



Hoch zuverlässige Elektroniksysteme für intelligente Fahrzeuge
Programmable Systems for Intelligence in Automobiles
- PRYSTINE -



Abschlussbericht
Bayerische Motoren Werke Aktiengesellschaft



Rolls-Royce
Motor Cars Limited

öffentlich

Zuwendungsempfänger	BMW Group
Förderkennzeichen	16ESE0331
Vorhabensbezeichnung	PRYSTINE: Hoch zuverlässige Elektroniksysteme für intelligente Fahrzeuge
Laufzeit des Vorhabens	16.06.2018 - 30.04.2021
Datum	30.09.2021

 
i.A. Kathrin Seibold | i.A. Viktor Rack

GEFÖRDERT VOM



**Bundesministerium
für Bildung
und Forschung**

Inhalt

1	Kurzdarstellung	3
1.1	Aufgabenstellung	3
1.2	Voraussetzungen der Durchführung	4
1.3	Planung und Ablauf des Vorhabens.....	6
1.4	Ausgangsstand von Wissenschaft und Lösungsansatz	8
1.4.1	Stand der Wissenschaft und Technik.....	8
1.4.2	Neuheit des Lösungsansatzes.....	8
1.5	Zusammenarbeit mit anderen Stellen	9
2	Eingehende Darstellung.....	9
2.1	Verwendung der Zuwendung und erzielte Ergebnisse.....	9
2.1.1	WP1 – Anforderungen und Spezifikation	9
2.1.2	WP2 – System Level Design	14
2.2	Erfolgte und geplante Veröffentlichungen	41
3	Abbildungsverzeichnis	42
4	Tabellenverzeichnis.....	43
5	Literaturverzeichnis.....	44
Anhang A	Zeitplanung Teilvorhaben.....	45

1 Kurzdarstellung

1.1 Aufgabenstellung

Im kommenden Jahrzehnt wird sich die Automobilindustrie stärker verändern als in den vergangenen 30 Jahren. Schon heute stehen wir an der Schwelle zum hochautomatisierten Fahren. Die steigende Automatisierung von Fahrzeugen – mit dem letztlichen Ziel des vollständig autonomen Fahrens – wurde dabei als einer der Hauptfaktoren für die Bewältigung der großen gesellschaftlichen Herausforderungen "Individuelle Mobilität" und "Energieeffizienz" identifiziert. Das Beherrschen von hochautomatisierten Fahrfunktionen ist dabei ein entscheidender Schritt. Eine der großen Herausforderungen, um hochautomatisiertes Fahren erfolgreich zu realisieren, ist der Schritt von SAE Level-2 (Teilautomatisierung) zu SAE Levels-3 (Bedingte Automatisierung) und darüber. Auf Stufe 3, bleibt der Fahrer als Ausweichoption im Falle eines Fehlers in der Automatisierung verfügbar oder greift ein, wenn die Assistenzfunktion ihre Betriebsgrenzen erreicht. Auf höheren Levels kann nicht darauf vertraut werden, dass der Fahrer rechtzeitig und angemessen eingreifen kann; folglich muss die Automatisierung hier in der Lage sein, selbst sicherheitskritische Situationen zu bewältigen. Ausfallsicheres Verhalten ist dabei in der Erkennung, Planung und Ausführung der automatisierten Fahrfunktionen essentiell. Das Gesamtprojekt „Hoch zuverlässige Elektroniksysteme für intelligente Fahrzeuge“ zielte darauf, eine ausfallsichere Urban Surround perzeptiON (FUSION)-Funktion zu realisieren, die auf einer robusten Kamera-, Radar- und LiDAR-Sensorfusion sowie auf Steuerungsfunktionen basiert, um ein sicheres automatisiertes Fahren in städtischer und ländlicher Umgebung zu gewährleisten.

Im Rahmen des Gesamtvorhabens PRYSTINE lag der Schwerpunkt des Teilvorhabens der BMW Group auf der ausfallsicheren Umfelderkennung für hochautomatisierte Fahrzeuge. Fokus der Arbeiten war dabei die Definition von Anforderungen und Spezifikationen aus Sicht der Automobilindustrie. Ebenso gehörte das System Level Design, die simulative Analyse der eingebetteten Sensor- und Steuerungskomponenten, die E/E-Architektur, sowie die Auslegung und Bewertung des Gesamtsensorverbunds. Innerhalb von nur wenigen Jahren wird vernetztes und automatisiertes Fahren die individuelle Mobilität revolutionieren, indem es sicherere, sauberere und stärker nutzerorientierte Dienste anbietet, die sich gegenseitig auf lange Sicht verstärken. In diesem Zusammenhang entstehen E-Mobilität, autonomes Fahren und neue Geschäftsmodelle, insbesondere in den Bereichen digitaler Dienste, in denen die Konnektivität und Verfügbarkeit von Echtzeitdaten für die BMW Group von zentraler Bedeutung ist.

Aus den oben genannten Zielen ergaben sich folgende Forschungsschwerpunkte:

- **Definition von Anforderungen und Spezifikation aus Sicht der Automobilindustrie**
Der Fokus des Projekts Prystine ist die Umsetzung einer ausfallsicheren urbanen Umfelderkennung, um automatisierte Mobilitätsszenarien so zu verwirklichen, dass diese in einem unstrukturierten urbanen Umfeld sowohl unfallfrei als auch ohne Verletzung aller Verkehrsteilnehmer möglich sind. Insbesondere sollen mit den entwickelten Systemen die spezifischen Bedürfnisse der am meisten gefährdeten Verkehrsteilnehmern, wie z.B. Fußgänger, Fahrradfahrer, Kinder und vor allem körperlich beeinträchtigte Personen und ältere Menschen, berücksichtigt werden, um ihre Mobilität und Sicherheit im Verkehr zu erhöhen.
Um dieses Ziel erreichen zu können, war es notwendig ein klares und organisiertes Anforderungsmanagement zu etablieren. Alle beteiligten Partner im Arbeitspaket 1

mussten mit den Anforderungen systematisch umgehen, d.h. Anforderungen strukturiert erfassen, dokumentieren, überprüfen, untereinander abstimmen und verfeinern.

- **Simulation eines LiDAR Einzelsensors basierend auf probabilistischen Sensormodellen**

Inhalt der Aufgabe war die Simulation eines einzelnen Lidarsensors auf Basis eines probabilistischen Sensormodells. Die durchgeführte Simulation wurde anschaulich beschrieben, sowie die Simulationsdaten inklusive der Konfiguration allen Projektpartnern bereitgestellt, sodass alle Partner im Projekt dies in ihrer Arbeit ebenso umsetzen können.

- **System Design und Fahrzeugbustopologie für jedes Automatisierungslevel mit einer bereichs- und zonenorientierten E/E Architektur am Beispiel von Sensorik im Exterieur**

Wesentlicher Beitrag dieser Aufgabe war die Beschreibung eines neuartigen Systemdesigns und einer Bustopologie in Abhängigkeit des jeweiligen Automatisierungslevels, die in einzelne Zonen im Fahrzeug aufgeteilt wird und damit eine skalierbare Bustopologie, sowie E/E Architektur auf Basis unterschiedlicher Anforderungen in nur einem einzelnen System ermöglicht.

- **Methode zur Evaluation beliebiger Gesamtsensorkonfigurationen**

In dieser Aufgabe wird das System Design in der frühen Phase der Entwicklung dadurch adressiert, dass unterschiedliche Sensortypen in einem Gesamtkontext gleichzeitig simuliert werden, um die Umfelderkennung durch dieses Sensorsetup in Summe abzuschätzen. Auf diese Art und Weise können unterschiedliche Sensorkonfiguration in einem Gesamtsystem schon sehr früh evaluiert werden, um in der Folge das beste Sensorsetup mit der jeweiligen Verortung am Fahrzeug, Anzahl, sowie Sensortyp festzulegen.

1.2 Voraussetzungen der Durchführung

Neben der Fokussierung auf elektrische Antriebe, sind hochautomatisierte Fahrzeuge ein anhaltender Trend in der Automobilindustrie, dem selbst Internet- und Technologiekonzerne folgen [1] [2]. Die deutsche Regierung betont die Bedeutung des automatisierten Fahrens für den Wirtschaftsstandort Deutschland auch durch verschiedene Strategiepapiere [3] [4].

Die BMW Group arbeitet seit dem Jahr 2000 am hochautomatisierten Fahren und bietet ihren Kunden bereits heute deutliche Entlastungen durch Assistenz-Funktionen im Fahrzeug. Die nächste Stufe des hochautomatisierten Fahrens wird den Komfort und die Effizienz während der Fahrt weiter signifikant steigern und einen positiven Beitrag zur Erhöhung der Verkehrssicherheit leisten.

Innerhalb von nur wenigen Jahren wird vernetztes und automatisiertes Fahren die individuelle Mobilität revolutionieren, indem es sicherere, sauberere und stärker nutzerorientierte Dienste anbietet, die sich gegenseitig auf lange Sicht verstärken. In diesem Zusammenhang entstehen E-Mobilität, autonomes Fahren und neue Geschäftsmodelle, insbesondere in den Bereichen digitaler Dienste, in denen die Konnektivität und Verfügbarkeit von Echtzeitdaten für die BMW Group von zentraler Bedeutung ist.

System-Architekturen für die Automobilelektronik unterliegen einer schnellen Veränderung. Autonomes Fahren erfordert eine Multisensor-Fusion weg von funktional isolierten Steuergeräten. Connectivity und Infotainment haben das Auto zu einem IT-System mit Cloud-Zugang gemacht – funktional unterstützt durch over-the-Air-Upgrades, sowie wachsende und breite Zugriffsmöglichkeiten auf Kartendienste, Medieninhalte, andere Fahrzeuge und die umliegende Infrastruktur. Sensorfusion und Digitalisierung schaffen Möglichkeiten, die unser Verständnis von Mobilität verändern werden. Das Automobil als Teil der vernetzten Welt wird nicht nur neue Möglichkeiten zur Information und Unterhaltung von Kunden bieten, sondern

auch durch weitere Automatisierung von Komfort- und Assistenz-funktionen den Weg in Richtung autonomer Fahrzeuge bereiten. Diese Innovationen erfordern einen immer schnelleren Anstieg der Elektronik-, Software- und Kommunikationsfähigkeiten. Der nahtlose Einsatz von IT-Standards, die Verfügbarkeit leistungsfähigster Rechenkapazitäten und vernetzte Sensorsysteme werden daher als entscheidend für den Weg in die digitale automobiler Zukunft der BMW Group gesehen.

1.3 Planung und Ablauf des Vorhabens

Für die Durchführung des Vorhabens wurde die Inhalte des Projektes in 8 Work Packages und 10 Supply Chains aufgeteilt, dargestellt in Abbildung 1. In der Tabelle 1 (Work Package 1) und Tabelle 2 (Work Package 2) sind neben den Inhalten der einzelnen Arbeitspakete auch die Leiter der Arbeitspakete festgehalten.

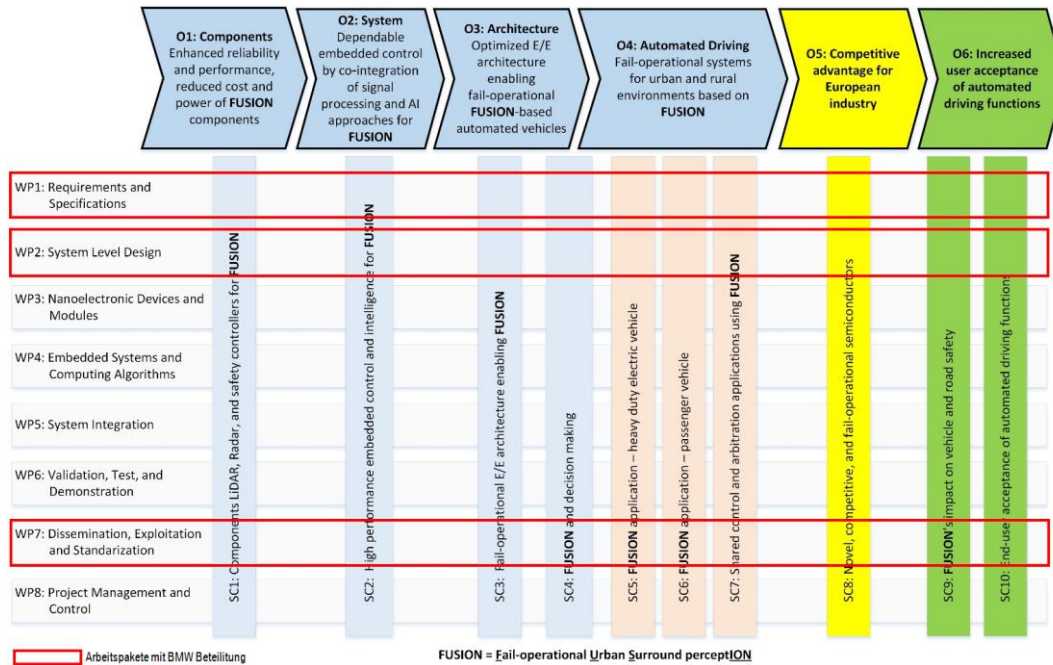


Abbildung 1: Gesamtvorhaben PRYSTINE, gegliedert in Arbeitspakete (WP) und Supply Chains (SC)

Task no.	Task title	Lead partic. short name
T1.1	[SC1] Requirements and specifications for fail-operational sensor components	IFAG
T1.2	[SC1] Requirements and specifications for fail-operational embedded control components	TTTDE
T1.3	[SC2] Requirements and specifications for fail-operational embedded intelligence	GNT
T1.4	[SC3] Requirements and specifications for fail-operational communication and E/E architecture	CISC
T1.5	[SC4] Requirements and specifications for robust and fail-operational environment perception based on sensor fusion	TUE
T1.6	[SC5, SC9] Requirements and specifications for the heavy duty vehicle demonstrator	VIF
T1.7	[SC6, SC9] Requirements and specifications for the passenger vehicle demonstrator	UNIMORE
T1.8	[SC7] Requirements and specifications for shared control and arbitration demonstrator using FUSION	TECNALIA

Tabelle 1: Übersicht Aufgabenpakete in Work Package 1

BMW übernahm die Leitung von WP1, in dem die Anforderungen und Spezifikationen erarbeitet wurden.

In Work Package 1 wurden die Systemanforderungen der OEMs, hier speziell BMW, an PRYSTINE's FUSION Technologie in Spezifikationen für Subsysteme, Module und Komponenten überführt, wobei auch die Restriktionen in der Anwendung berücksichtigt wurden. Für jedes Subsystem, das im Projekt adressiert wurde, war ein involvierter Projektpartner zuständig, die Anforderungen und Spezifikationen in enger Abstimmung mit dem OEM, d.h. BMW, zu definieren. Dies beinhaltete die Anforderungen und Spezifikationen der finalen Lösungen für:

- Komponenten: Radar, LiDAR und Sicherheitscontroller für FUSION
- Integrierte Hochleistungssteuerung und Intelligenz für FUSION
- Ausfallsichere E/E Architektur zur Befähigung von FUSION
- FUSION und Entscheidungsfindung
- FUSION Anwendung – E-LKW
- FUSION Anwendung – PKW
- Geteilte Kontrolle und Kotrollarbitrierung mittels FUSION

Task no.	Task title	Lead partic. Short name
T2.1	[SC1, SC9] System design and simulative analysis of fail-operational sensor components	DICE
T2.2	[SC1, SC9] System design and simulative analysis of fail-operational embedded control components	EPOS
T2.3	[SC2, SC9] System design and simulative analysis of the fail-operational reference embedded intelligence	VID
T2.4	[SC3, SC9] System design and simulative analysis of the fail-operational communication and E/E architecture	CISC
T2.5	[SC4, SC9] System design and simulative analysis of robust and fail-operational environment perception based on sensor fusion	TUE
T2.6	[SC5] System design and simulative analysis of the heavy duty vehicle demonstrator	UTU
T2.7	[SC6] System design and simulative analysis of the passenger vehicle demonstrator	UNIMORE
T2.8	[SC7] System design and simulative analysis of the shared control and arbitration demonstrator using FUSION	TECNALIA

Tabelle 2: Übersicht Aufgabenpakete in Work Package 2

Work Package 2 zielte darauf, ein umfassendes und eindringliches quantitatives Verständnis der Subsysteme zu etablieren, welches in höhere Ebenen des FUSION Systems und Demonstratoren integriert wurde, durch Ausführung, Modellierung und Simulation auf unterschiedlichen Ebenen.

- Auf Komponentenebene wurden die elektrische und thermische Modellierung, sowie Simulation genutzt, um die Parameter der elektrischen und elektronischen Komponenten zu dimensionieren.
- Auf der Systemebene haben die Modellierung und Simulationen jedes einzelnen ausfallsicheren Subsystems u.a. die Modellierung der multiphysischen Kommunikationskanäle beinhaltet. Konzepte der Sensor- und Erfassungssysteme, sowie smarte Aktuatoren wurden ebenfalls über Simulationen validiert.

- Auf dem Fahrzeuglevel wurden die Systemanalysen und Simulationen in der Art und Weise limitiert, um eine erhöhte Verfügbarkeit, Zuverlässigkeit und Fehlertoleranz zu erreichen, sodass die Sicherheitsanforderungen auf Systemebene weiterhin erreicht wurden.
- Insbesondere wesentliche ausfallsichere Systeme wurden in der Systemebene analysiert. Zusammenwirkende Sicherheitssysteme wurden hinsichtlich der Anforderungen aus Work Package 1 modelliert, simuliert und getestet. Des Weiteren wurden spezifische Konzepte für die Funktionale Sicherheit entwickelt.

Die zugehörige Terminübersicht in der ursprünglich geplanten Version (ohne Laufzeit-Verlängerung) ist im Anhang A abgebildet.

1.4 Ausgangsstand von Wissenschaft und Lösungsansatz

1.4.1 Stand der Wissenschaft und Technik

Der aktuelle Stand der Technik im Bereich automatisierten Fahrens liegt auf dem SAE-Level 3, d.h. der Fahrer ist mit einer gewissen Rückholzeit die letzte Rückfallebene bei technischen Problemen. Die E/E-Systeme dieser Generation sind sehr stark auf einem evolutionären Pfad unterwegs, um die Skalierbarkeit und die Wiederverwendung von bereits erprobten Technologien und Lösungsmuster zu berücksichtigen. Dabei stoßen diese Systeme immer mehr an ihre Grenzen. Weiterhin stehen immer mehr neue Technologien zur Verfügung, die den nächsten großen Schritt bei der Auslegung der E/E-Systeme ermöglicht, siehe hierzu auch [5]. Diese Systeme der nächsten Generation (Zeithorizont 2021) werden als „fail-operational“, d.h. ausfallsicher ausgelegt, um die SAE-Level 4 und SAE-Level 5 zu ermöglichen.

1.4.2 Neuheit des Lösungsansatzes

Ausgehend von der E/E-Generation 2021 wird die nächste E/E-Generation für das automatisierte Fahren im urbanen und ländlichen Raum mit SAE-Level >3 vorbereitet. Die drei wichtigen Schritte gliedern sich wie folgt. Siehe hierzu auch Abbildung 2.

1. Lessons learned: Sammlung und Auswertung der Erfahrungen seitens OEMs, Zulieferern, uvm., die sie bei der aktuellen E/E-Generation im Rahmen der Auslegung, Umsetzung und Integration gemacht haben.
2. Sammlung der Use Cases und Anforderungen an die nächste E/E-Generation mit Einführung ab 2025. Als Hauptmaßgabe gilt hier die Umsetzung des SAE-Levels >3 des automatisierten Fahrens.
3. Erarbeitung von neuen Lösungen unter der Berücksichtigung von neuen Technologien und den Erkenntnissen aus Punkt 1 sowie den Anforderungen aus Punkt 2.

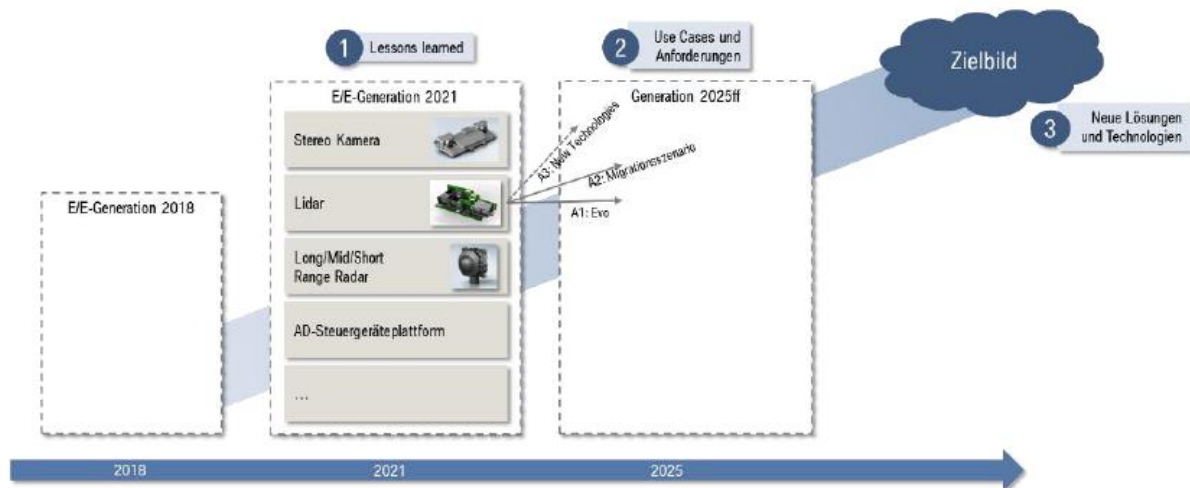


Abbildung 2: Weiterentwicklungspfad der E/E-Architekturgenerationen bei BMW und mögliche Umsetzungspfade im Jahr 2025, die über PRYSTINE erarbeitet und verifiziert werden

1.5 Zusammenarbeit mit anderen Stellen

Im Rahmen des durchgeführten Vorhabens gab es in allen Teilaufgaben und Arbeitspaketen mit BMW Beteiligung eine enge Zusammenarbeit zwischen den Projektpartnern. Um den Projektfortschritt sicherzustellen, wurden die erhaltenen Ergebnisse zwischen den Partnern ausgetauscht und diskutiert. Ein zentrales Austauschmedium war der von der OTH Amberg-Weiden zur Verfügung gestellte Projektserver.

2 Eingehende Darstellung

2.1 Verwendung der Zuwendung und erzielte Ergebnisse

2.1.1 WP1 – Anforderungen und Spezifikation

Seitens Fahrzeughersteller ergeben sich Anforderungen an das Fahrzeug-Verhalten in gegebenen Fahrsituationen und daraus heruntergebrochen Anforderungen an die einzelnen Systeme und Komponenten. Die Anforderungen an das Fahrzeug-Verhalten beziehen sich in der Regel auf typische oder besonders kritische Fahrsituationen, für die ein entsprechendes Fahrmanöver erfolgen soll. Im Projekt PRYSTINE wurden die Anforderungen anhand der Use Cases der Demonstratoren abgeleitet.

2.1.1.1 Organisation des Arbeitspaketes

In einem Kickoff-Meeting zum Arbeitspaket 1 wurde mit den beteiligten Partnern ein Arbeitsprozess vereinbart. Als Teil des Arbeitsprozesses wurde ein Jour Fixe Meeting im zweiwöchigen Rhythmus festgelegt, an dem die beteiligten Leiter der Supply Chains, aber auch die Leiter der acht Tasks im Arbeitspaket 1 regelmäßig teilnehmen.

Hauptanteil des Jour Fixe Meetings ist das Review des Fortschritts im Prozess der Anforderungs-sammlung und Spezifikation, aber auch Abstimmung unter den einzelnen Tasks und Supply Chains, Klärung von Fragen und Problemen, Dokumentation und Abarbeitung offener Punkte, sowie eine gesammelte Aktualisierung der bisher erarbeiteten Anforderungen und Spezifikationen aus den einzelnen Arbeitsteams.

2.1.1.2 Motivation des Arbeitspaketes

Der Fokus des Projekts Prystine ist die Umsetzung einer ausfallsicheren urbanen Umfelderkennung, um automatisierte Mobilitätsszenarien so zu verwirklichen, dass diese in einem unstrukturierten urbanen Umfeld sowohl unfallfrei als auch ohne Verletzung aller Verkehrsteilnehmer möglich sind.

Insbesondere sollen mit den entwickelten Systemen die spezifischen Bedürfnisse der am meisten gefährdeten Verkehrsteilnehmer, wie z.B. Fußgänger, Fahrradfahrer, Kinder und vor allem körperlich beeinträchtigte Personen und ältere Menschen, berücksichtigt werden, um ihre Mobilität und Sicherheit im Verkehr zu erhöhen.

Um dieses Ziel erreichen zu können, ist es notwendig ein klares und organisiertes Anforderungs-management zu etablieren. Alle beteiligten Partner im Arbeitspaket 1 müssen mit den Anforderungen systematisch umgehen, d.h. Anforderungen strukturiert erfassen, dokumentieren, überprüfen, untereinander abstimmen und verfeinern.

Für dieses Ziel wurden im Arbeitsprozess folgende Schritte definiert:

- Arbeitswerkzeug (Tool) und Dokumentstruktur definieren
- Erfassung und Sammlung der funktionalen und nichtfunktionalen Anforderungen
- Überführung der Funktionsanforderungen in Systemanforderungen und Dokumentation
- Ableitung der Komponentenanforderungen aus den Systemanforderungen und Dokumentation
- Abstimmung der Anforderungen mit den Projekt- und Prozesspartnern
- Definition der Abgabedokumente
- Erstellung eines Anforderungsdokuments (Lastenheft) für die nachfolgenden Arbeitspakete und Supply Chains
- Verfeinerung der Anforderungen und Spezifikationen während des Projekts, z.B. durch Reviews und Diskussion mit den implementierenden Partnern

2.1.1.3 Vorgeschlagenes Arbeitsmodell und methodisches Vorgehen

Grundsätzlich wurde allen beteiligten Arbeitsteams in den jeweiligen Teilaufgaben des Arbeitspakets die Arbeitsweise offengelassen, im Kickoff-Meeting wurden jedoch mögliche Arbeitsmodelle kurz vorgestellt und angesprochen.

Einerseits wäre bei dem Vorhaben der Anforderungserfassung die agile Methode SCRUM sehr gut geeignet, bei der jede Aufgabe in einem Feature Team organisiert wird und der jeweilige Taskleader die Aufgabe des Scrum Masters übernimmt.

Andererseits wurde auch die Methode SMArDT vorgestellt, die für „Spezifikations-Methode für Anforderung Design und Test“ steht. Hierbei handelt es sich um eine Methode für das Systems Engineering, die die strukturierte und systematische Spezifikation von Anforderungen und Design ermöglicht und beide miteinander in einem Model verknüpft.

SMArDT baut auf modellbasierter Entwicklung auf, nutzt die grafische Modellierungssprache SysML und ist damit toolbasiert. Dadurch können mit SMArDT komplexe Systeme realisiert, die Qualität der Projekte und Systeme gesichert und die Effizienz gesteigert werden.

Da die Anwendung der oben genannten Methoden im Arbeitsmodell freigestellt wurde, hat BMW zur Erreichung und Wahrung eines Mindeststandards bei den Anforderungen ein Dokument erarbeitet, um damit auch die Qualität der Ergebnisse sicherzustellen. Dieses Dokument hat sehr vielen Projektpartnern geholfen, insbesondere Partnern mit ausschließlich wissenschaftlichem Hintergrund, wie z.B. Forschungseinrichtungen, Universitäten und Hochschulen. Der wichtigste Leitsatz des Dokuments „Jede Anforderung muss testbar sein.“ Stellt nicht nur sicher, dass die Anforderungen nicht in mehrzeilige Beschreibungen ausarten,

sondern sichert gleichzeitig den Bezug bzw. schlägt die Brücke zum Arbeitspaket 6, welches sich um die „Test und Absicherung“ der Systeme kümmert. Eine sehr gute Visualisierung dieses Verhältnisses ist in Form eines V-Modells in der Abbildung unten, der Arbeitspaketstruktur des Projekts PRYSTINE, gelungen.

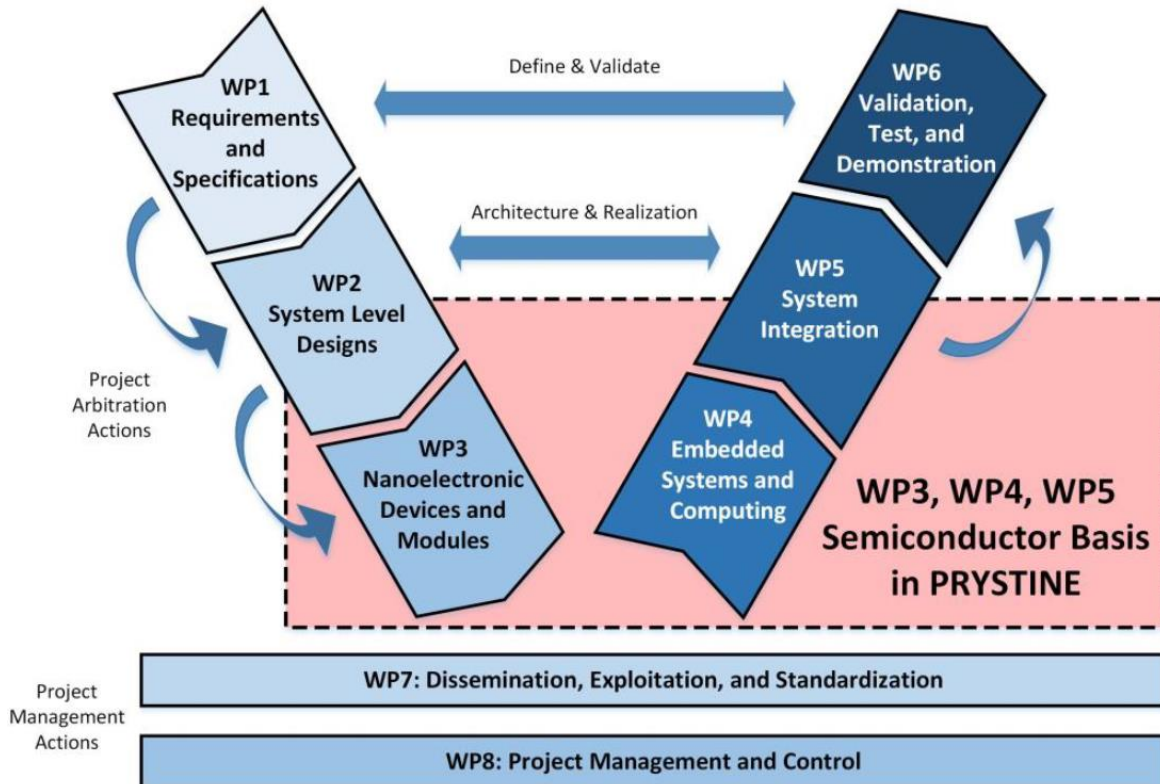


Abbildung 3: Arbeitspaketstruktur im Projekt PRYSTINE als V-Modell

Einen Auszug aus dem oben genannten Dokument mit dem Titel „Fit for Requirements“ stellt folgende Abbildung dar.

RULES FOR FORMULATING REQUIREMENTS.

Specify nouns completely. In the case of "data" or "user", you should ask, "which data" or "who is the user".

Do not use any vague quantifiers, such as "many", "mostly", but clear quantifiers, such as "every", "always", "no".

Specify adjectives using objectively measurable information. The adjective "large" requires, for example, a length specification in millimeters.

Formulate requirements as complete sentences in active voice (hence subject, verb, object) to make them easier to understand.

QUALITY CRITERIA FOR REQUIREMENTS.

BINDING CHARACTER

Requirements with the word "shall" (respectively "shall not" / "shall be a maximum") are binding only.

UNIQUENESS

The requirement must be formulated in a manner comprehensible by the supplier and may allow one interpretation only.

INDIVISIBILITY

It must not be possible to subdivide a requirement into further requirements.

Abbildung 4: Regeln zur Formulierung von Anforderungen und Qualitätskriterien, Auszug aus dem Regelkatalog „Fit for Requirements“ [6]

SENTENCE TEMPLATE FOR REQUIREMENTS.

Building block	Addressee of the requirement (subject)	Commitment indicator (modal verb)	Main verb (finite verb)	Object of the requirement <i>Optional</i>	Parameter of the requirement <i>Optional</i>	Temporal and/or logical conjunction <i>Optional, possibly several</i>
Question template	Who or what	is required	to assume, offer or provide	for whom or what	which target value / target condition or which function variant	when, in which situation and under which conditions
Sentence template	The development object / The addressee of the requirement	shall (at least)/ shall not/ shall be a maximum	provide (the requested functionality or the functionality to be provided)	the object of requirement / the "profiteer" of the requirement	value ... / condition ... / following ... / according to appendix ...	under conditions
Example building blocks	The vehicle The system ... The component ... The interface ... The function ... The property ...	shall shall be at least shall not shall be a maximum	delay display lock amount to meet be assume	the user / occupant the surroundings ... the vehicle the system ... the component ... the interface ... the function ... the property...	value ... / condition ... / output ... / the following value / according to appendix	during... as soon as... until... within... after... in the case of... when... according to... if...
Example requirements	The vehicle The steering force The Bowden cable The convertible top The pressure vessel system The parking brake The completely opened cover of the glove department	shall shall be a maximum shall not shall be shall have shall shall not	be able to electrically drive amount to show openable reduced hold touch	 the fuel pressure the vehicle the legs of the occupants	in reverse. xy N red rust up to a speed of xy km/h. within xy seconds up to a longitudinal inclination of xy degrees for all percentiles up to xy %.	 after a duration of 480 hours according to GS 90010. after operation of the vessel venting switch. under all conditions (weather, ...).

Abbildung 5: Vorlage zur Formulierung von Anforderungen, Auszug aus dem Regelkatalog „Fit for Requirements“ [6]

Als Tools für die Erfassung der Anforderungen wurden zwei Softwareumgebungen im Gremium des Arbeitspakets 1 diskutiert. Als ein gängiges und in der Entwicklung eingesetztes Tool zur Anforderungserfassung und -management wurde IBM Doors vorgeschlagen. Da die Software leider nicht die breite Anwendung über alle Partner des Projektkonsortiums findet, ist die Entscheidung auf die Alternative Microsoft Word und Excel gefallen.

Dennoch konnte auf Basis von Microsoft Word eine Struktur im Vorlagendokument erarbeitet werden, die den gleichen Vorgaben des Anforderungstools IBM Doors folgt und sogar aus MS Word heraus in Doors importiert werden kann, solange die definierte Struktur beibehalten wird.

Die Vorgabe des Vorgabedokuments sieht wie folgt aus.

TEMPLATE FOR SINGLE REQUIREMENT

ID: PRYSTINE_WP1_T1.1_1

Name:

Description:

Rationale:

Metrics:

Owner:

Reference UC:

Dependencies:

Conflicts:

README:

x: your task

id-number: latest existing ID number + 1

ID: please use *PRYSTINE_WP1_T1.x_* as prefix for the number identifying the requirement

Name: unique name for the requirement

Description: Concise definition of the requirement. One sentence max per one requirement.

Rationale: Explanation why this requirement is necessary.

Metrics: Proof to assess if the requirement is satisfied or not.

Owner: Beneficiary partner who created this requirement.

Reference UC: The reference to the UC which generated the requirement.

Dependencies: optional – leave empty if not needed

Conflicts: optional – leave empty if not needed

separation line: please use the separation line to separate the requirements clearly

AUTOMATION:

Use commands to automatically add a new requirement template form for a new requirement as described in the following.

Write **r1** and press F3 on the keyboard to add a new requirement field for task 1.1 or **r8 + F3** for task 1.8.

The enumeration of the requirements will be also done automatically.

Or insert an automatic text field from the menu shown in the picture below. You can choose between the tasks and select yours.

Tabelle 3: Vorgabedokument zur Anforderungserfassung mit Automatisierungsfunktion

In der Vorlage wurde eine Automatisierungsfunktion implementiert, die eine leere Anforderungsobjektstruktur erzeugt und die Anforderungen automatisch durchnummeriert. Mit Hilfe der vorliegenden Struktur konnten die Anforderungen schnell und effizient erfasst werden.

2.1.1.4 Ergebnisse des Arbeitspaketes

Das Arbeitspaket ist in acht Aufgabenpakete unterteilt, welche jeweils für jedes Subsystem stehen, die im Projekt verwirklicht werden sollen, siehe auch Tabelle 1. Für jedes dieser Aufgabenpakete wurden die Anforderungen und Spezifikationen in enger Abstimmung mit den OEMs, u.a. Maserati und BMW, durch die beteiligten Partner mit Hilfe der oben beschriebenen Arbeitsmethoden und Vorlagen definiert.

Da die Darstellung aller erarbeiteten Ergebnisse den Rahmen dieses Berichts sprengen würde, wird auf eine Auflistung verzichtet. Alle erarbeiteten Dokumente und Anforderungen stehen allen Partnern auf der Projekt Cloud der OTH Amberg-Weiden ohne jegliche Beschränkung zur Verfügung, siehe <https://prystine-cloud.automotive.oth-aw.de/>.

Auf Anforderung können entweder die Dokumente oder der Zugang zu den Dokumenten für eine Durchsicht zur Verfügung gestellt werden.

2.1.2 WP2 – System Level Design

Im Work Package 2, d.h. dem System Level Design, wurden von BMW im Rahmen des beschriebenen Teilvorhabens insgesamt drei wichtige Aufgaben bearbeitet. Diese Beiträge adressieren eins zu eins die Tasks 2.1, 2.4 und 2.5 des Arbeitspaketes, siehe auch **Fehler! V erweisquelle konnte nicht gefunden werden..** Eine kurze inhaltliche Beschreibung der jeweiligen Aufgabe ist im Abschnitt 1.1 zu finden.

2.1.2.1 Simulation eines LiDAR Einzelsensors basierend auf probabilistischen Sensormodellen

Mit steigendem Automatisierungsgrad benötigen die Fahrzeuge mehr und mehr Umfeldsensorik, um ihr Umfeld zu erfassen. Es gibt viele Umfeldsensoren auf dem Markt und neue automotivetaugliche Sensortechnologien werden laufend auf den Markt gebracht. Allerdings ist es eine herausfordernde Aufgabe ein geeignetes Sensorarrangement zu finden, das alle (Sicherheits-)anforderungen abdeckt. Neben Sensorleistung und Sichtfeld bzw. Abdeckung des zu überwachenden Bereichs müssen auch andere Faktoren, wie z.B. Kosten, Fahrzeugintegration und Designaspekte in Betracht gezogen werden. Außerdem sind ein redundantes Sensorsetup und die Sensorempfindlichkeit auf Umwelteinflüsse für die Sicherheit von enormer Bedeutung.

Es ist nicht möglich, jede einzelne Sensorkombination in Testfahrten zu erproben, die zeit- und kostenintensiv sind. Daher wurde bei der BMW AG eine Simulationsinfrastruktur eingeführt, um die Erkennungsleistung neuer Sensortechnologien und Gesamtsensorverbunde für zukünftige Fahrzeugkonzepte bereits in einem sehr frühen Stadium der Vorentwicklung zu evaluieren.

Das Simulationsworkflow basiert auf der Simulationsumgebung von Baselabs Create, die generische und probabilistische Sensormodelle, sowie ein Multi-Sensor Datenfusionsmodul zur Verfügung stellt. Baselabs Create bietet einige Sensormodell-Vorlagen für (intelligent) Radar und Kamera, siehe Abbildung 6.

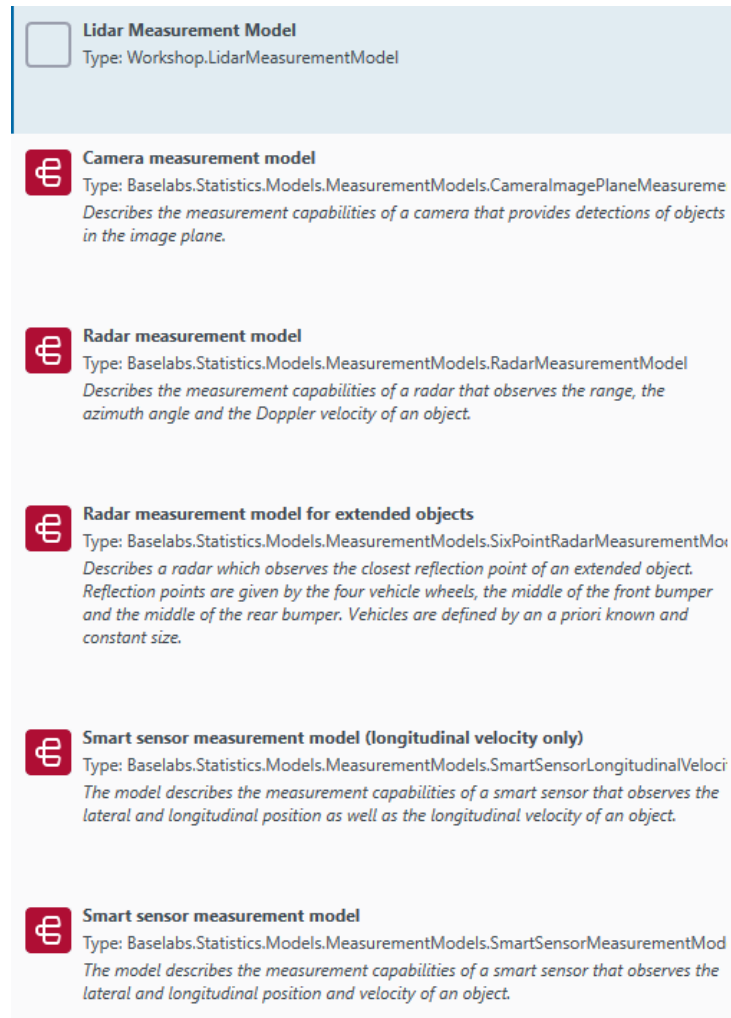


Abbildung 6: Sensormodell Vorlagen in der Simulationsumgebung Baselabs Create (rot) und neue LiDAR Sensormodell Vorlage von BMW (blau)

In der grafischen Benutzeroberfläche (GUI) ist es möglich die Eigenschaften jedes einzelnen Sensors im konfigurierten Setup individuell einzustellen. Jedes Sensormodell besteht aus einem Messmodell (Einbaulage, Genauigkeit), einem Detektionsmodell (Sichtbereich, Erkennungswahrscheinlichkeit) und dem Fahrbahnführungsverfahren. Da Baselabs Create auf probabilistische Sensormodelle setzt, arbeitet das LiDAR und alle anderen Sensoren deshalb nur mit Objektlisten und nicht mit Punktwolken, Radar Rohdaten oder Bildpixel. Abgesehen davon, ist es auch möglich die existierenden Vorlagen der Sensormodelle zu modifizieren oder eigene Sensormodelle mit einem Software Development Kit (SDK) hinzuzufügen.

Da LiDAR Sensoren als eine Schlüsseltechnologie für das automatisierte Fahren angesehen werden, sollten LiDAR Sensoren ebenfalls in die Sensorsetupanalysen integriert werden, indem diese Simulationsumgebung genutzt wird. Deshalb wurde ein geeignetes LiDAR Sensormodell erstellt und in die Simulationsumgebung zu Evaluierungszwecken implementiert. In den Projektdaten, die ebenfalls allen Projektpartnern zur Verfügung stehen, kann dieses benutzerdefinierte LiDAR Sensormodell ausgewählt werden, siehe Abbildung 6.

Die Eigenschaften dieses neuen LiDAR Sensormodells sind in Abbildung 7 (Einstellungen innerhalb des Mess- und Detektionsmodells), sowie in Abbildung 8 (Einstellungen innerhalb des Fahrbahnführungsverfahrens). Die Beschreibung der Polarkoordinaten, die im Messmodell integriert sind, sind auf der linken Seite der Abbildung 9 dargestellt. Im Gegensatz zu anderen Baselabs Sensormodellen, berücksichtigt das neue LiDAR Sensormodell auch die

Verdeckung von Objekten. Abbildung 9 (Situation auf der rechten Seite) erklärt, wie die Verdeckung von Objekten funktioniert. Wenn der Vektor von der Verbauposition des Sensors zum Referenzpunkt eines Objekts (hier: graues Fahrzeug) die Referenzlinie eines anderen Objekts kreuzt (hier: rotes Fahrzeug), wird das Objekt (graues Fahrzeug) als verdeckt (vom roten Fahrzeug) angesehen und wird bei der Detektion / Verfolgung nicht berücksichtigt, solange es verdeckt bleibt.

Diese Berücksichtigung der Verdeckung ist im Detektionsmodell inbegriffen. Abbildung a zeigt das Sichtfeld des LiDARs, während die Sensorsimulation mit einem beispielhaften Fahrscenario läuft.

Lidar

Measurement Model Lidar Measurement Model

Mounting

SensorPositionX8	1,7 m	<input type="button" value="..."/>	<input type="button" value="i"/>
SensorPositionY8	0 m	<input type="button" value="..."/>	<input type="button" value="i"/>
SensorRotationZ8	0 rad	<input type="button" value="..."/>	<input type="button" value="i"/>

Sensor characteristics

SigmaAzimuth19	0,004 rad	<input type="button" value="..."/>	<input type="button" value="i"/>
SigmaLength	0,5	<input type="button" value="..."/>	<input type="button" value="i"/>
SigmaRange19	0,4 m	<input type="button" value="..."/>	<input type="button" value="i"/>

Detection Model Detection Model with Occlusion

Sensor characteristics

DetectionProbability12	0,9	<input type="button" value="..."/>	<input type="button" value="i"/>
MaximumDetectionPhi	0,52 rad	<input type="button" value="..."/>	<input type="button" value="i"/>
MaximumDetectionRange	180 m	<input type="button" value="..."/>	<input type="button" value="i"/>
MinimumDetectionPhi	-0,52 rad	<input type="button" value="..."/>	<input type="button" value="i"/>
MinimumDetectionRange	0 m	<input type="button" value="..."/>	<input type="button" value="i"/>

Depends on State Measurement

Abbildung 7: Einstellungen zum Mess- und Detektionsmodell des neuen LiDAR Sensormodells

Lidar

Track Proposer	DataFusion.Simulation1.LidarClusterTrackPropc	i
Ego motion		
EgoVelocity20	Ego Motion Velocity (dynamic)	i
Mounting		
SensorPositionX10	SensorPositionX8 (1,7 m)	i
SensorPositionY10	SensorPositionY8 (0 m)	i
SensorRotationZ10	SensorRotationZ8 (0 rad)	i
Sensor characteristics		
SigmaAzimuthAngle	0,01 rad	i
SigmaLength2	0,5 m	i
SigmaRange20	0,4 m	i
Tracking		
InflationSigmaAcceleration2	1,4 m/s ²	i
InflationSigmaHeading2	0,25 rad	i
InflationSigmaLength	0,2 m	i
InflationSigmaPositionX10	0,2 m	i
InflationSigmaPositionY10	0,2 m	i
InflationSigmaVelocity10	20 m/s	i
InflationSigmaWidth	0,2 m	i

Abbildung 8: Einstellungen zum Fahrbahnführungsverfahren des neuen LiDAR Sensormodells

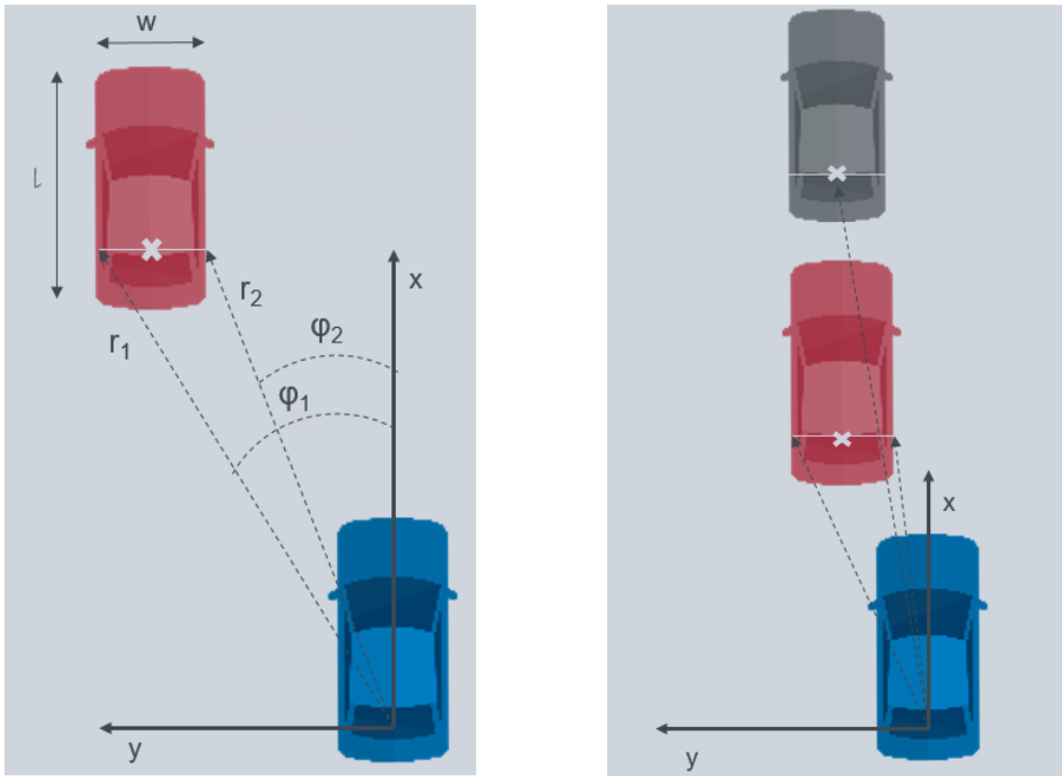


Abbildung 9: Beschreibung der Polarkoordinaten des neuen LiDAR Sensormodells (links) und Integration der Objektverdeckung innerhalb des Detektionsmodells für das neue LiDAR Sensormodell

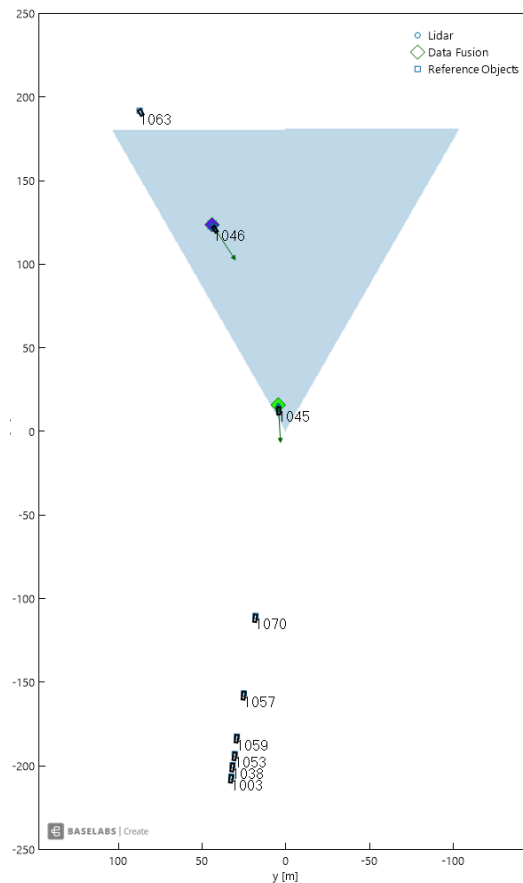


Abbildung 10: Visualisierung des Sichtfelds vom LiDAR Sensor während einer aktiven Sensorsimulation

2.1.2.2 System Design und Fahrzeugbustopologie für jedes Automatisierungslevel mit einer bereichs- und zonenorientierten E/E Architektur am Beispiel von Sensorik im Exterieur

Auf Basis der gesammelten Anforderungen im Projekt PRYSTINE und den Ergänzungen aus weiteren Trends wurde ein Systementwurf für eine zukünftige Bordnetztopologie abgeleitet. Ein Hauptaugenmerk des Entwurfs lag dabei auf einer durchgängigen Skalierbarkeit für die unterschiedlichen Automatisierungsstufen (Level 1 – Level 5). Dabei wurde auch der Sprung zwischen einem Fail-Safe zu einem Fail-Operational System mitberücksichtigt, siehe Abbildung 11.

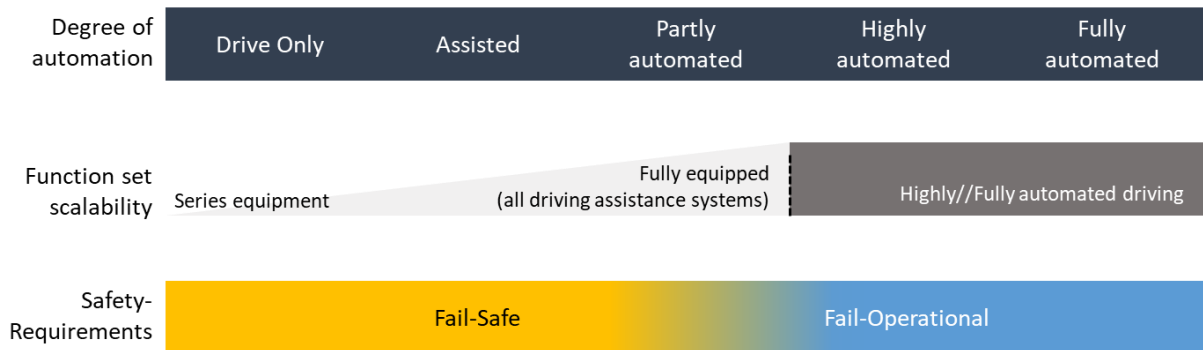


Abbildung 11: Zusammenhang Automatisierungsgrad, Funktionsset und Sicherheitsanforderungen

Durch die weitere Hochintegration der zentralen Funktionen im Fahrzeug werden zukünftig je nach Skalierungsstufen zwischen 1-2 leistungsfähige Rechenplattformen im Fahrzeug verbaut sein. Diese bieten die Rechenressourcen für Funktionen aus Powertrain, Chassis, ADAS, u.a. Die Plattformen unterstützen ASIL B bis ASIL D.

Des Weiteren werden der Kabelbaum und das Energiebordnetz zonale geschnitten. Über lokale Daten-/Energieverteiler (DEV) ist eine deutliche Optimierung des Kabelbaums (Länge, Bündelquerschnitte, usw.) erreichbar. Für die Anbindung der Umfeldsensorik in diesem Ansatz sind zwei Alternativen denkbar:

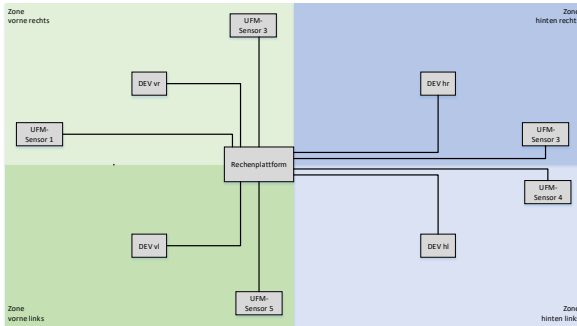
- 1) Anbindung der Sensorik lokal über die DEVs
- 2) Anbindung der Sensorik direkt an den Rechenplattformen

Beide Alternativen haben Vor-/Nachteile, z.B. höhere Bandbreite für die Anbindung der DEVs an den Rechenplattformen bei Alternative 1. Die Detaillierung und Bewertung der Alternativen ist nicht Gegenstand des Projektes PRYSTINE und wird daher in anderen Projekten erarbeitet. Abbildung 12 zeigt die beiden Alternativen jeweils für die Automatisierungsstufe 1 und 4. Für die Stufe 4 wurde aus Darstellungsgründen die Anbindung der Sensorik an der Rechenplattform (Redundanz) nicht mit aufgeführt.

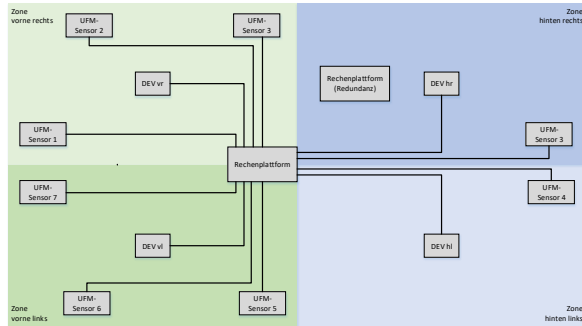
Prinzipdarstellung Level 1

Prinzipdarstellung Level 4

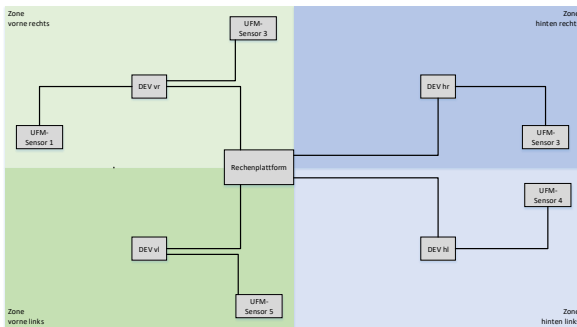
Zentrale Anbindung der Sensorik (level 1)



Zentrale Anbindung der Sensorik (level 4)



Lokale Anbindung der Sensorik (level 1)



Lokale Anbindung der Sensorik (level 4)

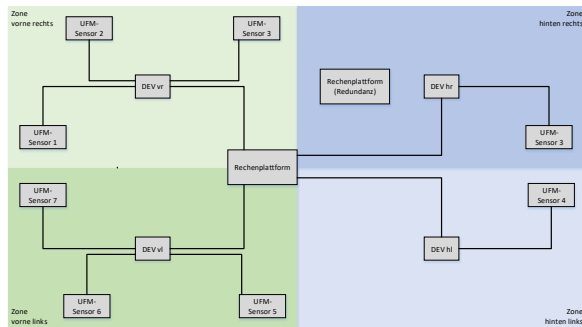


Abbildung 12: Prinzipdarstellung der Sensoranbindung bei unterschiedlichen Automatisierungsgraden

2.1.2.3 Methode zur Evaluation beliebiger Gesamtsensorkonfigurationen

2.1.2.3.1 Einleitung

Der Beitrag von BMW zum Task 2.5 im Arbeitspaket 2 adressiert die Evaluierung des Systemdesigns durch die Simulation verschiedener Sensoren, beispielsweise Kamera, Radar und LiDAR in einem Gesamtkontext und um die Umfelderkennung basierend auf dem Sensorfusionsmodell im Hintergrund abzuschätzen.

2.1.2.3.1.1 Motivation

Fortschrittliche Fahrerassistenzsysteme (ADAS) unterstützen den Fahrer mit Funktionen wie adaptive Geschwindigkeitsregelung (ACC), Notbremsung und Parkassistenten. Die Fahrzeuge sind mit mehreren externen Sensoren ausgestattet, um diese Funktionen mit Informationen über die Umgebung des Fahrzeugs zu versorgen. Die Automobilentwicklung geht nun in Richtung des hochautomatisierten Fahrens (HAF) und schließlich in Richtung des autonomen Fahrens (AF), um den Fahrkomfort und die Verkehrssicherheit zu erhöhen. Dieser steigende Automatisierungsgrad geht mit einer zunehmenden Anzahl von Sensoren wie Kameras, Radar, Lidar und Ultraschallsensoren einher, um eine ausreichende Erfassung der Fahrzeugumgebung zu gewährleisten (siehe Abbildung 13).

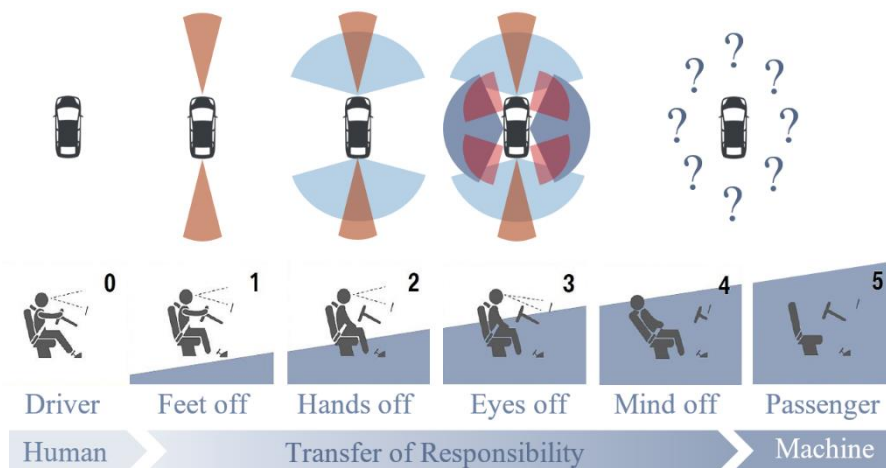


Abbildung 13: Zunahme der Komplexität des Sensoraufbaus im Zusammenhang mit der Entwicklung der Fahrautomatisierung

Während der Sensoraufbau für ein typisches ADAS System mit einem Radar, einer Kamera und Ultraschallsensoren an der Front bzw. am Heck noch überschaubar war, werden zukünftige Aufbauten komplexer sein. Um alle Anforderungen, die sich aus dem höheren Automatisierungsgrad ergeben, wie z.B. 360° Rundumsicht und eine redundante Fern- und Nahfeldabdeckung, zu erfüllen, müssen Sensoraufbauten mit einer viel höheren Anzahl unterschiedlicher Sensoren aufgebaut werden.

Dies führt zu einer Vielzahl von Sensorkonstellationen, die analysiert werden müssen, um die beste Lösung zu finden. Bei der Konfiguration des Aufbaus müssen drei Aspekte berücksichtigt werden:

Anordnung der Sensoren:

Der Aufbau kann mit mehreren Sensoren unterschiedlicher Sensortechnologien erfolgen. Eine redundante Sensoranordnung stellt sicher, dass wichtige Funktionen auch dann noch ausgeführt werden können, wenn ein Sensor ausfällt oder eine bestimmte Sensortechnologie in einer bestimmten Situation in ihrer Leistung nachlässt. Die Anordnung muss die Umgebung des Fahrzeugs ohne gefährliche tote Winkel abdecken. Außerdem müssen drei unterschiedliche Bereiche abgedeckt werden: das Fernfeld (Autobahnfahrt), das Nahfeld (Stadtfahrt) und das Ultra-Nahfeld (Park-/Startfahrt).

Integration der Sensoren:

Neben der optimalen Einbauposition und Sensorausrichtung im Hinblick auf Sichtfeldabdeckung und Sensorfunktionalität müssen auch die Machbarkeit der geometrischen Integration und Designaspekte berücksichtigt werden. Darüber hinaus sind Umwelteinflüsse wie Sensorverdeckung durch Schmutz, Witterung und Lichtverhältnisse für die Leistungsfähigkeit des Aufbaus entscheidend.

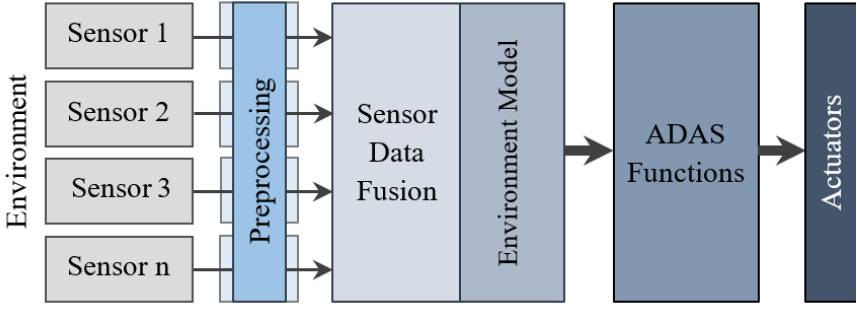
Sensor-Benchmarking:

Sensorspezifikationen wie FOV, Reichweite, Erkennungswahrscheinlichkeit und Genauigkeit sind entscheidend. Ein weiterer wichtiger Faktor sind jedoch die Kosten für die Gesamtkonfiguration. Kosten-Nutzen-Analysen können z. B. aufzeigen, ob zwei nebeneinander liegende Sensoren mit kleinem FOV einen teuren Sensor mit hohem FOV ersetzen können.

Die Definition eines guten Sensoraufbaus ist eine komplexe Aufgabe. Bislang fehlen ein konsistentes Bewertungsverfahren und ein geeignetes Werkzeug, das die Entwickler dabei unterstützt, diese Aufgabe in der Konzeptphase zukünftiger automatisierter Fahrzeuge zeit- und kosteneffizient zu lösen.

Wir haben daher ein simulationsbasiertes Evaluationsverfahren entwickelt, das die Entwicklung einer optimalen Sensorausstattung im Kontext des automatisierten Fahrens unterstützt. Das Simulations-Framework erlaubt die Konfiguration beliebiger Sensor-Setups und ermöglicht virtuelle Testfahrten innerhalb spezifischer Szenarien, um die Umweltwahrnehmung in frühen Entwicklungsphasen mit Metriken und Key Performance Indikatoren zu bewerten. Diese Evaluierungssuite könnte ein wichtiges Werkzeug für Forscher und Entwickler sein, um Zusammenhänge zwischen den Einstellungen zu analysieren und optimale Lösungen für die Umgebungswahrnehmung zu definieren.

2.1.2.3.1.2 Problemstellung

Sensoren sind der erste Teil einer komplexen Datenverarbeitungskette, die schematisch in  visualisiert ist.

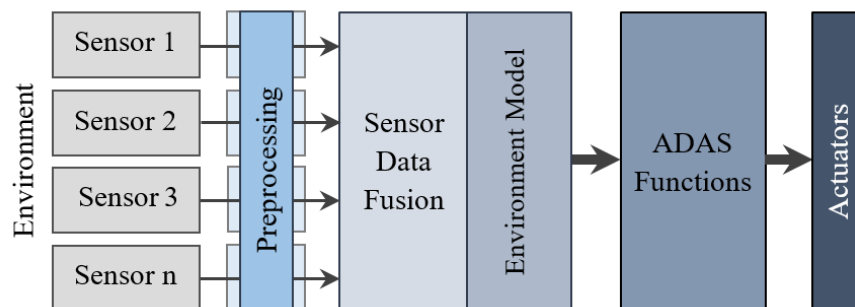




Abbildung 14: Schema der Datenverarbeitungskette in automatisierten Fahrzeugen

Die Messdaten werden vorverarbeitet und gehen dann in den Datenfusionsteil ein. Danach liefert ein Umgebungsmodell alle gesammelten Informationen über die Umgebung des Fahrzeugs, die dann an die ADAS-Funktionen weitergeleitet werden, die entsprechend Aktoren ansteuern können, z.B. Bremsen und Lenkung. Die Bestimmung des endgültigen Sensor-Setups spielt eine zentrale Rolle bei der Entwicklung des gesamten Datenverarbeitungssystems, da alle diese Teile darauf aufbauen.

Folglich muss das Setup in einer frühen Phase des Entwicklungsprozesses festgelegt werden, was im V-Modellschema in  (rot) visualisiert wird. Das Setup-Konzept muss in den Phasen "Systemarchitektur" und "Systemkonzept" festgelegt werden, da das Setup auch die Bauraumaufteilung (Fahrzeugintegration), sowie die Dimensionierung der Stromversorgung und der elektronischen Steuergeräte (ECUs) beeinflusst, was wiederum Auswirkungen auf die gesamte E/E-Architektur hat. Die Sensorhardware und die nachfolgenden Module der Wahrnehmungskette sind in den nachfolgenden Phasen des V-Modells zu entwickeln und stehen in den Phasen "Modul- und Systemintegrationstest" zur Verfügung, wie in  (grün) dargestellt.

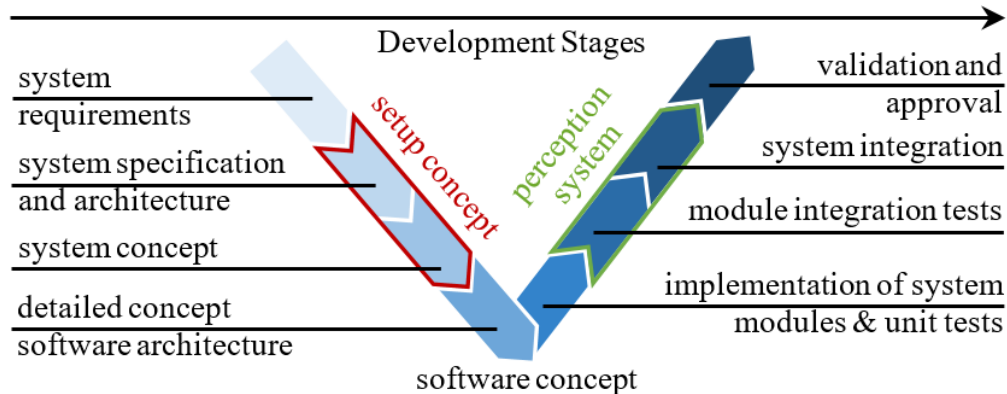


Abbildung 15: Fahrzeugentwicklungsprozess entsprechend des V-Modells

Diese Lücke zwischen der notwendigen Festlegung des Aufbaus und der Verfügbarkeit des entsprechenden Erfassungssystems macht es sehr schwierig, neue Aufbauten für zukünftige Fahrzeugkonzepte zu bewerten. Für vielversprechende neue Sensortechnologien auf dem Markt stehen derzeit noch keine serienreifen Sensoren zum Testen oder Sensormodelle für Simulationszwecke zur Verfügung.

Die Evaluierung von Sensoren besteht aus verschiedenen Messverfahren, um spezifische Anwendungsfälle zu analysieren und die angegebenen Sensorspezifikationen und Messprotokolle unter bestimmten Bedingungen in diesem frühen Entwicklungsstadium zu validieren. Dies ermöglicht den Vergleich verschiedener Sensoren der gleichen Technologie.

Es fehlt jedoch eine praktikable Methode, um die Leistungsfähigkeit des gesamten Aufbaus darzustellen. Die Durchführung von Testfahrten mit verschiedenen Sensorsystemen sowie die anschließende Datenanalyse sind zeit- und kostenintensiv. Daher ist diese Methode nicht geeignet, um viele Aufbaukonzepte in einer frühen Entwicklungsphase zu vergleichen.

Der Prozess der Definition des jeweiligen Sensoraufbaus sollte einem dreistufigen Verfahren folgen:

(1) Identifizierung der Sensorspezifikationen

Bestimmung der Leistung bestimmter Sensoren anhand von Datenblättern, Messprotokollen und Eckfalltests.

(2) Simulation des Sensoraufbaus

Verwendung der Informationen aus Schritt (1), um ein Simulationsprogramm zu speisen und Ergebnisse über die Gesamtleistung des Aufbaus zu erhalten.

(3) Testfahrten

Optimierung der Einrichtung durch Leistungstests unter realen Umgebungsbedingungen für die in Schritt (2) gewählte Einrichtung.

Die Schritte (1) und (3) beschreiben den Stand der Technik bei der Bewertung. Der Zwischenschritt (2) bietet die Möglichkeit, Zusammenhänge innerhalb dieses komplexen Zusammenspiels von Sensoren zu klären.

2.1.2.3.1.3 Lösungsvorschlag

Wir haben eine simulationsbasierte Bewertungsmethode entwickelt, um der Herausforderung zu meistern, die leistungsfähigste Lösung für die Sensoreinrichtung in frühen Entwicklungsphasen zu ermitteln. Der von uns vorgeschlagene Simulationsrahmen umfasst die Simulation von Umgebungs- und Sensordaten, die Datenfusion und eine Evaluierungssuite. Der Arbeitsablauf ermöglicht die Konfiguration beliebiger Sensor-Setups sowie die quantitative Bewertung ihrer Wahrnehmungsqualität in virtuellen Testfahrten anhand von Metriken und Key Performance Indikatoren (KPIs). Mit der von uns vorgeschlagenen Lösung ist es möglich, Sensor-Setups für automatisierte Fahrzeuge hinsichtlich bestimmter Sensor-Einbaupositionen, diverser Setup-Konfigurationen und unterschiedlicher Sensortechnologien zu evaluieren.

2.1.2.3.2 Hintergrund

Es gibt eine Menge an verfügbaren Simulationswerkzeugen für die Entwicklung von Automobilsystemen, siehe Tabelle 4.

Simulationswerkzeug	Unternehmen
CarMaker	IPG Automotive
PreScan + DRS360	Siemens
Speos, HFSS, VRXPERIENCE	ANSYS
DRIVE	NVIDIA
Virtual Test Drive VTD	Vires
CANape / vADASdeveloper	Vector
DYNA4 Driver Assistance	TESIS
ASM Traffic	dSPACE
Pro-SiVIC	CIVITEC
Automated Driving System Toolbox	MathWorks
Other simulation tools	rFpro, Addfor, ...

Tabelle 4: Simulationswerkzeuge für die Entwicklung von Automobilsystemen

Die meisten Simulationswerkzeuge wurden für die technische Konsolidierung und Prüfstandanwendungen in Hardware-in-the-Loop (HiL)- und Software-in-the-Loop (SiL)-Systemen entwickelt, um ADAS-Funktionen zu validieren oder um Datenfusions- und Erkennungsalgorithmen durch virtuelle Tests zu entwickeln. Um verschiedene Aufbaukonzepte zu evaluieren, muss der Fokus auf die Aufbaukonfiguration und eine entsprechende Evaluierungsstrategie gelegt werden. Im Folgenden werden die Schlüsselemente und Kriterien beschrieben, die bei der Entwicklung eines geeigneten Rahmens berücksichtigt werden müssen.

Virtuelle Umgebung:

Die Umweltsimulation kann hinsichtlich der bereitgestellten Datenausgabe in zwei Kategorien eingeteilt werden: basierend auf Objektlisten und physikbasiert.

Die meisten Umweltsimulationen erzeugen Objektlisten als Ground-Truth-Output, d.h. die virtuelle Umgebung liefert bereits Erkennungen, mit denen die Sensormodelle arbeiten können (High-Level-Daten). Die Objektlisten enthalten Angaben wie Objekttyp, ID, Position, Richtung, Geschwindigkeit, Beschleunigung und Bounding-Box-Abmessungen aller Objekte für jeden Zeitstempel. Der Begrenzungsrahmen ist ein Quader um die Außenabmessungen des Objekts, so dass die Objektlisten nicht die tatsächliche Form des Objekts enthalten.

Die andere Kategorie physikbasierter Umgebungssimulationen ermöglicht die Generierung von Low-Level-Sensordaten, d. h. die folgenden Sensormodelle erhalten realistische Eingabedaten zur Erzeugung von Sensor-Rohdaten (Radarsignale, Lidar-Punktwolken, Kamerabilder). Diese Art von virtueller Umgebung muss physikalische Parameter wie Objektform und Materialeigenschaften für alle Oberflächen liefern, z.B. Reflektivität, Absorption und Streueffekte. So können Wetter- und Beleuchtungsbedingungen einbezogen werden, die die Ausbreitung des Messsignals beeinflussen. Bei physikbasierten Simulationen ist darauf zu achten, dass die Physik wirklich integriert ist und sich nicht auf Rendering und Spezialeffekte beschränkt, wie sie aus Game-Engines bekannt sind.

Physikbasierte Simulationen liefern detaillierte Daten über die virtuelle Welt für die nachfolgenden Simulationsschritte, während Objektlisten eher abstrakte Informationen liefern. Dennoch haben wir uns für eine Umweltsimulation auf Basis von Objektlisten entschieden, da die physikbasierten Simulationswerkzeuge noch in der Entwicklungsphase sind, insbesondere im Hinblick auf valide Regenmodelle und die Datenbank mit allen Materialparametern für verschiedene Objektoberflächen.

Sensor-Modelle:

Sensormodelle sind virtuelle Sensoren, die in Simulationsanwendungen den realen Sensor repräsentieren. Sie lassen sich hinsichtlich ihrer Abstraktionsebene in vier Gruppen einteilen (siehe Abbildung 16):

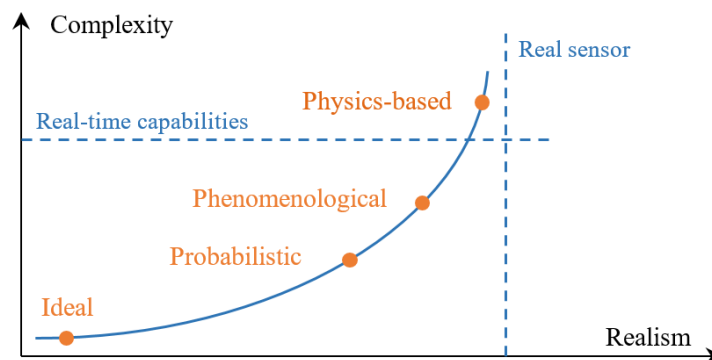


Abbildung 16: Abstraktionsstufen von Sensormodellen

Ideale Sensormodelle:

Sie sind rein geometriebasiert und funktionieren daher wie "perfekte Sensoren". Es ist möglich, den Sichtbereich des Sensors und den Erfassungsbereich zu definieren. Jedes Objekt innerhalb dieses Bereichs wird perfekt erkannt. Diese Modelle sind relativ einfach zu erstellen, zu implementieren und zu verwenden. Sie werden hauptsächlich zum Testen von Algorithmen verwendet: Wenn ein Algorithmus mit solchen perfekten Informationen nicht umgehen kann, wird er auch in der Realität versagen. Allerdings sind diese Sensormodelle stark vereinfacht und decken Messfehler und Rauschen nicht ab.

Probabilistische Sensormodelle:

Neben Sichtbereich und Detektionsreichweite wird das statistische Fehlverhalten von Sensoren durch Einstellungen bezüglich Erfassungswahrscheinlichkeit, Genauigkeit, falsch positive und falsch negative Ergebnisse abgedeckt. Diese Modelle sind realistischer als ideale Sensormodelle. Sie bieten eine breite Palette von Modifikationsparametern und anpassbaren Einstellungen, um sie an die Eigenschaften eines bestimmten Sensors anzupassen. Dies macht sie zur bevorzugten Art von Sensormodellen für den vorgesehenen Simulationsrahmen zu Bewertungszwecken. Wenn die benötigten Informationen über das Verhalten des Sensors, z.B. das Detektionsvermögen, nicht in den Datenblättern deklariert sind, sind experimentelle Verifizierungen notwendig, um diese Spezifikationen zu bestimmen.

Phänomenologische Sensormodelle:

Sie sind wie probabilistische Modelle aufgebaut. Darüber hinaus beziehen sie "situative Effekte" mit ein. Wenn das virtuelle Fahrzeug zum Beispiel in einen Tunnel oder auf eine Brücke fährt, können die Messfehler höher angesetzt werden. Das bedeutet, dass diese Sensormodelle in der Lage sind, das tatsächliche Sensorverhalten gut abzubilden, was aber voraussetzt, dass die Informationen über die situativen Effekte zugänglich sind. Diese Informationen sind in den Datenblättern nicht enthalten und sind erst nach mehreren Testfahrten verfügbar. Daher können diese Modelle nicht für eine erste Bewertung von Aufstellungskonfigurationen verwendet werden. Der Komplexitätsgrad dieser Modelle ist höher, da die Phänomene ("mehr Fehler in Tunneln und vor Brücken") sowohl im Sensormodell implementiert als auch mit der virtuellen Welt verknüpft werden müssen, die die Situationsinformationen liefern soll.

Physikalische Sensormodelle:

Diese Modelle erfüllen physikalische Gesetze und bieten die höchste Genauigkeit. Sie arbeiten wie Raytracing-Systeme zur Simulation der Signalausbreitung. Bei den Berechnungen werden Wetter- und Beleuchtungseffekte, sowie Signalreflexionen an Objekten und nassem Boden berücksichtigt. Daher bieten die physikalischen Modelle realistischere Sensordaten als die zuvor genannten Modelle, die synthetische Sensordaten liefern. Die Ausgabe der physikalischen Modelle sind Low-Level-Daten. Es ist ein zusätzlicher Datenverarbeitungsteil erforderlich, der aus den simulierten Rohdaten Objekterkennungen extrahiert, um High-Level-Daten zu erhalten. Physikalische Modelle können für den vorgeschlagenen Arbeitsablauf nicht verwendet werden, da sie von Seiten der Sensorhersteller zu diesem frühen Zeitpunkt der Einrichtungsbestimmung nicht zur Verfügung stehen werden. Die Erstellung eigener physikalischer Modelle ist eine schwierige Aufgabe und würde einen zeitintensiven Validierungsprozess erfordern, der bisher noch nicht etabliert ist.

Die Art des Sensormodells muss zur Umgebungssimulation passen. Während die synthetischen Modelle getrennt von der virtuellen Umgebung operieren können, die lediglich die Objektliste als Ground-Truth-Daten anbietet, arbeiten die physikalischen Modelle interaktiv mit der virtuellen Umgebung.

Datenfusion und Umgebungsmodell:

Es gibt verschiedene Ansätze für Datenfusionslösungen für Multisensorsysteme, die hier nicht im Detail diskutiert werden sollen. In der Regel werden in diesem Zusammenhang erweiterte Kalman-Filter eingesetzt. Die Algorithmen zur Datenfusion und das Umgebungsmodell entwickeln sich ständig weiter. Bei der Fahrzeugentwicklung müssen sie an die jeweiligen Sensoren angepasst werden. Mit Hilfe von Simulationen werden sie validiert und entsprechend optimiert. Im Gegensatz dazu sollte dieser Teil im Sinne einer Evaluierungsanwendung für Sensor-Setups fix bleiben, um eine einheitliche Basis zu gewährleisten.

ADAS-Funktionen:

Eine wichtige Anwendung von Simulatoren ist die Validierung von ADAS-Funktionen, die in der Lage sind, Aktoren im Fahrzeug zu steuern. Simulatoren können in Open-Loop- oder Closed-Loop-Konstellationen aufgebaut sein. Während die Open-Loop-Variante ein vorgegebenes Szenario testet und die Reaktion der ADAS-Funktion analysiert, erlauben Closed-Loop-Simulationen den Funktionen, virtuelle Aktoren anzusteuern und damit in das Szenario einzugreifen. Für die Bewertung der Wahrnehmungsleistung des Fahrzeugs, die das Ziel der vorliegenden Arbeit ist, haben wir uns entschieden, die Daten vor der Eingabe des

ADAS-Funktionsteils in einem Open-Loop-orientierten Bewertungsmodul zu analysieren und zu bewerten, das Metriken für KPIs berechnet und die Ergebnisse in einem kompakten Bericht darstellt.

2.1.2.3.3 Simulationsumgebung

Der von uns vorgeschlagene Framework deckt die gesamte Prozesskette von der Datenerfassung (Ground Truth, Sensordaten), der Datenverarbeitung (Datenfusion), der Datenauswertung (Metriken) bis hin zur Datenanalyse (KPIs) zur Ableitung von Datenerkenntnissen und Wissen (Setup Performance) ab. Der Workflow stellt den Schritt (2) Simulation des Sensoraufbaus des vorgeschlagenen Drei-Schritte-Verfahrens dar und ist in Abbildung 17 visualisiert.

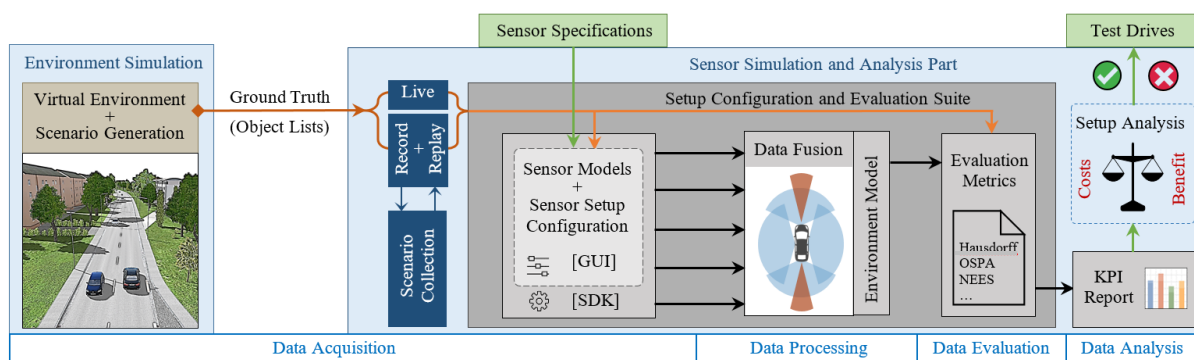


Abbildung 17: Schematische Visualisierung des vorgeschlagenen Simulationsframeworks und Arbeitsmodells für die Konfiguration und Evaluation von Sensorsetups in Fahrzeugen

Im ersten Teil werden mit einer auf Objektlisten basierenden Umweltsimulation Ground-Truth-Daten erzeugt. Im vorliegenden Kontext wird IPG CarMaker verwendet. Die virtuelle Umgebung, der Verkehr und das Fahrmanöver des Ego-Fahrzeugs werden so gestaltet, dass ein Szenario entsteht. Die simulierten Ground-Truth-Daten des Szenarios werden an den zweiten Workflow-Teil übergeben, der aus der Setup-Konfiguration und der Evaluationssuite besteht. Wir haben die Software Baselabs Create verwendet, ein Entwicklungswerkzeug für Umweltwahrnehmungsanwendungen mit Schwerpunkt auf Datenfusion. Die Ground-Truth-Daten werden in einer Szenario-Sammlung aufgezeichnet. Die Evaluationssuite kann separat mit diesem generierten Szenariendatensatz verwendet werden. Die Umweltsimulation wird nur benötigt, um zusätzliche Szenarien für die Szenariosammlung zu erstellen.

Es sei erwähnt, dass es technisch möglich ist, die objektlistenbasierte Umweltsimulation später zu einer physikbasierten aufzurüsten, da das Open Simulation Interface (OSI) für den Datentransfer von der Umweltsimulation zur Sensorsimulation verwendet wird. Innerhalb des OSI-Formats können spezifische Informationen über das Objekt (z.B. Fahrzeugtyp und Farbe) und den Wetterzustand geliefert werden. Diese Informationen können genutzt werden, um in einem zwischengeschalteten Simulationsschritt die für die physikbasierte Sensorsimulation notwendigen Informationen wie Objektform und Signaldämpfung durch Wettermodelle zu berechnen. OSI wird derzeit im Rahmen des Gemeinschaftsprojekts PEGASUS und der Association for Standardization of Automation and Measuring Systems (ASAM) als gemeinsamer Standard von Automobilherstellern, Sensorentwicklern und anderen Zulieferern betrachtet.

Die Setup-Konfiguration und die Auswertungssuite bestehen aus drei Segmenten: Erstens, die Ground-Truth-Daten gehen in den Datenfusionsdesigner mit den Sensormodellen und der Setup-Konfiguration ein. In diesem Teil werden die Ground-Truth-Daten entsprechend den Einstellungen der probabilistischen Sensormodelle modifiziert. Für jeden Sensor des

Setups wird über eine grafische Benutzeroberfläche (GUI) ein Messmodell, ein Detektionsmodell und eine Spurverwaltungsstrategie ausgewählt. Dabei werden die einzelnen Sensorparameter entsprechend der aus dem Dreischritt (1) *Identifikation von Sensorspezifikationen* gepflegten Eigenheiten eingestellt. Darüber hinaus ist es möglich, die vorhandenen Sensormodelle zu modifizieren oder neue Modelle mit Hilfe des Software Development Kit (SDK) hinzuzufügen.

Die simulierten Sensordaten werden im Datenfusionsteil fusioniert, der sich auf erweiterte Kalman-Filter stützt, und fließen in das nachfolgende Umgebungsmodell ein. Anschließend werden die simulierten und fusionierten Sensordaten mit den anfänglichen Ground-Truth-Daten im Modul für Bewertungsmetriken verglichen, das speziell für diesen Arbeitsablauf entwickelt wurde. In diesem Teil werden benutzerdefinierte Bewertungsmetriken hinzugefügt. Auf der Grundlage dieser Metriken wird ein KPI-Bericht erstellt, der die Ergebnisse für alle Szenarien enthält, um die Wahrnehmungsleistung in einem kompakten Überblick quantitativ zu bewerten.

Danach werden die KPI-Berichte verschiedener Setups verglichen und weitere Aspekte wie Kosten-Nutzen-Überlegungen analysiert, um zu entscheiden, ob ein bestimmtes Sensor-Setup im nächsten Schritt des dreistufigen Verfahrens getestet werden sollte: (3) *Testfahrten*.

2.1.2.3.3.1 Szenarien für das virtuelle Testen (CarMaker)

In der Endphase der Fahrzeugentwicklung müssen die Sicherheitsfunktionen für hochautomatisiertes Fahren und fortschrittliche Fahrerassistenzsysteme in vielen verschiedenen Szenarien mit Millionen von virtuellen Testkilometern validiert werden, um ihre korrekte und sichere Funktionsweise in jeder Situation zu gewährleisten. Zu diesem Zweck gibt es Szenariensammlungen mit mehreren hundert Fahrmanövern. In der frühen Konzeptphase der Fahrzeugentwicklung ist es nicht möglich, alle diese Szenarien für die Bewertung verschiedener Sensoreinstellungen zu verwenden. Es ist produktiver, einige aussagekräftige Szenarien zu extrahieren, die für die Bewertung der Leistung des Sensorsystems in diesem frühen Entwicklungsstadium relevant sind. Da die Umgebungs- und Sensorsimulation auf Objektlisten basiert, ist es nicht sinnvoll, viele Szenarienvariationen für die Erkennung von Fahrbahnmarkierungen und Verkehrszeichen zu haben. Das Hauptaugenmerk in der Konzeptphase liegt auf der Bereitstellung einer Sensoranordnung mit einer guten Umgebungsabdeckung ohne tote Winkel.

Um die Auswahl aussagekräftiger Szenarien für die Bewertung des Aufbaus zu erleichtern, wurde eine Heatmap mit Interessensgebieten (ROIs) erstellt (siehe Abbildung 18: Heatmap mit Interessensregionen und korrespondierenden Kritikalitätsindizes). Bei den ROIs handelt es sich um die dreidimensionalen Räume um das Ego-Fahrzeug herum, die das Fahrzeug in irgendeiner Weise beeinflussen können, wenn es seine verschiedenen Fahraufgaben ausführt, und die von den Sensoren überwacht werden müssen. Die Interessengebiete wurden durch die Untersuchung der Szenariensammlung für die Validierung der Betriebssicherheit und durch die Analyse der gesammelten Testfahrt Daten definiert. Für die Entwicklung der ROI-Wärmelandkarte wurde der Bereich um das Ego-Fahrzeug in fünf grundlegende Sektoren unterteilt (vorne, hinten, seitlich, vordere Ecken, hintere Ecken) mit einer weiteren Klassifizierung in "nah" (bis zu 5 m), "mittel" (bis zu 50 m) und "fern" (mehr als 50 m).

Die Szenarien werden außerdem in drei Kategorien unterteilt, die auf den Straßentypen und der Umgebung basieren, in der sie stattfinden: Stadt, Land (ländlich) und Autobahnen.

- Städtisch:

Geschwindigkeiten von nicht mehr als 60 km/h, geringer Krümmungsradius der Straßen, viel mehr Verkehrsteilnehmer und schwächere Verkehrsteilnehmer wie Fußgänger, Radfahrer usw. sowie besondere Fahrszenarien wie Parken.

- Ländlich:

Geschwindigkeiten von nicht mehr als 100 km/h, weniger Verkehrsteilnehmer, aber unvorhersehbare Straßenführung in Bezug auf Krümmungsradius, Höhenlage, Straßenzustand usw., unvorhersehbare Objekte wie Tiere, landwirtschaftliche Maschinen, die schwer zu verfolgen und wahrzunehmen sein können.

- Schnellstraße:

Hohe Geschwindigkeit (mindestens 60 km/h) und in einigen Fällen bis zu 250 km/h, aber niedrige Relativgeschwindigkeiten, sehr wenige unerwartete Verkehrsteilnehmer, sehr vorhersehbare Fahrzeugtrajektorien, begrenzte Anzahl von Verkehrsteilnehmern.

Neben den oben genannten Klassifizierungen werden die Szenarien auch nach der Relativgeschwindigkeit zwischen dem Ego-Fahrzeug und anderen Teilnehmern des Szenarios klassifiziert. Die Relativgeschwindigkeit wird später als Schlüsselindikator für die Kritikalität des Szenarios verwendet.

Zusammenfassend haben die Untersuchungen ergeben, dass die vorderen Ecken die wichtigsten Regionen sind, da sie die höchste Anzahl von Szenarien aufweisen und mit einer hohen Relativgeschwindigkeit gekennzeichnet sind. Die vorderen Eckbereiche sind auch deshalb wichtig, weil sie bei komplexen Manövern wie Überholen, Erkennen von Verkehrsschildern, Bahnplanung (beim Abbiegen) und Ausschau halten nach Objekten, die die Fahrzeugbahn seitlich kreuzen (z. B. wenn das Fahrzeug bei einer roten Ampel mit grünem Pfeil rechts abbiegen will oder Fußgänger die Straße an einem Zebrastreifen überqueren), eine Schlüsselrolle spielen. Außerdem ist der Raum, der in den Eckbereichen abgedeckt wird, größer als der vordere und hintere Bereich. Die vorderen und hinteren Bereiche sind ebenfalls wichtig, aber aufgrund ihres engen Sichtfeldes relativ einfach, und die genannten Szenarien sind nicht sehr komplexer Natur. Die seitlichen Regionen sind weniger komplex, da sie einen viel geringeren Sichtbereich benötigen, und daher werden die Szenarien nur nach relativen Geschwindigkeiten klassifiziert. Auch die Relativgeschwindigkeiten dieser Szenarien sind im Allgemeinen niedriger. Die hinteren Eckregionen weisen die geringste Anzahl von Szenarien auf und können als die am wenigsten komplexen Regionen angesehen werden.

Die entwickelte Heatmap (Abbildung 18) zeigt die identifizierten ROIs mit ihrem entsprechenden kritischen Index c . Der kritische Index ist ein Maß für die Kritikalität oder Wichtigkeit einer ROI. Er basiert auf drei Faktoren: der relativen Geschwindigkeit zwischen dem Ego-Fahrzeug und einem Zielobjekt, der Entfernung zwischen dem Ego-Fahrzeug und einem Zielobjekt und der Art des Zielobjekts (z. B. ist ein Fußgänger kritischer als ein Pkw).

$$c = \frac{v^2}{2a} \cdot F \cdot d$$

Wobei c = Kritikalitätsindex, v = relative Geschwindigkeit, a = Verzögerung, d = Abstand zwischen Ego-Fahrzeug und Objekt, F = Objektfaktor

Der Kritikalitätsindex c kann einen Wert zwischen 0 und 1 annehmen, wobei 0 das am wenigsten kritische und 1 das am meisten kritische Szenario darstellen würde. Die Gleichung zur Bestimmung des kritischen Indizes ist das Verhältnis zwischen dem Bremsweg und dem Abstand zwischen dem Ego-Fahrzeug und einem Zielobjekt. Es wird davon ausgegangen, dass der Bremsweg immer kleiner ist als der Abstand zum betrachteten Objekt, da sonst eine Kollision mit keiner Sensoranordnung verhindert werden kann und somit kein Bewertungspotenzial besteht. Um den Kritikalitätsfaktor des Zielobjekts einzubeziehen, wird ein Objektfaktor mit diesem Verhältnis multipliziert. Der Objektfaktor richtet sich nach der Kritikalität des Zielobjekts in der Szene und der Wahrscheinlichkeit von schweren Verletzungen, die bei einem Unfall mit dem Ego-Fahrzeug zum Tod einer Person führen

können. Im Hinblick auf die Unfallstatistik wurden die Objektfaktoren gemäß Tabelle 5 festgelegt.

Objekt im Szenario	Objektfaktor
Fussgänger	1
Kollision mit einem Baum und anderen statischen Objekten	1
Motorradfahrer	0.9
Fahrradfahrer	0.8
Lastkraftwagen	0.7
Personenkraftwagen	0.5

Tabelle 5: Objektfaktoren (F) für diverse Objekte

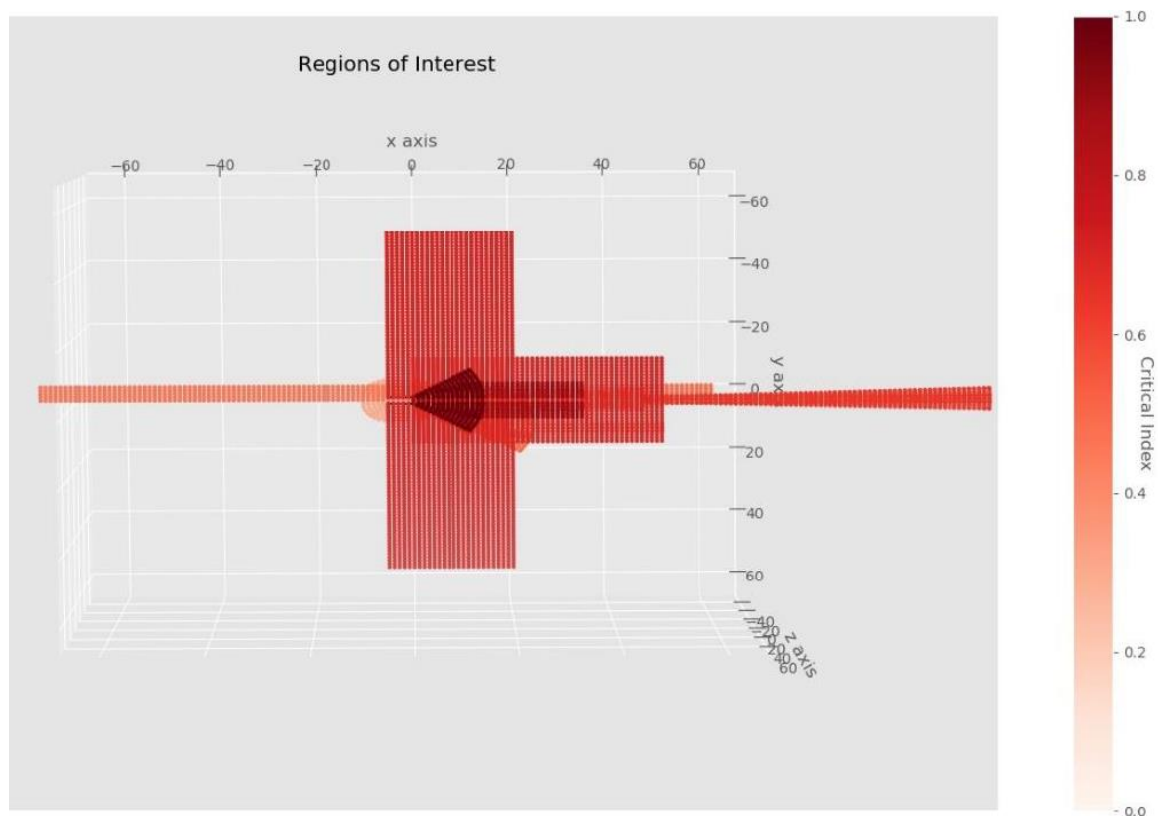


Abbildung 18: Heatmap mit Interessensregionen und korrespondierenden Kritikalitätsindizes

Die Heatmap enthält Interessensbereiche (ROIs), die auf den folgenden Überlegungen basieren:

- Notbremsassistent und Erkennung von Hindernissen
- Senkrechter Übergang für gefährdete Verkehrsteilnehmer
- Gefährdete Verkehrsteilnehmer beim Überqueren von Kreuzungen
- Erkennung von Verkehrszeichen
- Fahrspurhaltesystem
- Kreuzung und rotes Signal mit grünem Pfeil
- Einfahrt auf die Autobahn
- Autobahnausfahrt

Auf der Grundlage dieser Heatmap wurden mit IPG CarMaker Testszenarien erstellt, die alle Teile dieser ROIs abdecken. Sie sind in Tabelle 6 aufgeführt. Zum besseren Verständnis der Kreuzungsszenarien zeigt Abbildung 19 diese Szenarien im Detail.

Szenarien	Parken	Kreuzung	Autobahn
Szenario #1	Vorwärts Einparken	Kreuzende Fussgänger	Autobahnauffahrt
Szenario #2	Rückwärts Einparken	Radfahrer Geradeausfahrt und abbiegendes Ego-Fahrzeug	Autobahnabfahrt
Szenario #3		Geradeausfahrt Ego-Fahrzeug und verdeckter Motorradfahrer von links	Notbremsung – statisches Objekt
Szenario #4		Geradeausfahrt Ego-Fahrzeug und links abbiegendes Ziel von rechts	Notbremsung – dynamisches Objekt
Szenario #5		Gegenüberliegende Linksabbieger	

Tabelle 6: von einer ROI Heatmap abgeleitete relevante Szenarien zur Evaluation von Sensorsetups

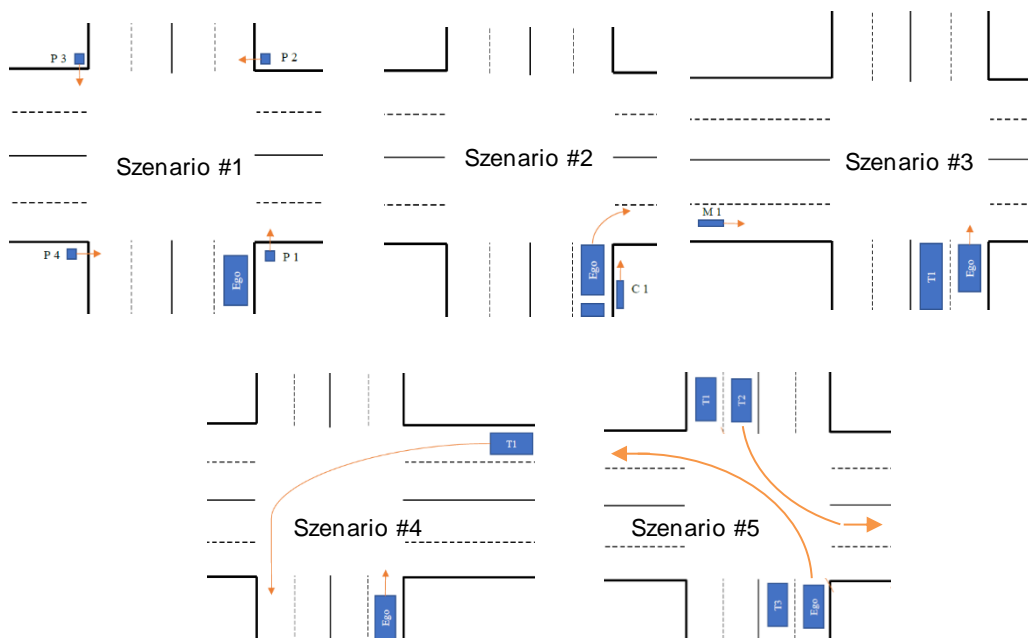


Abbildung 19: Szenarien der Szenariogruppe „Kreuzung“

Zusätzlich wurde das Parametrisierungstool AVL Cameo verwendet, um verschiedene Variationen der einzelnen Szenarien zu erstellen und das kritischste auszuwählen. AVL Cameo wurde aufgrund der verschiedenen verfügbaren Versuchsplanungsmodule und seiner bewährten Modellierungs- und Optimierungsfähigkeiten im Bereich der Motorenentwicklung und Kalibrierung in der Automobilindustrie ausgewählt. Cameo bietet außerdem eine Vielzahl von Skriptsteuerungsfunktionen. Die Simulationsversion von Cameo wird außerdem mit einer einsatzbereiten Schnittstelle zwischen Cameo und CarMaker geliefert.

2.1.2.3.3.2 Setupkonfiguration, Parametrierung der Sensormodelle und Datenfusion (Baselabs Create)

Innerhalb des Datenfusionsdesigners von Baselabs Create können beliebige Sensoraufbauten konfiguriert werden. Abbildung 20 zeigt die verschiedenen Sensormodelle, die ausgewählt werden können, um einen bestimmten Sensor im virtuellen Setup zu repräsentieren (links), und die angepasste Setup-Anordnung (rechts). Bei Bedarf können die Sensormodelle mit dem Software Development Kit (SDK) modifiziert oder neue Sensormodelle hinzugefügt werden.

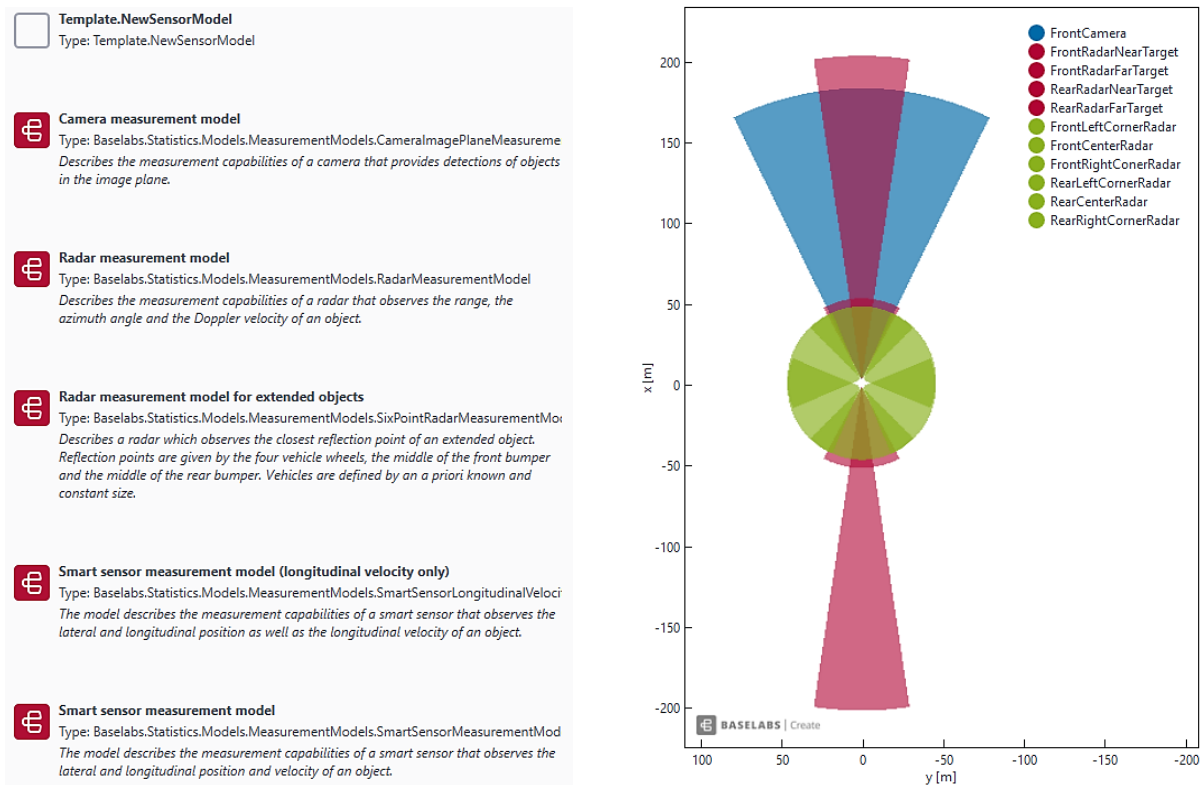


Abbildung 20: Konfiguration virtueller Sensorsetups mit unterschiedlichen Sensormodellen in Baselabs Create

Abbildung 21 zeigt beispielhaft die Sensormodelleinstellungen für das Messmodell, das Erkennungsmodell und den Track Proposer für einen Radarsensor.

<p>Measurement Model Radarsensor</p> <p>Ego motion</p> <p>EgoAngularRate: Ego Motion Yaw Rate (dynamic)</p> <p>EgoVelocity: Ego Motion Velocity (dynamic)</p> <p>Mounting</p> <p>RadarPositionX2: 3,7 m</p> <p>RadarPositionY2: 0 m</p> <p>RadarRotationZ2: 0 rad</p> <p>Sensor characteristics</p> <p>SigmaAzimuth2: 0,01 rad</p> <p>SigmaRange2: 1,2 m</p> <p>SigmaRangeRate2: 0,45 m/s</p>	<p>Track Proposer Radarsensor (inkl. Kreuzverkehr)</p> <p>Ego motion</p> <p>EgoAngularRate14: Ego Motion Yaw Rate (dynamic)</p> <p>EgoVelocity14: Ego Motion Velocity (dynamic)</p> <p>Mounting</p> <p>RadarPositionX12: RadarPositionX2 (3,7 m)</p> <p>RadarPositionY12: RadarPositionY2 (0 m)</p> <p>RadarRotationZ12: RadarRotationZ2 (0 rad)</p> <p>Sensor characteristics</p> <p>SigmaAzimuth12: SigmaAzimuth2 (0,01 rad)</p> <p>SigmaRange12: SigmaRange2 (1,2 m)</p> <p>SigmaRangeRate12: SigmaRangeRate2 (0,45 m/s)</p> <p>Tracking</p> <p>InflationSigmaPositionX3: 0,001 m</p> <p>InflationSigmaPositionY3: 0,001 m</p> <p>InflationSigmaVelocity3: 1,4 m/s</p> <p>InitialExistenceProbability3: 0,6</p> <p>SigmaAcceleration3: 1,4 m/s²</p> <p>SigmaHeading3: 0,2 rad</p>
<p>Detection Model Radarsensor</p> <p>Field of view</p> <p>MaximumPhi2: 0,5236 rad</p> <p>MaximumRange3: 250 m</p> <p>MinimumPhi2: -0,5236 rad</p> <p>MinimumRange3: 0 m</p> <p>Sensor characteristics</p> <p>DetectionProbability3: 0,9</p> <p>Depends on: <input type="radio"/> State <input checked="" type="radio"/> Measurement</p>	

Abbildung 21: Messmodell, Detektionsmodell und Track Proposer Einstellungen der Sensormodelle

Während des Simulationslaufs kann die Objekterkennung und -verfolgung auf einem Visualizer überwacht werden. Sobald ein Objekt im Sichtfeld eines Sensors auftaucht, wird ein Track erstellt, abhängig von der Messgenauigkeit, der Erkennungswahrscheinlichkeit und der Track-Initialisierung, die im Sensormodell festgelegt wurden. Wird ein Objekt von zwei oder mehr Sensoren mit geringfügigen Unterschieden in der Objektposition oder -geschwindigkeit erkannt, gewichtet die Datenfusion die unterschiedlichen Werte im Hinblick auf die in den Sensormodellen eingestellte Leistung der Sensoren und erzeugt entsprechend eine Spur. Für den Fall, dass keine neue Messung die Spur verifiziert, wird die Spur unter Berücksichtigung der gewählten Spurverwaltungsstrategie entfernt.

Generell erlaubt das Tool Baselabs Create die Optimierung des Datenfusionsalgorithmus für eine bestimmte Sensoranordnung. Für die Evaluierung des Aufbaus sollte die Datenfusion festbleiben, um eine konsistente Basis zu schaffen.

2.1.2.3.3 Erweiterung der Umgebung: 3D und Verdeckung (Baselabs Create)

Das Softwaretool Baselabs Create basiert auf einer zweidimensionalen Architektur, d. h. alle Sensoren und alle Bodenwahrheitsobjekte befinden sich auf derselben Ebene in Bezug auf die Z-Achse, und die Bodenwahrheitsobjekte werden als Objektlistenpunkte dargestellt (die sich in der Mitte der Hinterachse des Fahrzeugs befinden, da dies der Drehpunkt ist). Dies mag ein geeigneter Simulationsansatz für die Datenfusionstechnik sein, nicht aber für die Bewertung des Aufbaus.

Daher wurde der Simulationsrahmen angepasst, um eine dreidimensionale Basis für die Sensorpositionierung (Z-Position der Sensoren) und die Sensormessung (vertikales FOV; Nick- und Rollwinkel für die Sensororientierung) zu schaffen. Zusätzlich wurde die rechnerische Berücksichtigung von Verdeckungseffekten implementiert, um zwischen verschiedenen Montagepositionen genau unterscheiden zu können. Die Verdeckungsberechnungen können anhand von Abbildung 22 erläutert werden.

Für jedes Bodenwahrheitsobjekt innerhalb des Sichtfeldes eines Sensors wird ein Strahl zwischen der Position des Sensors auf dem Ego-Fahrzeug (blau) und dem Mittelpunkt der Bounding Box (BBC) des jeweiligen Bodenwahrheitsobjekts (rot) gezeichnet. Wenn dieser Strahl eine andere Bounding Box eines beliebigen Grundwahrheitsobjekts (grün) kreuzt, wird das (rote) Objekt als verdeckt eingestuft, d. h. es wird bei der Datenfusionsverfolgung nicht berücksichtigt, bis es wieder sichtbar ist. Im Vergleich zur 2D-Version dürfte die Erkennungsrate bei zufälligen Referenzszenarien viel niedriger sein, da viele Objekte verdeckt sein können und von der Einrichtung nicht gesehen werden.

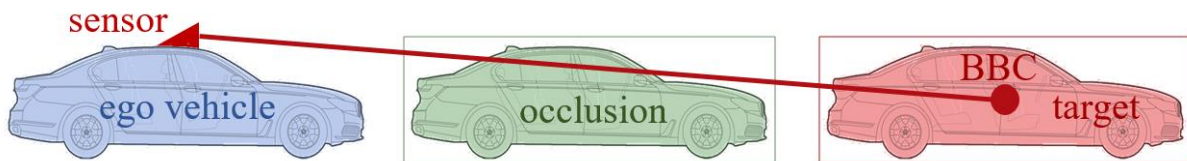


Abbildung 22: Verdeckungsberechnung innerhalb des Detektionsmodells in Baselabs Create

Um Rechenzeit zu sparen, prüft der Verdeckungsalgorithmus des Erkennungsmodells zunächst, ob sich ein potenzielles Verdeckungsobjekt (grün) zwischen dem Ego-Fahrzeug (blau) und dem Zielfahrzeug (rot) befindet, indem er die Abstände der Objekte vergleicht, und ob seine größte Abmessung (Diagonale der Bounding Box BB des Objekts) größer ist als sein Abstand zum Strahl. Die notwendigen Informationen für die BB (Länge, Breite, Höhe, BB-Mitte BBC und Vektor von BBC zu BB-Hinterachse) werden von den Ground-Truth-Objektlisten im OSI-Format geliefert. Wenn beide Bedingungen erfüllt sind, wird das betrachtete Objekt als potenzielle Verdeckung eingestuft. Anschließend prüft der Verdeckungsalgorithmus, ob das Objekt das Zielobjekt tatsächlich verdeckt, wie in Abbildung 22 dargestellt. Für diesen Schritt werden alle Schnittpunkte der sechs BB-Ebenen und des Strahls berechnet (siehe Abbildung 23). Die Schnittpunktswerte werden nach ihrem Abstand zum Sensor sortiert, und es wird jeweils ein Array für die entfernten und die nahen Schnittpunkte erstellt. Ist der kleinste Wert der entfernten Schnittpunkte größer als der größte Wert der nahen Schnittpunkte, schneidet der Strahl die BB und das Objekt stellt somit eine Okklusion dar.

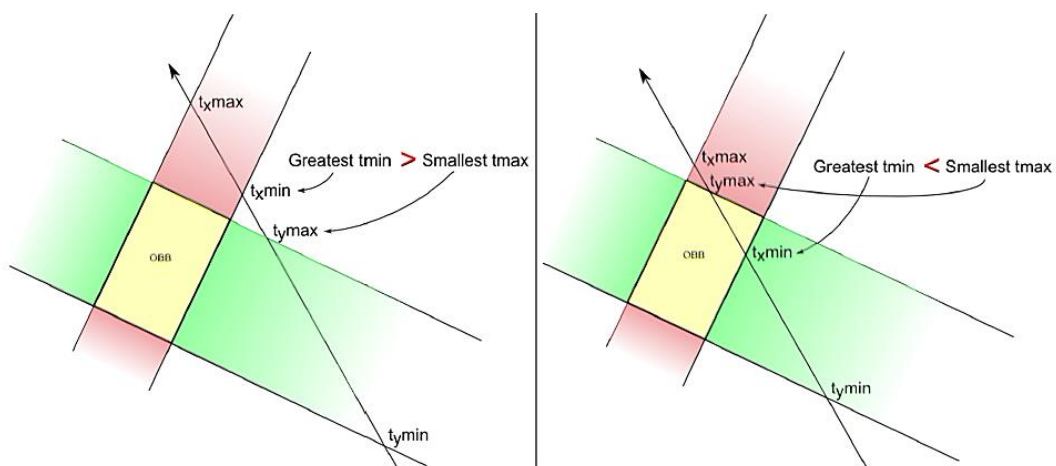


Abbildung 23: Verdeckungsberechnung durch das Detektionsmodell

Die Implementierung der Z-Achse und die Berücksichtigung der Okklusion betreffen die gesamte Simulationsstruktur von Baselabs Create: Der Zustandsraum "CASpace" (Constant Acceleration Space) muss zu "CASpace3D" erweitert werden. Darauf aufbauend müssen auch alle Sensormodelle (d.h. Messmodelle, Erkennungsmodelle und entsprechende Track Proposer) angepasst werden. Während das Messmodell und das Bewegungsmodell die Objekte immer noch in Form von "Objektlistenpunkten" betrachten, arbeitet das Erkennungsmodell auf der Grundlage der 3D BB der Objekte, um die Verdeckung und die Z-Position, die Orientierungswinkel und das vertikale FOV der Sensoren zu berücksichtigen.

Alle diese 3D-Anpassungen wurden in einer Bibliothek gesammelt. Um die 3D-Anpassungen in Baselabs Create zu verwenden, kann die Bibliothek mit dem spezifischen Baselabs-Projekt verknüpft werden, indem sie einfach als Referenz hinzugefügt wird.

Abbildung 24 zeigt als Beispiel die 3D-Version des Radarsensormodells in Baselabs Create.

Far Range Radar (Front)

⊙ Measurement Model	RadarMeasurementModel 3D	ⓘ
⊙ Detection Model	DetectionModel 3D	ⓘ
Field of view		
HorizontalAperture2	1,0472 rad	ⓘ
MaximumRange	250 m	ⓘ
VerticalAperture2	0,5236 rad	ⓘ
Mounting orientation		
SensorPitch2	0 rad	ⓘ
SensorRoll2	0 rad	ⓘ
SensorYaw2	0 rad	ⓘ
Mounting position		
SensorPositionX4	RadarPositionX (3,7 m)	ⓘ
SensorPositionY4	RadarPositionY (0 m)	ⓘ
SensorPositionZ	0,4 m	ⓘ
Sonstiges		
DetectionProbability2	0,9	ⓘ
Depends on	<input checked="" type="radio"/> State <input type="radio"/> Measurement	
⊙ Track Proposer	CrossTrafficRadarTrackProposer3D	ⓘ
⊙ General		

Abbildung 24: 3D-Erweiterung des Sensormodells mit vertikalem Blickfeld, Sensorausrichtung mit Nick- und Rollwinkel, Montagehöhe und Berücksichtigung der Verdeckung im Detektionsmodell

2.1.2.3.3.4 Evaluierungsmetriken und KPI

Die Evaluationsumgebung berechnet die KPIs "Erkennungsrate", "Fehlalarmrate", "Erkennungszeit" und "Unfallrisiko".

Die Erkennungs- und Fehlalarmrate wird berechnet, indem die Assoziation zwischen den simulierten Spuren und den Objekten der Bodenwahrheit innerhalb eines bestimmten Bereichs, z. B. eines Rechtecks um das Fahrzeug mit 600 m Länge und 300 m Breite, berücksichtigt wird. Wenn eine Spur mit einem Objekt der Bodenwahrheit assoziiert wird, steigt

die Erkennungsrate an. Wenn eine Spur keinem Objekt zugeordnet werden kann, trägt sie zur Fehlalarmrate bei.

Die Detektionszeit gibt den Zeitpunkt in der Simulationszeit an, an dem ein bestimmtes Objekt zum ersten Mal von der Anlage gesehen wurde. Mit diesem KPI lässt sich feststellen, welche Anlage schneller auf das Vorhandensein eines Objekts reagieren kann. Darüber hinaus erlaubt der KPI für das Unfallrisiko eine Einschätzung, ob dieser Zeitpunkt der Erkennung kritisch ist oder nicht. Wenn der Bremsweg des Ego-Fahrzeugs zum Zeitpunkt der ersten Erkennung größer ist als der Abstand zum Objekt, kommt es zu einem Unfall, wenn das Ego-Fahrzeug direkt auf dieses Objekt zufährt. In diesem Fall ist das Unfallrisiko R_{acc} , das durch das Verhältnis von Brems- und Objektabstand in Gleichung (1) dargestellt wird, größer als 1. Der Bremsweg wird mit der Geschwindigkeit v_{ego} und der Verzögerung a des Ego-Fahrzeugs berechnet. Es wird zwischen Komfort- ($a = -3,7 \text{ m/s}^2$) und Notbremsung ($a = -8 \text{ m/s}^2$) unterschieden.

$$R_{acc} = \frac{d_{braking}}{d_{object}} = \frac{v_{ego}^2}{2a \cdot d_{object}} \quad (1)$$

Zusätzlich zu diesen KPIs wurden drei Metriken implementiert. Ein Satz $G\{g_i\}$ an Ground-Truth-Objekten g_i für $i = 1, \dots, m$ sowie ein Satz $E\{e_j\}$ geschätzter Objekte e_j for $j = 1, \dots, n$ werden angenommen. Die Elemente dieser Sätze sind multidimensional inbegriffen der Objektposition, Richtung, Geschwindigkeit und Beschleunigung. Die euklidische Distanz $d(g_i, e_j)$ kalkuliert den Abstand zwischen dem tatsächlichen und den angenommenen Objekten.

Die Hausdorff Metrik $d_H(G, E)$ in (2) ist unempfindlich gegenüber unterschiedlichen Kardinalitäten und gewichtet Ausreißer stark.

$$d_H(G, E) = \max \left\{ \max_{g_i \in G} \min_{e_j \in E} d(g_i, e_j), \max_{e_j \in E} \min_{g_i \in G} d(g_i, e_j) \right\} \quad (2)$$

Das „Optimal SubPattern Assignment“ (OSPA) $\bar{d}_{OSPA}^{(c)}(G, E)$ in (3) bestraft Ausreißer weniger stark als die Hausdorff-Metrik, abhängig vom Grenzwertparameter c (here: $c = 2m$). Stattdessen verschlechtert sie sich, wenn ein Objekt der Grundwahrheit mehrere geschätzte Objekte enthält.

$$\bar{d}_{OSPA}^{(c)}(G, E) = \sqrt{\left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m d^{(c)}(g_i, e_{\pi(i)})^2 + c^2(n - m) \right) \right)} \quad (3)$$

Um die Positionszuverlässigkeit der geschätzten Objekte zu berücksichtigen, beinhaltet der Normalized Estimation Error Squared (NEES) in (4), auch bekannt als Mahalanobis-Abstand, den Positionsfehler $(g - e)$ und die Kovarianzmatrix P_{ge} .

$$NEES(g_i, e_j) = \sqrt{(g_i - e_j)^T P_{ge}^{-1} (g_i - e_j)} \quad (4)$$

2.1.2.3.4 Ergebnisse

Um die Ergebnisse und Analysemöglichkeiten der vorgestellten Auswertungsmethode zu demonstrieren, wurden zwei verschiedene Konfigurationen simuliert. Abbildung 25 zeigt deren Konfiguration. Beide bestehen aus einer vorderen und einer hinteren Kamera (blau) bzw. einem vorderen und hinteren Fernbereichsradar (FRR) (rot). Zusätzlich verfügt Aufbau A über vier Short Range Radars (SRR) an den Ecken (grün). Aufbau B verfügt stattdessen über ein SRR an jeder Fahrzeugseite (orange). Die wichtigsten Sensorparameter, z. B. FOV, Reichweite und Montagepositionen, sind in Tabelle 7 aufgeführt.

Setup	Sensor	Range	FOV	X	Y	Yaw	Pitch	Roll
A, B	Camera (front and rear)	150 m	80°	2,0 m / -0,8 m	0 m	0° / 180°	0°	0°
A,B	FRR (front and rear)	250 m	60°	3,7 m / -1,0 m	0 m	0° / 180°	0°	0°
A	SRRs corners (4x)	100 m	70°	3,5 m / -0,9 m	± 0,75 m	± 45° / ± 135°	0°	0°
B	SRRs sides (left and right)	100 m	70°	1,2 m	± 0,75 m	± 90°	0°	0°

Tabelle 7: Sensorparameter, Verbaupositionen und Alignmentwinkel für die Setups A und B (Ursprung der Koordinaten: Mitte der Heckachse)

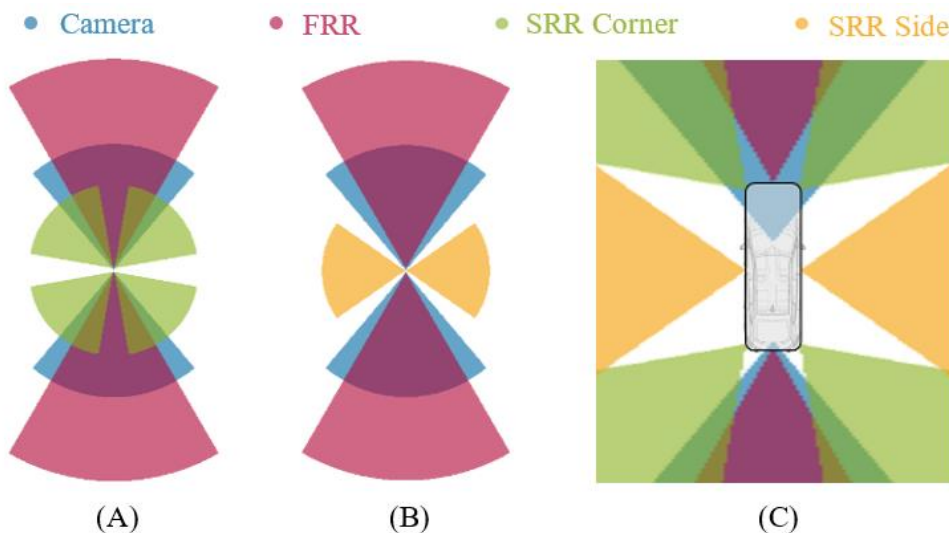


Abbildung 25: (A) Illustriert Aufbau A mit vier Short Range Radaren (SRR) an den Ecken, (B) zeigt Aufbau B mit einem SRR auf jeder Fahrzeugseite und (C) ist die detaillierte Ansicht auf die Sensorpositionierung der Setups A und B

Beide Szenarien wurden in einem Referenzszenario auf einer Landstraße $v = 100 \text{ km/h}$ mit zufälligem Verkehr auf der Gegenfahrbahn und in einem Stadtszenario simuliert, bei dem das Ego-Fahrzeug an einer Kreuzung nach rechts abbiegt, während ein Radfahrer den Weg kreuzt. Das Kreuzungsszenario ist in Abbildung 26 dargestellt. Das Ego-Fahrzeug wartet an einer roten Ampel an einer Kreuzung, um rechts abzubiegen. Ein Fahrradfahrer mit $v_B = 15 \text{ km/h}$ nähert sich aus der gleichen Richtung. Da die Ampel grün wird, überquert der Radfahrer die Kreuzung geradeaus, ohne seine Geschwindigkeit zu verringern. Das Ego-Fahrzeug fährt im gleichen Moment los, um mit $v_C = 10 \text{ km/h}$ nach rechts abzubiegen. Diese Situation ist sehr kritisch und verursacht viele schwere Unfälle. Ein hochautomatisiertes oder autonomes Fahrzeug muss in der Lage sein, solche Objekte zu erkennen und zu verfolgen, die in der Nähe des Fahrzeugs kreuzen.

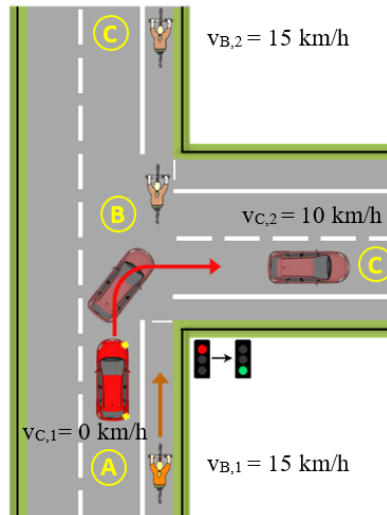


Abbildung 26: Kreuzungsszenario mit einem Rechtsabbieger und kreuzendem Radfahrer

Im Anschluss an die virtuelle Testfahrt werden der Datensatz der "Ground Truth" und der simulierte Sensordatensatz anhand der im vorherigen Abschnitt beschriebenen Metriken verglichen. Die Ergebnisse sind in Tabelle 8 dargestellt.

Metrik / KPI	Ländliches Szenario		Kreuzungsszenario (Rechtsabbieger und kreuzender Radfahrer)	
	Setup A	Setup B	Setup A	Setup B
Detektionsrate / Falschalarmrate [%]	68,69 / 0,35	68,96 / 0,15	94,62 / -	83,21 / -
Hausdorff [m]	309,13	310,07	84,90	79,01
OSPA [m]	1,73	1,73	1,44	2,00
Positionsfehler [m] / NEES [-]	2,31 / 4,67	2,27 / 4,68	1,19 / 15,72	11,17 / 95,12
Detektionszeit [s]	0,24	0,74	0	3,22
Unfallgefahr (komfort / Notfall) [-]	0,41 / 0,19	0,47 / 0,22	0,00 / 0,00	0,01 / 0,00

Tabelle 8: Ergebnisübersicht der Setups A und B bezüglich eines ländlichen Szenarios und eines Kreuzungsszenarios mit einem Rechtsabbieger und kreuzendem Radfahrer

Die Setups A und B zeigen sehr ähnliche Ergebnisse für das Landstraßenszenario. Der Grund dafür ist, dass in einem solchen Szenario die Unterschiede zwischen den beiden gewählten Setups (Eck-SRRs versus Seiten-SRRs) nicht sehr entscheidend für die Erkennung des Verkehrs auf der Gegenfahrbahn sind. Die Objekte nähern sich meist von vorne und werden daher von der Frontkamera und dem vorderen FRR erfasst. Bei einer Geschwindigkeit von 100 km/h befinden sich die vorbeifahrenden Objekte nur für wenige Millisekunden im Sichtfeld der Eck- oder Seiten-SRRs und tragen nicht wesentlich zur Gesamterkennungsleistung bei.

Ein Blick auf ein bestimmtes Objekt zeigt, dass Setup A es 0,5 Sekunden früher erfassen konnte als Setup B, was einem Unterschied von 27 m im verbleibenden Bremsweg entspricht. Der KPI für das Unfallrisiko zeigt jedoch, dass es sich in keinem der beiden Fälle um eine kritische Situation handelt, da der KPI auch bei einer Komfortbremsung unter $R_{acc} = 1$ bleibt.

Das Kreuzungsszenario ist relevanter, um die Leistungsunterschiede zwischen den gewählten Konfigurationen aufzuzeigen. Die Ergebnisse in Tabelle III zeigen, dass Setup A eine um 11 % höhere Erkennungsrate hat als Setup B. Obwohl die Hausdorff-Metrik im Vergleich zu Setup B um 7 % höher ist, d.h. dass der maximale Ausreißer einer Datenfusionsmessung höher war,

zeigt die OSPA-Metrik deutlich bessere Ergebnisse für Setup A. Es hat einen OSPA-Wert von 1,44 m, während Setup B den schlechtesten möglichen Wert von 2 m hat, da der Cutoff-Parameter c auf 2 m gesetzt wurde: Setup B hat fast 10-mal höhere Fehlerwerte. Darüber hinaus wurde der Radfahrer bei Einstellung A dank des SRR an der hinteren rechten Ecke sofort erkannt. Daraus ergibt sich eine Erkennungszeit von 0 Sekunden, und somit besteht keinerlei Unfallrisiko. Bei Einstellung B wurde der Radfahrer nach 3,22 s erfasst, als er das Sichtfeld des SRR auf der rechten Seite erreichte. Dies bedeutet, dass Setup B mehr als 3 Sekunden weniger Zeit hat, um rechtzeitig auf diesen Vorfall zu reagieren. Der KPI für das Unfallrisiko zeigt unkritische Ergebnisse, da das Ego-Fahrzeug gerade erst losgefahren ist und somit einen sehr kurzen Bremsweg hätte.

Das Diagramm in Abbildung 27 (links) zeigt die Objekterkennung der einzelnen Sensoren und ermöglicht eine genauere Analyse der Objektverfolgung und Redundanz. Aufbau A war in der Lage, den Fahrradfahrer von Anfang an kontinuierlich zu verfolgen. Die Lücke zwischen den FOVs des rechten hinteren SRR und des rechten vorderen SRR war klein genug, dass die Datenfusion die Spur weitergeben konnte. Die beiden vorderen Eck-SRRs trugen ebenfalls zur Redundanz zusammen mit der Frontkamera und dem vorderen FRR bei, was zu einer höheren Messgenauigkeit (Positionsfehler und NEES) führen konnte. Der vordere linke SRR verfolgte auch den Radfahrer nach der Rechtskurve des Fahrzeugs. Setup B zeigt nur zwei kurze Spuren, die von den seitlichen SRRs erzeugt werden und die die Spur der Frontkamera und des vorderen FRR verlängern. Es gibt eine Erkennungslücke zwischen der Messung der Frontkamera und der linken Seiten-SRR. Die Datenfusion versucht, die Spur für einen kurzen Zeitraum mit zunehmender Messunsicherheit beizubehalten und verliert schließlich die Spur für einen kurzen Moment, bevor sie durch den linken SRR eine neue Spur erzeugt. Dies kann auch in Abbildung 27 (rechts) beobachtet werden, die zeigt, dass der Positionsfehler in diesem bestimmten Zeitabschnitt stark ansteigt.

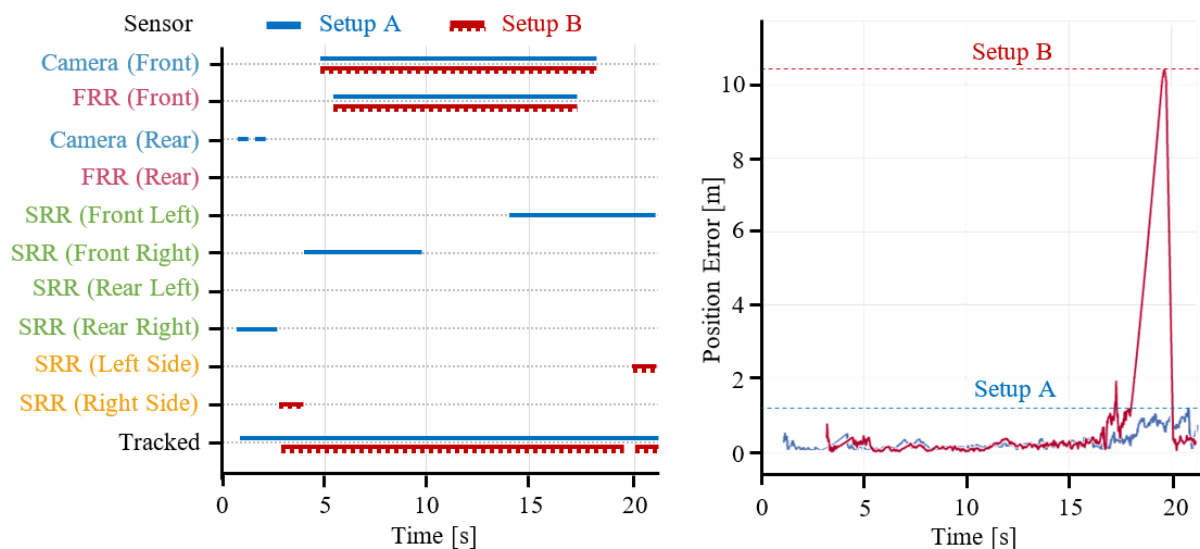


Abbildung 27: Objekterkennung (Radfahrer) durch jeden Sensor und Objektverfolgung für das Setup A und B (links) und Positionsfehler der Radfahrerposition für die Setups A und B (rechts)

Weitere Untersuchungen zur Wahrnehmungsqualität können für andere wichtige Fahrmanöver durchgeführt werden. Insbesondere für komplexere Sensoranordnungen ermöglicht dieses Verfahren eine systematische Analyse des Zusammenspiels der verschiedenen Sensoren und der Korrelationen innerhalb des Tracking-Teils der Wahrnehmung.

2.1.2.3.5 Schlussfolgerung

In dieser Arbeit wurde ein neues simulationsbasiertes Verfahren zur Bewertung von Sensoraufbauten für zukünftige automatisierte Fahrzeuge hinsichtlich bestimmter Sensoreinbaupositionen, diverser Aufbaukonfigurationen und unterschiedlicher Sensortechnologien entwickelt. Es besteht ein dringender Bedarf an einem Evaluierungsprozess, der zuverlässige und vergleichbare Ergebnisse entlang eines standardisierten Verfahrens in frühen Phasen der Fahrzeugentwicklung erzeugt. Das Framework erlaubt es, beliebige Sensoraufbauten zu konfigurieren und deren Wahrnehmungsqualität in virtuellen Testfahrten quantitativ mit Metriken und KPIs zu bewerten. Zwei verschiedene Setups wurden untersucht, um die Ergebnisse und Analysemöglichkeiten dieser Bewertungsmethode zu demonstrieren. Die Evaluationssuite ist ein wichtiges Instrument, das Automobilsystemingenieuren in der Konzeptphase der Fahrzeugentwicklung hilft und sie mit einem Leistungsüberblick für verschiedene Setups in relevanten Szenarien unterstützt.

Mit dem dargestellten Rahmenwerk ist es möglich:

- *neuartige, innovative und bessere Setup-Lösungen zu identifizieren*

Mit einem zeit- und kosteneffizienten Evaluationsverfahren kann eine größere Anzahl möglicher Setup-Konstellationen untersucht werden. Außerdem erlaubt die Simulation die Berücksichtigung riskanter Fahrmanöver, die in der Realität nicht getestet werden können.

- *Beschleunigung der Entwicklung von automatisierten Fahrzeugen*

Wenn der Fahrzeugentwicklungsprozess bereits mit einem fundierten Setup-Konzept beginnt, spart dies Optimierungszeit bei den nachfolgenden Entwicklungsschritten.

- *Rechenleistung sparen*

Durch die Reduzierung der Anzahl der Sensoren während der Setup-Optimierung können Steuergeräte und Stromversorgung profitieren.

- *Reduzierung der Kosten für das gesamte Sensorsystem*

Kosten-Nutzen-Analysen können die günstigste Setup-Lösung aufzeigen, die dennoch alle Anforderungen erfüllt.

- *Erstellung einer Roadmap für Sensortechnologien*

Durch die Untersuchung der Zusammenhänge innerhalb der Sensordatenverarbeitungskette lassen sich Anforderungsprofile im Sinne einer Roadmap für zukünftige Sensortechnologien ableiten.

2.2 Erfolgte und geplante Veröffentlichungen

- I. M. Hartstern, V. Rack und W. Stork, "Towards an Evaluation Methodology for the Environment Perception of Automotive Sensor Setups," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Towards Cognitive Vehicles (TCV) Workshop, Macau, China, 8th November 2019. Proceedings: <https://cogvehicles2019.github.io/assets/proceedings/proceedings.pdf>, S. 27–30.
- II. M. Hartstern, V. Rack, M. Kaboli und W. Stork, "Simulation-based Evaluation of Automotive Sensor Setups for Environmental Perception in Early Development Stages," 31st IEEE Intelligent Vehicles (IV) Symposium, 2020. DOI: 10.1109/IV47402.2020.9304771
- III. M. Hartstern, V. Rack und W. Stork, "Conceptual Design of Automotive Sensor Systems: Analyzing the impact of different sensor positions on surround-view coverage," IEEE SENSORS, 2020. DOI: 10.1109/SENSORS47125.2020.9278638
- IV. N. Druml et al., "Programmable Systems for Intelligence in Automobiles (PRYSTINE): Technical Progress after Year 2," 23rd Euromicro Conference on Digital System Design (DSD), 2020. DOI: 10.1109/DSD51259.2020.00065
- V. AutoSens Brüssel 2020: Maike Hartstern, „Simulation-based Evaluation of Sensor Setups for Highly Automated Driving – How to define optimal sensor setups for future vehicle concepts“, 16. September 2020. <https://auto-sens.com/speakers-brussels/> [Jahr: 2020]

3 Abbildungsverzeichnis

Abbildung 1: Gesamtvorhaben PRYSTINE, gegliedert in Arbeitspakete (WP) und Supply Chains (SC).....	6
Abbildung 2: Weiterentwicklungspfad der E/E-Architekturgeneration bei BMW und mögliche Umsetzungspfade im Jahr 2025, die über PRYSTINE erarbeitet und verifiziert werden	9
Abbildung 3: Arbeitspaketstruktur im Projekt PRYSTINE als V-Modell	11
Abbildung 4: Regeln zur Formulierung von Anforderungen und Qualitätskriterien, Auszug aus dem Regelkatalog „Fit for Requirements“ [6].....	11
Abbildung 5: Vorlage zur Formulierung von Anforderungen, Auszug aus dem Regelkatalog „Fit for Requirements“ [6].....	12
Abbildung 6: Sensormodell Vorlagen in der Simulationsumgebung Baselabs Create (rot) und neue LiDAR Sensormodell Vorlage von BMW (blau)	15
Abbildung 7: Einstellungen zum Mess- und Detektionsmodell des neuen LiDAR Sensormodells	16
Abbildung 8: Einstellungen zum Fahrbahnführungsverfahren des neuen LiDAR Sensormodells	17
Abbildung 9: Beschreibung der Polarkoordinaten des neuen LiDAR Sensormodells (links) und Integration der Objektverdeckung innerhalb des Detektionsmodells für das neue LiDAR Sensormodell.....	18
Abbildung 10: Visualisierung des Sichtfelds vom LiDAR Sensor während einer aktiven Sensorsimulation.....	18
Abbildung 11: Zusammenhang Automatisierungsgrad, Funktionsset und Sicherheitsanforderungen	19
Abbildung 12: Prinzipdarstellung der Sensoranbindung bei unterschiedlichen Automatisierungsgraden	20
Abbildung 13: Zunahme der Komplexität des Sensoraufbaus im Zusammenhang mit der Entwicklung der Fahrautomatisierung.....	21
Abbildung 14: Schema der Datenverarbeitungskette in automatisierten Fahrzeugen.....	22
Abbildung 15: Fahrzeugentwicklungsprozess entsprechend des V-Modells.....	23
Abbildung 16: Abstraktionsstufen von Sensormodellen.....	25
Abbildung 17: Schematische Visualisierung des vorgeschlagenen Simulationsframeworks und Arbeitsmodells für die Konfiguration und Evaluation von Sensorsetups in Fahrzeugen	27
Abbildung 19: Heatmap mit Interessensregionen und korrespondierenden Kritikalitätsindizes ..	30
Abbildung 19: Szenarien der Szenariogruppe „Kreuzung“	31
Abbildung 20: Konfiguration virtueller Sensorsetups mit unterschiedlichen Sensormodellen in Baselabs Create	32
Abbildung 21: Messmodell, Detektionsmodell und Track Proposer Einstellungen der Sensormodelle	33
Abbildung 22: Verdeckungsberechnung innerhalb des Detektionsmodells in Baselabs Create	34
Abbildung 23: Verdeckungsberechnung durch das Detektionsmodell.....	34
Abbildung 24: 3D-Erweiterung des Sensormodells mit vertikalem Blickfeld, Sensorausrichtung mit Nick- und Rollwinkel, Montagehöhe und Berücksichtigung der Verdeckung im Detektionsmodell.....	35
Abbildung 25: (A) Illustriert Aufbau A mit vier Short Range Radaren (SRR) an den Ecken, (B) zeigt Aufbau B mit einem SRR auf jeder Fahrzeugseite und (C) ist die detaillierte Ansicht auf die Sensorpositionierung der Setups A und B.....	37
Abbildung 26: Kreuzungsszenario mit einem Rechtsabbieger und kreuzendem Radfahrer .	38
Abbildung 27: Objekterkennung (Radfahrer) durch jeden Sensor und Objektverfolgung für das Setup A und B (links) und Positionsfehler der Radfahrerposition für die Setups A und B (rechts)	39
Abbildung 28: Ursprüngliche Zeitplanung des Teilvorhabens in Work Package 1, 2 und 7 ..	45

4 Tabellenverzeichnis

Tabelle 1: Übersicht Aufgabenpakete in Work Package 1	6
Tabelle 2: Übersicht Aufgabenpakete in Work Package 2	7
Tabelle 3: Vorgabedokument zur Anforderungserfassung mit Automatisierungsfunktion.....	13
Tabelle 4: Simulationswerkzeuge für die Entwicklung von Automobilsystemen.....	24
Tabelle 5: Objektfaktoren (F) für diverse Objekte.....	30
Tabelle 6: von einer ROI Heatmap abgeleitete relevante Szenarien zur Evaluation von Sensorsetups	31
Tabelle 7: Sensorparameter, Verbaupositionen und Alignementwinkel für die Setups A und B (Ursprung der Koordinaten: Mitte der Heckachse)	37
Tabelle 8: Ergebnisübersicht der Setups A und B bezüglich eines ländlichen Szenarios und eines Kreuzungsszenarios mit einem Rechtsabbieger und kreuzendem Radfahrer	38

5 Literaturverzeichnis

- [1] P. Vetter, „welt.de,“ 14. 09. 2015. [Online]. Available: <https://www.welt.de/wirtschaft/article146385922/Google-baut-das-Auto-und-Daimler-liefert-zu.html>. [Zugriff am 24. 11. 2020].
- [2] M. Ramsey und H. Varnholt, „finanzen.net,“ 07. 01. 2016. [Online]. Available: <https://www.finanzen.net/nachricht/aktien/update-was-die-autoindustrie-auf-der-ces-zum-thema-autonomes-fahren-sagt-4677631>. [Zugriff am 24. 11. 2020].
- [3] Bundesregierung, *Strategie automatisiertes und vernetztes Fahren*, Berlin: Bundesregierung, 2015.
- [4] B. f. W. u. Energie, *Hochautomatisiertes Fahren auf Autobahnen*, Berlin, 2015.
- [5] Matthias Traub, “Enabling connected mobility with an innovative e/e-architecture - Seamless use of IT standards pave the way into the digital automotive future”, Vector Congress 2017, Stuttgart.
- [6] BMW AG, *Fit for Requirements*, München, 2012.

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel Verbundprojekt: Hoch zuverlässige Elektroniksysteme für intelligente Fahrzeuge – PRYSTINE –; Teilvorhaben: Ausfallsichere Umfelderkennung für hochautomatisierte Fahrzeuge	
4. Autor(en) [Name(n), Vorname(n)] Rack, Viktor	5. Abschlussdatum des Vorhabens 30.04.2021
	6. Veröffentlichungsdatum
	7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) Bayerische Motoren Werke Aktiengesellschaft 80788 München	9. Ber. Nr. Durchführende Institution
	10. Förderkennzeichen 16ESE0331
	11. Seitenzahl 45 Seiten
12. Fördernde Institution (Name, Adresse) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. Literaturangaben 6 Referenzen
	14. Tabellen 8 Tabellen
	15. Abbildungen 28 Abbildungen
16. Zusätzliche Angaben	
17. Vorgelegt bei (Titel, Ort, Datum) Technische Informationsbibliothek, Welfengarten 1B, 30167 Hannover BMBF Bibliothek Forschung und Technologie, Heinemannstraße 2, 53175 Bonn	
18. Kurzfassung Das Ziel von PRYSTINE war, eine ausfallsicheres Urban Surround perceptiON (FUSION)-Funktion zu realisieren, die auf einer robusten Kamera-, Radar- und LiDAR-Sensorfusion sowie auf Steuerungsfunktionen basiert, um ein sicheres automatisiertes Fahren in städtischer und ländlicher Umgebung zu gewährleisten. Im Rahmen des Gesamtvorhabens PRYSTINE lag der Schwerpunkt des Teilvorhabens der BMW Group auf der ausfallsicheren Umfelderkennung für hochautomatisierte Fahrzeuge. Fokus der Arbeiten war dabei die Definition von Anforderungen und Spezifikationen aus Sicht der Automobilindustrie. Ebenso hat BMW zum System Level Design und zur simulativen Analyse der eingebetteten Sensor- und Steuerungskomponenten, der E/E-Architektur, sowie der Auslegung und Bewertung des Gesamtsensorverbunds beigetragen. Das Anforderungsmanagement wurde unter anderem mit Methoden wie SCRUM oder SMArDT unterstützt. Für die Untersuchung und Bewertung von Sensoren im Gesamtverbund konnte Baselabs Create als Tool um weitere Funktionen weiterentwickelt und zusätzliche KPI's eingeführt werden. Damit wurde es möglich, einzelne Sensoren, aber auch den Sensorverbund bereits in der Anforderungs- und Spezifikationsphase zu evaluieren und damit alle Parameter für die weitere Sensorentwicklung zu liefern. Innerhalb von nur wenigen Jahren wird vernetztes und automatisiertes Fahren die individuelle Mobilität revolutionieren, indem es sicherere, sauberere und stärker nutzerorientierte Dienste anbietet, die sich gegenseitig auf lange Sicht verstärken. In diesem Zusammenhang entstehen E-Mobilität, autonomes Fahren und neue Geschäftsmodelle, insbesondere in den Bereichen digitaler Dienste, in denen die Konnektivität und Verfügbarkeit von Echtzeitdaten für die BMW Group von zentraler Bedeutung ist.	
19. Schlagwörter Hochautomatisiertes Fahren, sicherheitskritisch, ausfallsicher, zuverlässig, intelligent, Umfelderkennung, Elektroniksysteme, Perzeption, Sensorverbund	
20. Verlag	21. Preis

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report
3. title Joint Research Project: Programmable Systems for Intelligence in Automobiles – PRYSTINE –; Partial intention: Fail-safe environment perception for high automated driving	
4. author(s) (family name, first name(s)) Rack, Viktor	5. end of project 30.04.2021
	6. publication date
	7. form of publication
8. performing organization(s) (name, address) Bayerische Motoren Werke Aktiengesellschaft 80788 München	9. originator's report no.
	10. reference no. 16ESE0331
	11. no. of pages 45 pages
12. sponsoring agency (name, address) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. no. of references 6 references
	14. no. of tables 8 tables
	15. no. of figures 28 figures
16. supplementary notes	
17. presented at (title, place, date) Technische Informationsbibliothek, Welfengarten 1B, 30167 Hannover BMBF Bibliothek Forschung und Technologie, Heinemannstraße 2, 53175 Bonn	
18. abstract The target of PRYSTINE was to develop a fail-safe urban surround perception (FUSION) function that is based on a robust camera, Radar and LiDAR sensor fusion but also control functions to ensure a safe automated driving in urban and rural areas. As part of the overall project PRYSTINE, the main emphasis of the BMW intention was to create a fail-safe environment perception for high automated vehicles. Focus of the work was the definition of requirements and specification from the automotive industries point of view. Also, BMW was able to contribute to the system level design and the simulative analysis of embedded sensor and control components, the E/E architecture but also the design and evaluation of integrated sensor setups. Among others, the requirements management was supported by methods like SCRUM or SMARDT. For the analysis and evaluation of integrated sensor setups the tool Baselabs Create could be used and extended with further functionality and KPIs. Thereby it was possible to evaluate single sensors but also sensor cluster and provide all the parameters for further sensor development already in the requirements and specification phase. In only a few years connected and automated driving will revolutionize the individual mobility by offering safe, clear and more user-oriented services that gain themselves in the long term. In this connection E-mobility, autonomous driving and new business models will rise, especially in the area of digital services, where connectivity and availability of real-time data is crucial for the BMW Group.	
19. keywords High automated driving, safety-critical, fail-safe, dependable, intelligent, environment perception, electronic systems, sensor cluster	
20. publisher	21. price