

Automatisiertes Kreuzungsmanagement im Urbanen Mischverkehr mit Reinforcement Learning

Marvin Klimke^{*†} Benjamin Völz^{*} Michael Buchholz[†]

Zusammenfassung: Das vernetzte automatisierte Fahren hat das Potential, die Verkehrseffizienz im innerstädtischen Raum signifikant zu steigern. So können Verdeckungseffekte, wie sie beispielsweise durch Gebäude oder andere Verkehrsteilnehmer entstehen, aufgelöst werden. Darüber hinaus wird eine kooperative Manöverplanung ermöglicht, bei der die Bewegung mehrerer vernetzter automatisierter Fahrzeuge gemeinsam optimiert wird. Viele existierende Ansätze für das automatisierte Kreuzungsmanagement betrachten lediglich den vollständig automatisierten Verkehr. In der Praxis wird Mischverkehr, also die Interaktion von menschlichen Fahrern und automatisierten Fahrzeugen, vorherrschend sein. Im vorliegenden Beitrag wird ein Verfahren basierend auf Reinforcement Learning und Graph Neural Networks für das automatisierte Kreuzungsmanagement im Mischverkehr vorgestellt. In simulativen Experimenten zeigt der Ansatz gegenüber einer regelbasierten Baseline einen deutlichen Gewinn an Effizienz. Zudem werden Experimente mit einem realen Versuchsträger auf einer Erprobungsbahn vorgestellt.

Schlüsselwörter: Kooperative Manöverplanung, Reinforcement Learning, Graph Neural Networks, vernetztes automatisiertes Fahren

1 Einleitung

Der innerstädtische Verkehr leidet vielerorts unter Störungen und Ineffizienzen aufgrund von stetig steigendem Fahrzeugaufkommen, dem klassische Verkehrsmanagementsysteme nicht gewachsen sind. Insbesondere kleinere Kreuzungen werden in der Regel durch statische Vorrangregeln koordiniert, sodass von einer Nebenstraße kommende Fahrzeuge Vorrang gewähren müssen. Darüber hinaus verdecken Gebäude und andere Verkehrsteilnehmer häufig die Sicht auf den kreuzenden Verkehr sowohl für menschliche Fahrer als auch fahrzeugebundene Sensorsysteme.

Der zunehmende Einsatz von vernetzten Fahrzeugen (VFs) und vernetzten automatisierten Fahrzeugen (VAFs) eröffnet neue Optionen, die Verkehrseffizienz zu verbessern. Diese Fahrzeuge sind über eine drahtlose Kommunikationsschnittstelle vernetzt und können neben der Ankündigung ihrer Präsenz auch Perzeptionsdaten teilen. Durch die zunehmende Verfügbarkeit von Edge-Computing-Ressourcen kann darüber hinaus ein lokales Umfeldmodell, beispielsweise einer Kreuzung und deren Umgebung, an zentraler Stelle vorbehalten werden. Ein solches Umfeldmodell kann durch einen Edge-Server an VFs im

^{*}Robert Bosch GmbH, Corporate Research, D-71272 Renningen (E-Mail: {marvin.klimke, benjamin.voelz}@de.bosch.com).

[†]Institut für Mess-, Regel- und Mikrotechnik – Universität Ulm, Albert-Einstein-Allee 41, D-89081 Ulm (E-Mail: michael.buchholz@uni-ulm.de).

Betriebsbereich verteilt werden. Diese zusätzlichen Informationen können zur Verbesserung der lokalen Planung auf einem VAF verwendet werden.

In dem öffentlich geförderten Projekt MEC-View wurde das Potential des vernetzten automatisierten Fahrens an einer als Pilotanlage ausgestatteten suburbanen Dreiwegekreuzung in Ulm-Lehr erforscht [1]. Die Sicht auf die vorfahrtsberechtigten Straße ist für Fahrzeuge auf der Nebenstraße durch Bebauung stark eingeschränkt. Nur mit Unterstützung durch das serverseitige Umfeldmodell ist es einem automatisierten Fahrzeug möglich, sich ohne starke Verzögerung in eine existierende Lücke im priorisierten Verkehr einzufädeln.

Die vorliegende Arbeit baut auf diesem Ansatz auf und untersucht das Potential einer zentralisierten, kooperativen Manöverplanung durch den Einsatz von Reinforcement Learning (RL). Auf Basis des serverseitigen Umfeldmodells wird ein gemeinsamer Handlungsplan für alle VAFs in der Verkehrsszene abgeleitet, der den Fahrzeugen als Handlungsempfehlung übermittelt wird. Durch die Einbeziehung der VAFs auf der Hauptstraße werden explizite Abweichungen von den Vorrangregeln möglich. Das Einfädeln eines VAFs von der Nebenstraße kann beispielsweise durch proaktives Abbremsen eines VAFs auf der Hauptstraße ermöglicht werden. Auf öffentlichen Straßen wird die Durchdringung mit VFs und VAFs in absehbarer Zeit nicht annähernd 100% erreichen. Stattdessen werden menschliche Fahrer und automatisierte Fahrzeuge im Mischverkehr interagieren. Daher müssen in der kooperativen Manöverplanung alle Fahrzeuge berücksichtigt werden, einschließlich solcher, die nicht beeinflusst werden können.

Im vorliegenden Beitrag wird in Kapitel 2 zunächst das RL-Modell und das Graph Neural Network (GNN) für die kooperative Manöverplanung im Mischverkehr aus [2] vorgestellt. Anschließend werden Ergebnisse der simulativen Evaluation im Vergleich mit einer regelbasierten Baseline präsentiert (Kap. 3). Kapitel 4 behandelt die Anbindung des Lernmodells an einen Bewegungsplanungsalgorithmus und zeigt Experimente unter Verwendung eines realen Versuchsträgers. Die vorliegende Arbeit wird in Kapitel 5 zusammengefasst.

2 Reinforcement Learning Modell für Kooperative Manöverplanung

Dieses Kapitel behandelt das vorgeschlagene Lernmodell (Abschnitt 2.1), die Graph-basierende Eingaberepräsentation (Abschnitt 2.2), die Netzarchitektur (Abschnitt 2.3) sowie die Gestaltung der Reward-Funktion (Abschnitt 2.4).

2.1 Lernmodell

Die kooperative Planung über mehrere Fahrzeuge wird als Mehragenten-RL-Problem aufgefasst. Aufgrund der zentralisierten Planung kann das Problem als einzelner, teilweise beobachtbarer Markov-Entscheidungsprozess (POMDP) formuliert werden:

$$(S, A, T, R, \Omega, O). \tag{1}$$

Die Menge der Zustände S beinhaltet alle erreichbaren Simulatorzustände einschließlich vollständiger Zustandsinformation aller automatisierter und manuell gefahrener Fahrzeuge. Während der Aktionsraum durch die Menge A definiert wird, ist die bedingte

Wahrscheinlichkeit eines Zustandswechsels von $s \in S$ nach $s' \in S$ unter Anwendung von Aktion $a \in A$ durch $T(s' | s, a)$ gegeben. Da die Reaktion menschlicher Fahrer nichtdeterministisch ist, sind auch die Zustandsübergänge nichtdeterministisch. $R : S \times A \rightarrow \mathbb{R}$ beschreibt die Reward-Funktion, die eine gewählte Aktion in einem gegebenem Zustand in Form eines Skalars bewertet. Da der Großteil des abstrakten Verkehrszustandes S für den kooperativen Planer nicht beobachtbar ist, wird eine reduzierte Beobachtungsmenge Ω definiert. Die Wahrscheinlichkeit, dass ein Zustand $s \in S$ in der Beobachtung $\omega \in \Omega$ resultiert, wird durch $O(\omega | s)$ angegeben. Die Abbildung ist nicht injektiv, was sich beispielsweise dann äußert, wenn der Fahrer eines nicht vernetzten Fahrzeugs seine Manöverintention ohne extern erkennbaren Hinweis ändert. Die Dimensionalität der Zustands- und Aktionsräume hängt von der aktuellen Anzahl (steuerbarer) Fahrzeuge ab und kann im Laufe der Zeit variieren.

2.2 Graph-basierende Eingaberepräsentation

Eine geeignete Eingaberepräsentation ist elementar für den erfolgreichen Einsatz von künstlichen neuronalen Netzen. Für die Eingaberepräsentation bei der kooperativen Manöverplanung sind dabei folgende drei zentralen Anforderungen zu berücksichtigen:

- Invarianz bezüglich der Anzahl der Fahrzeuge in der Szene,
- Permutationsinvarianz hinsichtlich der Eingabeknoten,
- Permutationsäquivarianz bezüglich der Ein- und Ausgabeknoten.

Einfache tabellarische Repräsentationen verletzen bereits die Invarianzeigenschaften. Die Beschränkungen von Eingängen mit fester Größe werden in [3] genauer diskutiert. Ein gerendertes Rasterbild der Verkehrsszene, wie es häufig mit Faltungsnetzen verwendet wird, erfüllt die Invarianzeigenschaften, aber erfordert typischerweise ein Ego-Fahrzeug, auf das es zentriert wird [4]. Dieser Prozess müsste für jedes beteiligte Fahrzeug wiederholt werden, was eine Anwendung auf komplexe Verkehrsszenen rechnerisch kaum durchführbar macht. Permutationsäquivarianz beschreibt die Tatsache, dass die abgeleiteten Ausgaben für jedes Fahrzeug unabhängig von der Reihenfolge in der Eingaberepräsentation sind. In der vorliegenden Arbeit wird daher eine flexible Graph-basierende Repräsentation vorge schlagen.

Folglich wird eine Beobachtung beschrieben durch $\omega = (V, E, U)$, wobei jedes Fahrzeug auf einen Knoten $\nu \in V$ abgebildet wird, die Menge der Kanten durch E und die Menge der Kantentypen durch U bezeichnet wird. Die verfügbaren Kantentypen ergeben sich aus dem kartesischen Produkt aus den Dimensionen Konfliktrelation und Automatisierungstyp der beteiligten Fahrzeuge, wie in Abbildung 1 dargestellt. Formal ist eine Kante definiert als $(\nu_i, \nu_j, g_{ij}, r) \in E$, wobei die Quell- und Zielknoten als ν_i respektive ν_j und der Kantentyp als r bezeichnet werden. Nach [2] werden die Position, Geschwindigkeit, gemessene Beschleunigung sowie ein Indikator, ob es sich um ein VAF handelt, als einseitige Knoteneigenschaften $\mathbf{h}^{(0)}$ encodiert. Die Kanteneigenschaften $\mathbf{g}_{ij}^{(0)}$ enthalten eine Distanzmetrik sowie eine relative Peilung und die Vorrangrelation zweier Fahrzeuge.

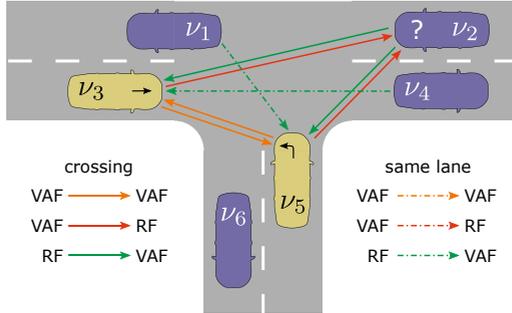


Abbildung 1: Die Graph-basierende Eingaberepräsentation gemäß [2]. Die Route eines VAF (gelb) ist durch einen Pfeil gekennzeichnet. Aufgrund der unbekannt Route des regulären Fahrzeugs ν_2 (RF, blau, gekennzeichnet mit '?') teilt es Kanten mit beiden VAFs, obwohl der Konflikt mit ν_3 nur bei einem Abbiegen von ν_2 auftreten wird.

2.3 Architektur des Graph Neural Networks

Für das Training des GNN wird der RL-Algorithmus TD3 aus [5] verwendet, der zur Familie der Actor-Critic-Methoden für kontinuierliche Aktionsräume gehört. Da sich das Actor- und das Critic-Netz nur unwesentlich auf der Ausgabeseite unterscheiden, wird im Folgenden lediglich die Actor-Netzarchitektur im Detail vorgestellt.

In diesem Beitrag wird eine GNN-Architektur, bestehend aus Schichten vom Typ Relational Graph Convolutional Network (RGCN) [6] sowie Graph Attention Network (GAT) [7], vorgeschlagen. Um neben Knoteneigenschaften auch Kanteneigenschaften verarbeiten zu können, wird in den RGCN-Schichten eine modifizierte Update-Vorschrift

$$\mathbf{h}_i^{(2)} = \sigma \left(\sum_{r \in U} \max_{j \in \mathcal{N}_i^r} \mathbf{W}_r^{(1)} [\mathbf{h}_j^{(1)}, \mathbf{g}_{ji}^{(1)}] + \mathbf{W}_0^{(1)} \mathbf{h}_i^{(1)} \right) \quad (2)$$

verwendet. Die unterschiedlichen Kantentypen, wie in Abschnitt 2.2 eingeführt, korrespondieren zu individuell gelernten Gewichtsmatrizen \mathbf{W}_r je Kantentyp r . Dies ermöglicht dem GNN die fundamentalen Unterschiede in der Interaktion automatisierter und nicht-automatisierter Fahrzeuge auf sinnvolle Aktionen zu übertragen.

Abbildung 2 stellt die Netzarchitektur schematisch dar. Sowohl die Knoten- als auch die Kanteneigenschaften werden zunächst durch vollständig verbundene Schichten encodiert. Anschließend werden die latenten Knoteneigenschaften durch drei GNN-Schichten für den sogenannten Nachrichtenaustausch geleitet, welche die Kanteneigenschaften jeweils als zusätzliche Eingabe erhalten. Alle GNN-Schichten erfüllen die Vorschrift $\text{conv} : V^n \times E^m \rightarrow V^n$, wobei n die Anzahl der Fahrzeuge (Knoten) und m die Anzahl der Konflikte (Kanten) bezeichnet. Die Kanteneigenschaften werden durch das Netz nicht aktualisiert, da auf den Kanten keine Ausgabe zu inferieren ist. Die Benutzung alternierender Schichttypen zeigte bessere Ergebnisse als ein reines RGCN- oder GAT-Netz. Damit wird der Attention-Mechanismus der GAT-Schicht mit expliziten Kanteneigenschaften und Kantentypen verbunden. Insbesondere nutzen die GAT-Schichten selbst die Kantentypen nicht und verwenden die Kanteneigenschaften lediglich für die Attention-Gewichte,

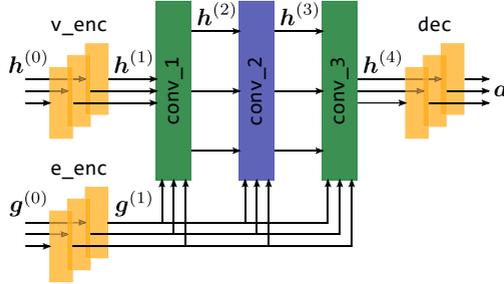


Abbildung 2: Die GNN-Architektur des Actor-Netzes aus [2]. Knoteneigenschaften $\mathbf{h}^{(0)}$ und Kanteneigenschaften $\mathbf{g}^{(0)}$ werden auf eine gemeinsame Aktion \mathbf{a} abgebildet. Die um Unterstützung für Kanteneigenschaften erweiterten RGCN-Schichten sind in grün abgebildet, die GAT-Schicht in blau und die vollständig verbundenen Schichten in gelb.

aber nicht für das Update der Knotenfeatures. Die latenten Knoteneigenschaften $\mathbf{h}^{(4)}$ werden dann durch weitere vollständig verbundene Schichten geleitet, um eine gemeinsame Aktion für alle Fahrzeuge in der Szene abzuleiten. Alle verdeckten Schichten werden mit einer ReLU-Aktivierungsfunktion versehen und alle vollständig verbundenen Schichten teilen ihre Gewichte über Knoten beziehungsweise Kanten. Im Gegensatz zum Actor-Netz werden im Critic-Netz die latenten Knoteneigenschaften $\mathbf{h}^{(4)}$ zu einem einzelnen Vektor aggregiert, aus dem sodann eine Schätzung des Q -Wertes abgeleitet wird.

2.4 Reward-Funktion

Die Reward-Funktion für die Steuerung des RL-Trainings ist als gewichtete Summe

$$R = \sum_{k \in \mathcal{R}} w_k R_k \quad (3)$$

definiert, wobei die Gewichte mit w_k und die Menge der Reward-Komponenten mit \mathcal{R} bezeichnet werden.

Die treibende positive Reward-Komponente begünstigt hohe mittlere Fahrgeschwindigkeiten mit einem linearen Anstieg bis zur gültigen Geschwindigkeitsbegrenzung. Ein triviales kollisionsfreies Verhalten wird erzielt, indem alle Fahrzeuge angehalten werden. Da dies keine zulässige Lösung für die kooperative Planung ist, wird dieser Fall durch einen Stillstand-Abschlag im Training vermieden. Um dem Modell das Einhalten sinnvoller Sicherheitsabstände beizubringen, wird eine geschwindigkeitsabhängige Distanzmetrik als weiterer Abschlag eingeführt. Simulationen im Mischverkehr zeigten ein prinzipiell sehr konservatives Verhalten der VAF, die potentiell jedes RF vorfahren lassen, selbst wenn das VAF Vorrang hat. Um dieses Verhalten zu korrigieren, wird eine weitere Reward-Komponente eingeführt, welche einen Abschlag auslöst, wenn ein VAF ohne erkennbaren Grund weit entfernt vom Kreuzungseingang zum Stehen kommt. Abschließend wird jede Kollision mit einem negativen Reward belegt, damit das Modell implizit Kollisionsvermeidung erlernt. Darüber hinaus wird die Trainingsepisode im Kollisionsfall abgebrochen.

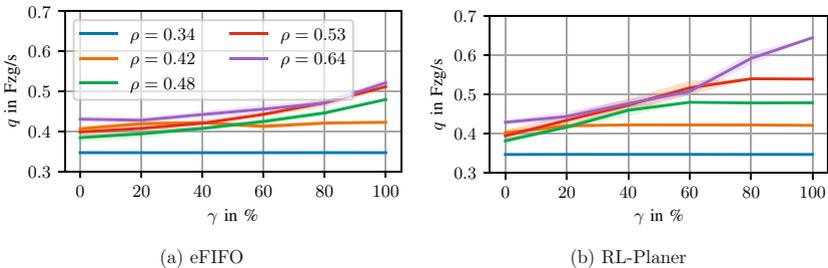


Abbildung 3: Erreichter Fahrzeugdurchsatz über variierenden Anteil an VAFs bei Einsatz der eFIFO-Vergleichsbasis und des RL-Planers. Jede Linienfarbe korrespondiert zu einem Szenario mit zunehmender Verkehrsnachfrage ρ in Fzg/s. Die schattierte Fläche markiert die Standardabweichung.

3 Simulative Evaluation

Das Training und die simulative Auswertung des RL-Planers erfolgen in der Open-Source-Simulationsumgebung Highway-env [8], die geringfügig für die Anwendung in kooperativer Manöverplanung angepasst wurde. Das Verhalten von Menschen gefahrener Fahrzeuge wird durch ein Fahrermodell, wie das Intelligent Driver Model (IDM) [9], simuliert. In der Evaluation wird abweichend ein erweitertes IDM [10] verwendet, welches nicht-deterministisches Verhalten abbildet. Darüber hinaus wurde die Vorranglogik in Highway-env verbessert, um auch bei hohen Verkehrsdichten einsetzbar zu sein.

In diesem Beitrag wird ein erweitertes First-In-First-Out-Schema (eFIFO) als Vergleichsbasis für die kooperative Manöverplanung verwendet. VAFs werden auf Basis ihrer Distanz zur Kreuzung priorisiert und Vorrang von RFs wird als Randbedingung berücksichtigt. Darüber hinaus können auch konfliktfreie Pfade zeitgleich befahren werden, was einen erheblichen Performance-Gewinn liefert.

Die kooperativen Planungsansätze wurden in fünf Szenarien mit unterschiedlicher Verkehrsnachfrage ρ an einer Vierwegekreuzung evaluiert. Dabei wurde jedes Szenario mit variierendem Anteil an VAFs im Gesamtverkehr γ mehrfach simuliert. Der tatsächlich erreichte Durchsatz wird durch die *Flow Rate* q erfasst. Wie in Abbildung 3 ersichtlich ist, erreicht der RL-Planer gegenüber der Vergleichsbasis einen deutlich höheren Fahrzeugdurchsatz, auch für geringe Ausstattungsdaten.

Die Generalisierung des Ansatzes auf andere Kreuzungen, die nicht im Training verwendet wurden, findet sich für vollständig automatisierten Verkehr in [11].

4 Experimente mit Bewegungsplanung

Das zentralisierte Planungsmodul im vernetzten automatisierten Fahren erstellt ein Verhaltensplan für alle beteiligten Fahrzeuge, sendet jedoch keine direkten Steuerkommandos an die VAFs. Stattdessen führt jedes VAF lokal einen Bewegungsplanungsalgorithmus aus, welcher die kooperativen Manöver in eine fahrbare Trajektorie umsetzt. Dafür ist ein

Vorausplanungshorizont von zumindest mehreren Sekunden erforderlich. Das RL-Modell liefert jedoch nur eine instantane Aktion für den nächsten Zeitschritt, bevor eine neue Zustandsbeobachtung eintrifft.

In der vorliegenden Arbeit wird der Einsatz einer serverseitig integrierten Simulationsumgebung nach [12] vorgeschlagen, um die Lücke zwischen iterativer RL-Planung und Vorausplanung einer Trajektorie zu schließen. Um ein kooperatives Manöver zu planen, wird der aktuelle Zustand des serverseitigen Umfeldmodells \mathcal{E} als Startzustand in der Simulationsumgebung abgebildet, wie in Abbildung 4 dargestellt. In der Folge wird die Entwicklung der Verkehrsszene in die Zukunft präzidiert, wobei das RL-Modell in jedem Zeitschritt abgefragt wird, um das Verhalten der VAFs zu bestimmen. Die regulären Fahrzeuge in der Szene werden unter Annahme eines Fahrermodells, wie dem IDM, mitpräzidiert. Sobald alle Fahrzeuge in der Simulation ihre jeweilige Zielseite der Kreuzung erreicht haben, endet der Simulationslauf. Sodann wird aus der simulierten Trajektorie je Fahrzeug eine abstrakte Manöverdarstellung \mathcal{O} abgeleitet, die den VAFs übermittelt wird.

Die in diesem Beitrag verwendete Darstellung basiert auf dem Vorschlag zur *Maneuver Coordination Message* aus [13]. Während die laterale Führung anhand einer Spurmittellinie erfolgt, werden die Fahrzeuge longitudinal durch zusätzliche Manöverwegpunkte beeinflusst. Neben einer Position ist ein Manöverwegpunkt durch Einschränkungen des Zeitpunktes δt_{MWP} und der Geschwindigkeit bei der Überquerung gekennzeichnet. Jedes VAF erhält in einem kooperativen Manöver einen Manöverwegpunkt am Kreuzungseingang. Durch Anpassung der Zeit- beziehungsweise Geschwindigkeitsvorgaben wird die Überquerung der Kreuzung durch alle beteiligten VAF koordiniert.

In Vorbereitung auf die Fahrtests mit mehreren VAFs im Realverkehr wurde der vorgeschlagene Ansatz zunächst in einem automatisierten Fahrzeug auf einer Erprobungsbahn getestet. Das Experiment folgt dem Prinzip einer *Vehicle-in-the-Loop*-Simulation, bei der das automatisierte Fahrzeug in der Realität bewegt wird und zusätzliche Objektfahrzeuge simuliert werden können. Im untersuchten Manöver quert das automatisierte Fahrzeug eine Kreuzung, an der es einem Objektfahrzeug Vorrang gewähren muss. Dabei werden zwei alternative Überplanungskonzepte evaluiert. Zunächst wird in einem idealisierten Modus das Manöver *einmalig* geplant und fortan unverändert ausgeführt. Insbesondere im Mischverkehr, unter Anwesenheit nicht beeinflussbarer Fahrzeuge, kann sich die tatsächliche Entwicklung des Manövers von dem ursprünglichen Plan unterscheiden. Um diese Fälle abzufangen, wird eine *zyklische* Neuplanung vorgeschlagen, wobei das Manöver in diesen Experimenten mit

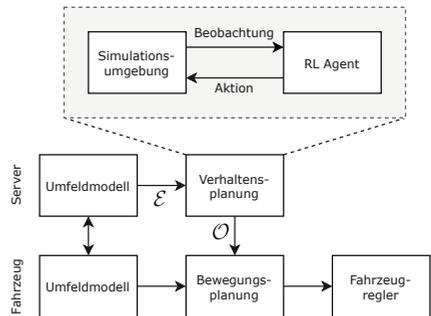
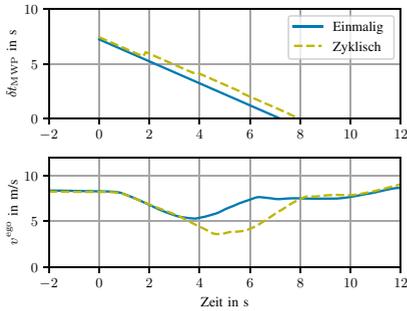
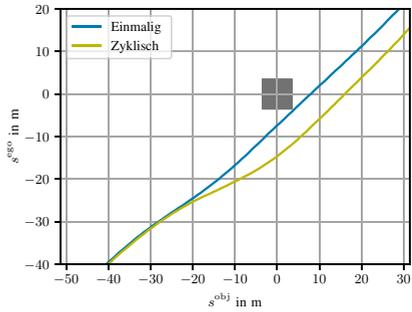


Abbildung 4: Verwendung einer serverseitig integrierten Simulationsumgebung für die kooperative Manöverplanung nach [12]. Aus dem serverseitigen Umfeldmodell \mathcal{E} wird eine abstrakte Manöverdarstellung \mathcal{O} für jedes VAF abgeleitet.



(a) Zeitvorgabe und Geschwindigkeitsprofil.



(b) Relative Position des Ego-Fahrzeugs gegenüber dem Objektfahrzeug.

Abbildung 5: Exemplarische Aufzeichnung eines Manövers bei einmaliger und zyklischer Planung aus [12]. Bei zyklischer Neuplanung wird die Überquerung der Kreuzung mehrfach verzögert. Das graue Rechteck stellt den Bereich dar, in dem es zur Kollision kommt.

einer Periodenlänge von 2 s neu geplant wird. Dieser Wert bezieht sich lediglich auf die serverseitige RL-basierte Verhaltensplanung, während die unterlagerte Bewegungsplanung im Fahrzeug die Trajektorie wesentlich hochfrequenter plant.

Eine exemplarische Aufzeichnung eines solchen Manövers ist in Abbildung 5 visualisiert. In diesem Fall fährt das Objektfahrzeug langsamer als die in der Karte hinterlegte Maximalgeschwindigkeit, die in der Verhaltensplanung als Zielgeschwindigkeit angenommen wird. Bei zyklischer Neuplanung wird die Zeitvorgabe δt_{MWP} des Manöverwegpunkts daher mehrfach verschoben, was sich durch eine stärkere Verzögerung auch im Geschwindigkeitsprofil bemerkbar macht. Bei Betrachtung von Abbildung 5(b) wird deutlich, dass diese Verzögerung erforderlich ist, um einen hinreichenden Sicherheitsabstand zum Objektfahrzeug zu wahren. Unter der Annahme von rechteckigen Fahrzeugausmaßen von $5\text{ m} \times 2\text{ m}$ kollidieren die Fahrzeuge, wenn die Trajektorie den grauen Bereich durchschreitet. In diesem Beispiel verbleibt bei einmaliger Planung keinerlei Abstand zwischen den Fahrzeugen, was in der Praxis nicht akzeptabel ist. Nur durch zyklische Neuplanung wird die geringere Geschwindigkeit des Objektfahrzeugs korrekt antizipiert und ein hinreichender Sicherheitsabstand eingehalten.

Für eine genauere Untersuchung dieses Effekts wurden 40 komplexere Szenarien mit bis zu sechs VAFs simulativ mit einmaliger und zyklischer Planung evaluiert. Dabei wird

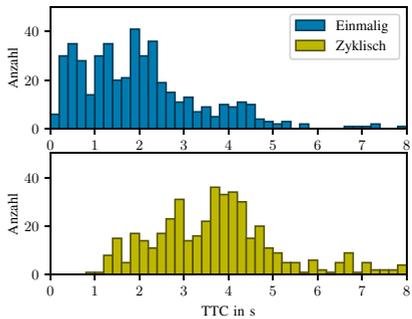


Abbildung 6: Verteilung der TTC über 40 simulierte Szenarien in einmaliger und zyklischer Planung.

eine verallgemeinerte Time-to-Collision (TTC) verwendet, die auch für kreuzende Pfade anwendbar ist und in [14] vorgeschlagen wurde. Abbildung 6 zeigt die Verteilung der TTC für beide Planungsmodi als Histogramm. Offensichtlich treten bei einmaliger Planung wesentlich häufiger kritische TTC von unter 1 s auf. In der Simulation mit einmaliger Planung endeten sechs Szenarien in einer Kollision, die durch eine TTC von 0 s im Plot wiederzufinden sind. Diese simulativen Experimente wurden im vollständig automatisierten Verkehr durchgeführt und es ist zu erwarten, dass sich der Effekt im Mischverkehr noch stärker ausgeprägt. Auf Basis dieser initialen Experimente auf der Erprobungsbahn und im Simulator erscheint der zyklische Planungsmodus prinzipiell geeignet, um die RL-basierte Verhaltensplanung an die fahrzeugseitige Bewegungsplanung anzubinden.

5 Zusammenfassung

In diesem Beitrag wurde ein auf maschinellem Lernen basierender Ansatz für die Planung kooperativer Manöver im urbanen Mischverkehr vorgestellt. Durch Verwendung einer Graph-basierten Eingaberepräsentation können komplexe Verkehrsszenen an Kreuzungen effizient encodiert werden. Aufgrund fehlender Trainingsdaten für kooperative Manöver wird das GNN durch Reinforcement Learning auf einer geeigneten Reward-Funktion trainiert. Auswertungen in der Simulationsumgebung Highway-env zeigen eine deutliche Verbesserung des Verkehrsdurchsatzes auch bei geringen Ausstattungsraten im Mischverkehr. Auf einer Erprobungsbahn wurden erste Experimente in einem Versuchsträger mit dedizierter Bewegungsplanung durchgeführt und gezeigt, dass unter zyklischer Neuplanung ausgewählte Kreuzungsszenarien beherrscht werden.

Danksagung

Teile dieser Arbeit wurden im Rahmen des Projekts LUKAS (Förderkennzeichen 19A2000 4A und 19A20004F) durchgeführt, welches durch das Programm “Hoch- und vollautomatisiertes Fahren in anspruchsvollen Fahrsituationen” des Bundesministeriums für Wirtschaft und Klimaschutz finanziert wird.

Literatur

- [1] M. Buchholz, J. Müller, M. Herrmann, J. Strohecker, B. Völz, M. Maier, J. Paczia, O. Stein, H. Rehborn, and R.-W. Henn, “Handling Occlusions in Automated Driving Using a Multiaccess Edge Computing Server-Based Environment Model From Infrastructure Sensors,” *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 3, pp. 106–120, 2022.
- [2] M. Klimke, B. Völz, and M. Buchholz, “Automatic Intersection Management in Mixed Traffic Using Reinforcement Learning and Graph Neural Networks,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, to appear.
- [3] M. Huegle, G. Kalweit, B. Mirchevska, M. Werling, and J. Boedecker, “Dynamic Input for Deep Reinforcement Learning in Autonomous Driving,” in *2019 IEEE/RSJ*

- International Conference on Intelligent Robots and Systems (IROS)*, pp. 7566–7573, 2019.
- [4] J. Chen, B. Yuan, and M. Tomizuka, “Deep Imitation Learning for Autonomous Driving in Generic Urban Scenarios with Enhanced Safety,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2884–2890, 2019.
- [5] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing Function Approximation Error in Actor-Critic Methods,” in *Proceedings of the 35th International Conference on Machine Learning (J. Dy and A. Krause, eds.)*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596, 2018.
- [6] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling Relational Data with Graph Convolutional Networks,” in *The Semantic Web (A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tor-dai, and M. Alam, eds.)*, vol. 10843, pp. 593–607, Springer International Publishing, 2018.
- [7] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph Attention Networks,” in *International Conference on Learning Representations*, 2018.
- [8] E. Leurent, “An Environment for Autonomous Driving Decision-Making,” 2018.
- [9] M. Treiber, A. Hennecke, and D. Helbing, “Congested Traffic States in Empirical Observations and Microscopic Simulations,” *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.
- [10] D. Salles, S. Kaufmann, and H.-C. Reuss, “Extending the Intelligent Driver Model in SUMO and Verifying the Drive Off Trajectories with Aerial Measurements,” in *SUMO User Conference*, 2020.
- [11] M. Klimke, J. Gerigk, B. Völz, and M. Buchholz, “An Enhanced Graph Representation for Machine Learning Based Automatic Intersection Management,” in *2022 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 523–530, 2022.
- [12] M. Klimke, B. Völz, and M. Buchholz, “Integration of Reinforcement Learning Based Behavior Planning With Sampling Based Motion Planning for Automated Driving,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, to appear.
- [13] M. B. Mertens, J. Müller, R. Dehler, M. Klimke, M. Maier, S. Gherekhloo, B. Völz, R.-W. Henn, and M. Buchholz, “An Extended Maneuver Coordination Protocol with Support for Urban Scenarios and Mixed Traffic,” in *2021 IEEE Vehicular Networking Conference (VNC)*, pp. 32–35, 2021.
- [14] F. Jimenez, J. E. Naranjo, and F. Garcia, “An Improved Method to Calculate the Time-to-Collision of Two Vehicles,” *International Journal of Intelligent Transportation Systems Research*, vol. 11, no. 1, pp. 34–42, 2013.