu verteilen. Dafür ließen sich mithilfe entsprechender Datenmengen statistische Aussagen darüber machen, welche Akteure den Parkstreifen als nächstes verlassen. Darauf aufbauend könnte ein Algorithmus entwickelt werden, der diese statistischen Informationen nutzt, um möglichst günstige Abstände zwischen den Fahrzeugen zu berechnen. Durch die Auswertung von Echtzeit-Daten ließen sich dahingehend weitere Potenziale erschließen, indem die Fahrzeuge untereinander Informationen darüber austauschen, an welchen Stellen zur Zeit freie Parkplätze zur Verfügung stehen.

7 Reflexion

Diese Arbeit beruht auf einem Projekt, mit dem wir bereits beim Regionalwettbewerb Jugend Forscht 2019/20 angetreten sind. Ziel der Weiterentwicklung unserer Arbeit war es, den wissenschaftlichen Fokus des Projekts zu schärfen, dabei methodisch formaler vorzugehen und wissenschaftlich rigoröser zu arbeiten. Daher befassen wir uns nun intensiver und primär mit der formalen Modellierung und Spezifikation von C-IPS und implementieren das System nur zu Test- und Evaluationszwecken. In dieser Arbeit befassen wir uns nicht weiter mit autonomem Fahren durch künstliche Intelligenz im Allgemeinen, um stattdessen ausführlicher und präziser auf die theoretische Modellierung des Parksystems eingehen zu können. Darüber hinaus haben wir die Entwicklung der Testfahrzeuge dahingehend fertiggestellt, dass die in dieser Arbeit beschriebenen Funktionen nun modellhaft präsentiert werden können.

Dennoch kann die Umsetzung der Testfahrzeuge weiter optimiert werden, da noch kleinere Bugs in der Software bestehen, die wir vor der weiteren Entwicklung beheben werden. Zudem ist der Code bisher nicht ausreichend dokumentiert, weswegen wir insbesondere die Kommentierung des Programms verbessern werden. Außerdem haben wir festgestellt, dass wir durch rigorosere Planung beim Softwareentwurf die Implementation eleganter hätten gestalten können. Daher werden wir bei der Implementierung zukünftiger Funktionalitäten ausführlicher planen und modellieren.

8 Unterstützungsleistungen und Danksagung

Abschließend möchten wir uns bei all denjenigen bedanken, die uns bei der Umsetzung dieses Projekts unterstützt haben. Das schließt vor allem unseren Betreuer Hendrik Büdding ein, der uns neben fachlichen Anregungen auch methodisch sowie bei der Planung und Finanzierung dieses Projekts geholfen hat. Unser besonderer Dank gilt auch der Gerlind & Ernst Denert-Stiftung, der Sparkasse Münsterland Ost und dem Förderverein des Annette-von-Droste-Hülshoff-Gymnasiums für ihre finanzielle Unterstützung, ohne die wir dieses Projekt nicht hätten umsetzen können. Wir bedanken uns außerdem bei Herrn Professor Denert für seine fachlichen Anregungen.

A Informationen zur Implementation

- [Adab] Adafruit-PCA9685. Version 1.0.1. URL: https://github.com/adafruit/Adafruit_Python_PCA9685. zuletzt abgerufen am 15.01.2021.
- [fla] flask. Version 1.1.1. URL: <https://github.com/pallets/flask>. zuletzt abgerufen am 15.01.2021.
- [net] netifaces. Version 0.10.9. URL: <https://github.com/al45tair/netifaces>. zuletzt abgerufen am 15.01.2021.
- [num] numpy. Version 1.16.4. URL: <https://github.com/numpy/numpy>. zuletzt abgerufen am 15.01.2021.
- [par] parknet. Version 0.2.0. URL: <https://github.com/lukashaverbeck/parknet>. zuletzt abgerufen am 15.01.2021.
- [Pil] Pillow. Version 6.2.0. URL: <https://github.com/python-pillow/Pillow>. zuletzt abgerufen am 15.01.2021.
- [PyA] PyAccessPoint. Version 0.2.5. URL: <https://github.com/Goblenus/pyaccesspoint>. zuletzt abgerufen am 15.01.2021.
- [pyz] pyzbar. Version 0.1.8. URL: <https://github.com/NaturalHistoryMuseum/pyzbar/tree/master/pyzba>. zuletzt abgerufen am 15.01.2021.
- [req] requests. Version 2.22.0. URL: <https://github.com/psf/requests>. zuletzt abgerufen am 15.01.2021.
- [RPi] RPi.GPIO. Version 0.7.0. URL: <https://pypi.org/project/RPi.GPIO>. zuletzt abgerufen am 15.01.2021.
- [wif] wifi. Version 0.6.1. URL: <https://github.com/rockymeza/wifi>. zuletzt abgerufen am 15.01.2021.
- [wir] wireless. Version 0.3.3. URL: <https://github.com/joshvillbrandt/wireless>. zuletzt abgerufen am 15.01.2021.

B Verwendete Bauteile

- [Aaaa] Adafruit. *PCA9685*. URL: <https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf>. zuletzt abgerufen am 10.01.2021.
- [Foua] Raspberry Pi Foundation. *Picamera*. URL: <https://www.raspberrypi.org/documentation/hardware/camera/>. zuletzt abgerufen am 10.01.2021.
- [Foub] Raspberry Pi Foundation. *Raspberry Pi 4B*. URL: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>. zuletzt abgerufen am 10.01.2021.
- [Pol] Pololu. *DRV8825*. URL: <https://www.pololu.com/file/0J590/drv8825.pdf>. zuletzt abgerufen am 10.01.2021.
- [RC] CARTEN RC. *M210-R*. URL: https://www.carten-rc.eu/product_info.php?info=p11010_m210r-m-chassis-kit.html,%20zuletzt%20abgerufen%20am%2012.01.2021. zuletzt abgerufen am 10.01.2021.
- [Spa] Sparkfun. *HC-SR04*. URL: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>. zuletzt abgerufen am 10.01.2021.
- [Ste] Stepperonline. *Nema17*. URL: <https://www.omc-stepperonline.com/download/17HS16-2004S1.pdf>. zuletzt abgerufen am 10.01.2021.

C Abbildungsverzeichnis

1	Exemplarische Darstellung der ineffizienten Nutzung eines Parkstreifens	6
2	Visuelle Darstellung der Synchronisation einer Formation	9
3	Darstellung der Situation nach dem kooperativen Aufrücken	10
4	Gleichmäßige Verteilung der Abstände zwischen den C-IPS Akteuren	11
5	Bild von einem der drei Modellfahrzeuge	12

D Quellen- und Literaturverzeichnis

- [Ass18] 5G Automotive Association. *V2X Functional and Performance Test Report; Test Procedures and Results*. 2018. URL: https://5gaa.org/wp-content/uploads/2018/11/5GAA_P-190033_V2X-Functional-and-Performance-Test-Report_final-1.pdf. zuletzt geprüft am 28.12.2020.
- [BL12] J. Bates and D. Leibling. *Spaced out. Perspectives on parking policy*. 2012. URL: <https://trid.trb.org/view/1225136>. zuletzt geprüft am 14.01.2021.
- [Con20] Car 2 Car Communication Consortium. *Lives and Investments are Safeguarded by ITS G5 Seamless Evolution*. 2020. URL: <https://www.car-2-car.org/press-media/press-releases/press-details/lives-and-investments-are-safeguarded-by-its-g5-seamless-evolution-93/>. zuletzt geprüft am 04.01.2021.
- [CP17] G. Cookson and B. Pishue. *The impact of parking pain in the US, UK and Germany*. 2017. URL: <http://inrix.com/research/parking-pain/>. zuletzt geprüft am 14.01.2021.
- [EN14] ETSI EN. *302 637-2 V1. 3.1-Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. 2014. URL: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01_03_01_30/en_30263702v010301v.pdf. zuletzt geprüft am 05.01.2021.
- [F A11] et al. F. Ahmed-Zaid. *Vehicle safety communications-applications (vsc-a) final report*. 2011. URL: <https://trid.trb.org/view/1136544>. zuletzt geprüft am 26.12.2020.
- [INR17] INRIX. *Deutsche Verschwenden 41 Stunden Im Jahr Bei Der Parkplatzsuche*. 2017. URL: <http://inrix.com/press-releases/parking-pain-de>. zuletzt geprüft am 03.01.2021.
- [J H14] et al. J. Harding. *Vehicle-to-vehicle communications: readiness of V2V technology for application*. 2014. URL: <https://rosap.ntl.bts.gov/view/dot/27999>. zuletzt geprüft am 30.12.2020.
- [K S17] et al. K. Sjöberg. *Cooperative intelligent transport systems in Europe: Current deployment status and outlook*. 2017. URL: <https://ieeexplore.ieee.org/abstract/document/7911287>. zuletzt geprüft am 27.12.2020.
- [NP19] B. Choudhury N. Gaurang and J. Park. *IEEE 802.11 bd & 5G NR V2X: Evolution of radio access technologies for V2X communications*. 2019. URL: <https://ieeexplore.ieee.org/abstract/document/8723326>. zuletzt geprüft am 04.01.2021.
- [SH20] R. Suthold and L. Harlos. *Dauerthema Parken: Kommunen müssen mehr Verantwortung übernehmen!* 2020. URL: <https://www.adac.de/der-adac/regionalclubs/nrw/nrw-kolumne-parken>. zuletzt geprüft am 07.01.2021.
- [TS12] ETSI TS. *102 724: Intelligent transport systems (ITS); harmonized channel specifications for intelligent transport systems operating in the 5 GHz frequency band*. 2012. URL: https://www.etsi.org/deliver/etsi_ts/102700_102799/102724/01_01_01_60/ts_102724v010101p.pdf. zuletzt geprüft am 05.01.2021.
- [WK17] M. Wilde and T. Klinger. *Städte für Menschen. Transformationen urbaner Mobilität*. 2017. URL: <https://www.bpb.de/apuz/260066/staedte-fuer-menschen-urbane-mobilitaet>. zuletzt geprüft am 14.01.2021.