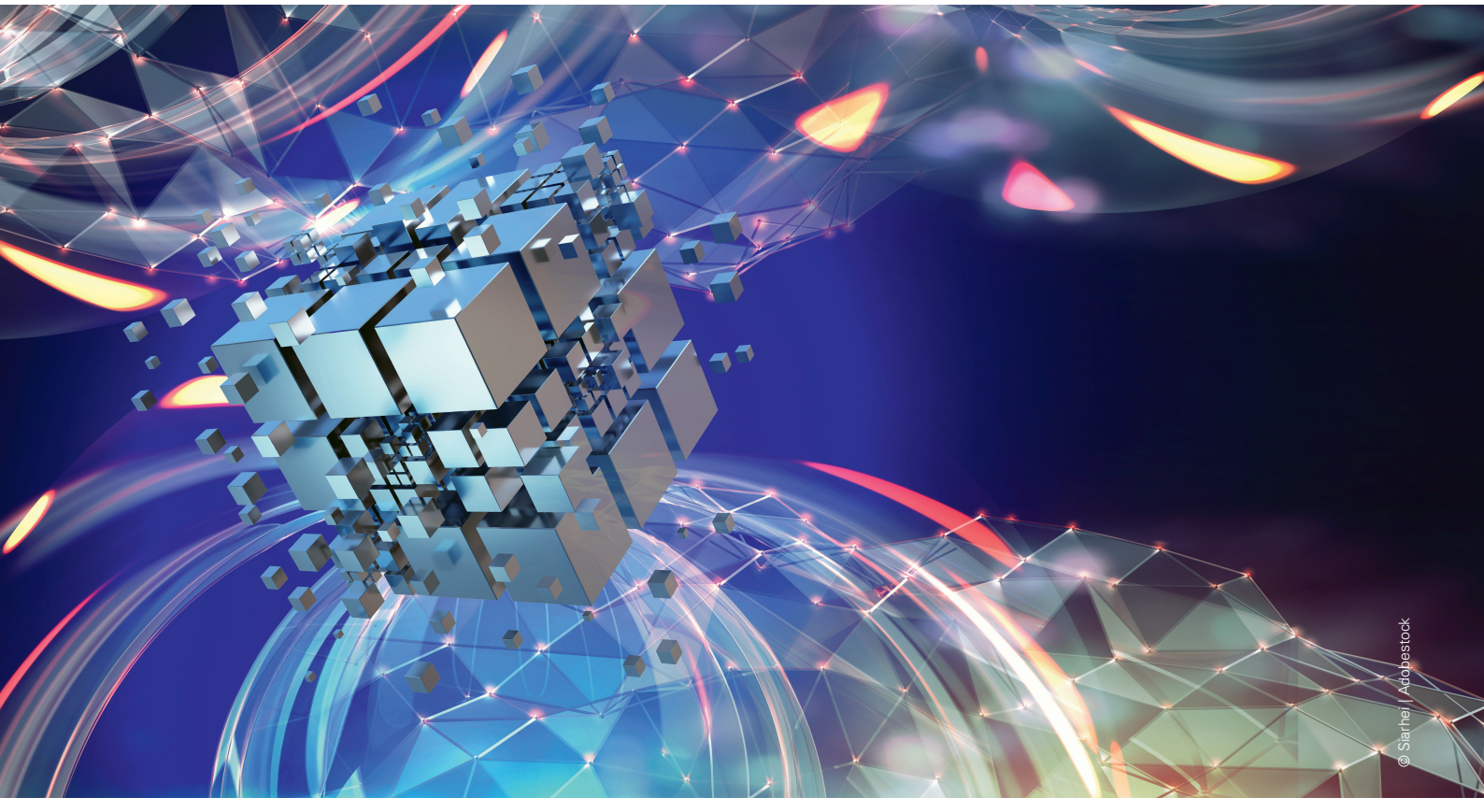
Autonomes Fahren dank moderne Software-Systeme sicher einführen

Zeitnahe Zukunftsmusik

Autonome, Software-definierte Fahrzeuge sind keine allzu ferne Zukunftsmusik mehr. Auf dem Weg dorthin, gilt es noch Herausforderungen zu bewältigen. In diesem Artikel geht es um die Frage, wie sich sichere Software zur Steuerung autonomer Fahrzeuge ermöglichen lässt.

Sabahudin Husic



Vernetzte Software-gesteuerte Fahrzeuge werden die zukünftige Mobilität grundlegend verändern. Angesichts diverser erfahrener Unternehmen im Automobilsektor, die neuen innovativen Technologien zum Durchbruch verhelfen, ist es nur eine Frage der Zeit, bis selbstfahrende Autos ein vertrauter Anblick sein werden [1]. Dieser Weg in die Zukunft ist jedoch noch voller Herausforderungen, von der zunehmenden Komplexität der Automobil-Software und der allmählichen Entkopplung der Hardware- und Software-Komponenten bis zum zunehmenden Bedarf an methodischen, sicherheitsbasierten Konzepten und Produktionsabläufen, die eine sichere Umsetzung zukunftsweisender Technologie voranbringen können.

Autonomiestufen

Es wurden fünf Stufen in der Entwicklung des autonomen Fahrens definiert, die jeweils beschreiben, inwieweit ein Fahrzeug Aufgaben und Verantwortlichkeiten des Fahrers übernimmt und wie Mensch und Maschine interagieren [2]. Die fünf Stufen der Fahrzeugautomatisierung umfassen:

- Level 0: Nur Fahrer
  Das System bietet eine vorübergehende Fahrassistenz in Form von Warnhinweisen und Benachrichtigungen und Sicherheitseingriffen im Notfall an, während der Fahrer beschäftigt und aufmerksam bleibt.
- Level 1: Fahrerunterstützung

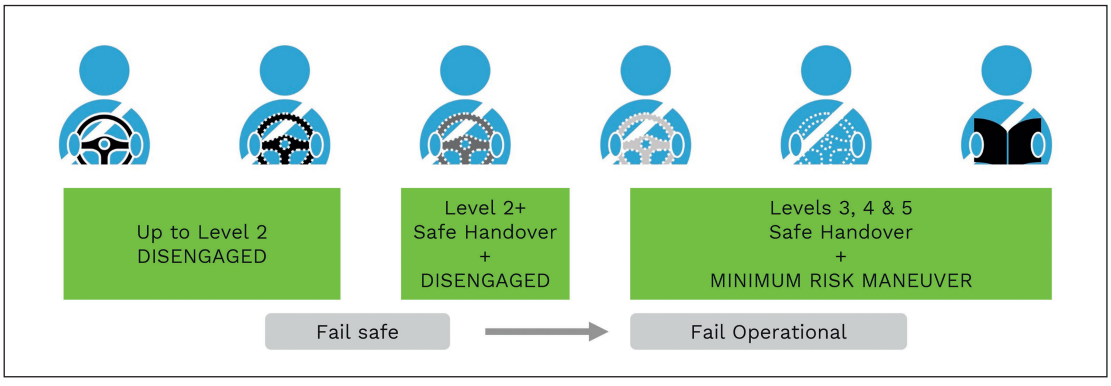


Bild 1: Die Entwicklung von Ausfall- zu Betriebssicherheit umfasst kritische Schritte mit dem Schwerpunkt Fehlererkennung, Redundanz und schrittweiser sicherer Stopp.
© KPIT Technologies

Das System hilft dem Fahrer, übernimmt aber nicht seine Aufgaben.
- Level 2: Teilautomatisiertes Fahren
  Das System kann Steuerungsfunktionen übernehmen, aber die Verantwortung für die Fahrzeugoperationen bleibt beim Fahrer.
- Level 3: Hochautomatisiertes Fahren
  Der Fahrer kann sich in bestimmten Situationen für einen bestimmten Zeitraum von seinen Fahrzeugführungsaufgaben freimachen.
- Level 4: Vollautomatisiertes Fahren
  Das Fahrzeug kann die meiste Zeit über selbständig fahren, während der Fahrer sich mit anderen Dingen beschäftigt, aber jederzeit bei Bedarf in der Lage ist, wieder die Kontrolle zu übernehmen.
- Level 5: Vollständige Automatisierung

Das Fahrzeug übernimmt die vollständige Steuerung der Fahrfunktionen, während die Insassen reine Fahrgäste sind. Diese Stufen lassen sich realisieren, wenn bestimmte Voraussetzungen für den Bau solcher Fahrzeuge erfüllt sind. Die aktuelle Architektur vernetzter Fahrzeuge beinhaltet AUTOSAR-Basissoftware. Diese funktioniert zwar effizient mit kleinen Mikrocontrollern und ist gut geeignet für zeitkritische, sichere und geschützte Anwendungen, aber den vorgesehenen Zweck erfüllt sie nicht vollständig. Die meisten dieser Funktionen priorisieren Komfort gegenüber Sicherheit und entsprechen daher nicht einer effektiven und auf methodischen Anforderungen basierenden Gestaltung und geeigneten Validierungs- und Verifizierungsprüfungen. Während für solche Fahrzeuge eine hohe Rechenleistung erforderlich ist, benötigt auch die integrierte Infrastruktur etwa 25 GB pro Stunde und verwendet fast 40 Mikroprozessoren und dutzende Sensoren zur Datenerhebung [3].

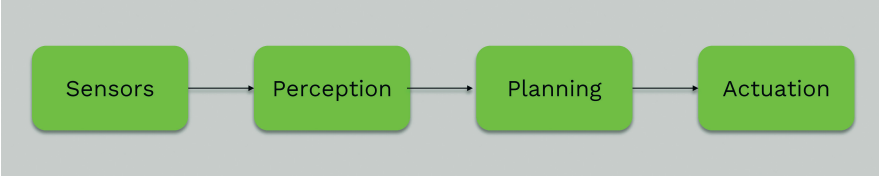


Bild 2: Die vier Säulen betriebssicherer Systeme © KPIT Technologies

Voraussetzungen für den Bau zukunftsfähiger Fahrzeuge

KPIT strebt eine Service-orientierte Architektur als solide Grundlage für vernetzte Fahrzeuge an. Mit aufgeteilten Motorsteuergeräten im Sicherheits-I/O- und Hochleistungs-Controller sowie dedizierter Infrastruktur zur Ermöglichung hoher Rechenleistung nutzt das KPIT-Rahmenwerk POSIX-artiges Betriebssystem, beispielsweise Linux, adaptives AUTOSAR und einen I/O-Controller, der Sensor- und Stellantriebsleistungen bereitstellt, zusammen mit einem Embedded-Betriebssystem wie Classic AUTOSAR.

Während der Performance-Controller den Fahrer unterstützt und sich auf die Nicht-Hochgeschwindigkeitsfunktionen konzentriert, kümmert sich der I/O-Controller um die Hochgeschwindigkeitsfunktionen, die Filterverarbeitung und die Berechnungen mit präziser Zeitplanung. Die Software unterscheidet sich im Hinblick auf Einschränkungen der verfügbaren Funktionen und die angeschossenen Sensoren/Stellantriebe. Die Vorzüge des Performance Controllers – die Anfrage von Daten nach Bedarf (SOME/IP) – werden über Funk aktualisiert und können neue Funktionen und Fehlerbereinigungen umfassen und sich gegenseitig ersetzen (betriebssicher). Alle genannten Elemente sind wesentlich, um sowohl die Systemintegrität und Verfügbarkeit für ein umfangreiches Software-System als auch autonomes Fahren zu ermöglichen.

Konzepte für Fail-operational-Systeme

Eines der Kernprinzipien bei der Konzeption und Entwicklung vernetzter Fahrzeuge war von Beginn an die Ausfallsicherheit der elektronischen Steuerungssysteme. Ausfallsicherheit ist heute unzureichend und die Betriebssicherheit stellt sich als absolute Mindestanforderungen heraus.
Der Weg von Ausfallsicherheit zu Betriebssicherheit, wie er in **Bild 1** schematisch dargestellt ist, enthält folgende Elemente:
- Deaktivierung/Einschränkung des funktionssicheren Statusumfangs
  – Den Fahrer informieren
  – Diagnosefehler melden
- Das Standardkonzept in zahlreichen sicherheitsrelevanten Systemen umfasst
  – Airbag, ESP, Klimaanlage, Batterieaufladung
  – Fahrerassistenz-Funktionen wie adaptiver Tempomat und Fahrspurassistent
- Einige Funktionen bieten einen eingeschränkten Notfallmodus, der

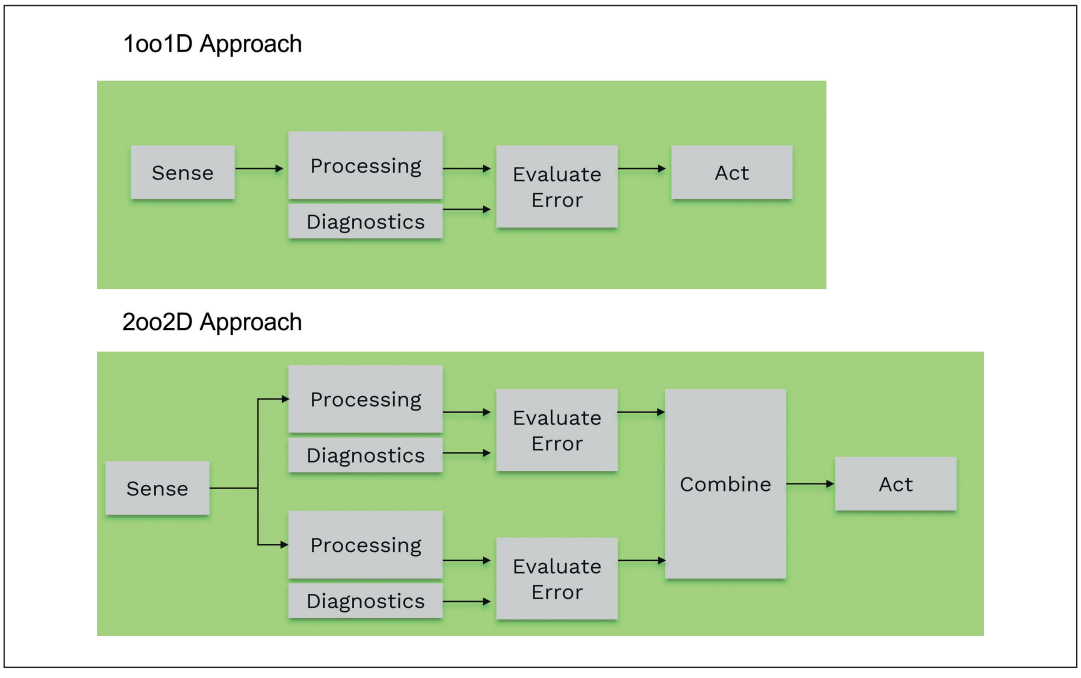


Bild 3: Vergleich zwischen zwei Ansätzen zur Einführung einer internen Diagnose in die Architektur autonomer Fahrzeuge

© KPIT Technologies

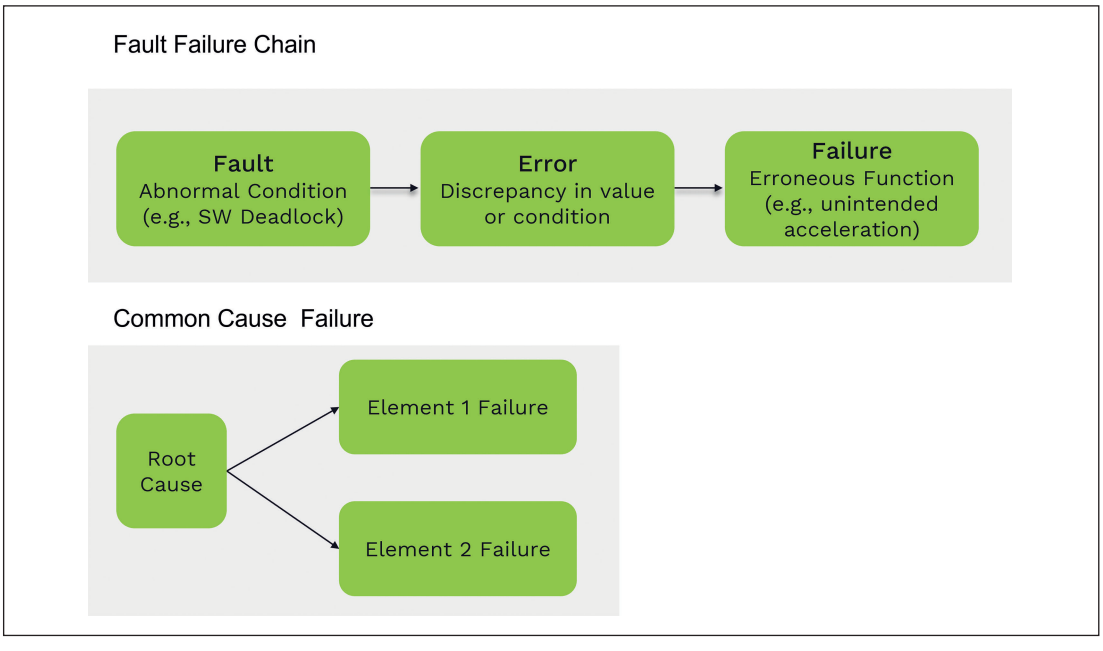


Bild 4: Analyse von Störungen, die zu einem Ausfall in einem sicherheitskritischen System führen

© KPIT Technologies

manchmal zeitlich begrenzt ist. Dazu zählen:
- Elektronische Servolenkung
- Bremsen

In hochgradig automatisieren Fahrzeugen sind sicherheitskritischen Systemen wie diesen oft dedizierte Systeme zugewiesen, die ihnen helfen, die genannten Funktionen wahrzunehmen, wenn eine Störung entdeckt wird. Das sorgt für die angestrebte Redundanz und ermöglicht dem Fahrzeug auch weiterzufahren, bis es selber zu einem sicheren Halt kommt.

Der sichere Zustand impliziert:

- Fortsetzung der Fahrt, bis der Fahrer in der Schleife ist
  - Ca. 7 bis 15 s bei bedingt autonomem Fahren
  - Mehrere Minuten bei hochgradig und vollständig autonomen Fahren
- Durchführung eines autonomen „sicheren Halts", also dem Stillstand an einem ungefährlichen Ort
  - Lenkung der Aufmerksamkeit des Fahrers auf die Situation
  - Kann je nach Situation mehrere Minuten dauern

Betriebssichere Systeme ermöglichen ein ganzheitliches Modell der Wirkungskette eines autonomen Fahrzeugs besteht aus Sensorelementen wie Radaren, Kameras und anderen Sensoren, die der Wahrnehmungsschicht zugeführt werden, die Sensoreingänge in einer überschaubaren Darstellung der Umgebung um das Fahrzeug verortet. Die dargestellte Umgebung wird dann in Planungskomponenten verwendet, um eine ausführbare Bahn für das Steuermodul zu erstellen (**Bild 2**).

Unter Sicherheitsgesichtspunkten ist es offensichtlich, dass die einfache Simpleximplementierung unzureichend ist, um ausfallsichere oder betriebssichere

Funktionen zu realisieren und dass zusätzliche Gestaltungselemente erforderlich sind, um Fehler zu erfassen und zu verhindern, dass Ausfälle das System in einen unsicheren Status bringen [4].
Das Konzept der Einführung einer internen Diagnose in ein System, sodass das System bei Erkennung eines Fehlers auf eine sichere Weise ausfallen kann, ist in **Bild 3 oben** dargestellt, wenn auch klar ist, dass sich mit einer 1oo1D-Architektur keine ausfall- und betriebssicheren Kriterien für hochwertige Sicherheitssysteme wie autonomes Fahren realisieren lassen. Eine Lösung, die in der Automobilindustrie immer beliebter wird, ist die Two-out-of-two-Architektur mit Diagnose (2002D), bei der zwei 1oo1D-Kanäle parallel betrieben werden, um maximale Verfügbarkeit sicherzustellen (**Bild 4**). Bei Ausfall eines Kanals, ist der andere weiterhin verfügbar, um den laufenden Betrieb zu gewährleisten. Der beim 2oo2D-Konzept zu berücksichtigende Faktor ist die Sicherstellung der Unabhängigkeit der Kanäle, um das System vor gemeinsam verursachten Ausfällen zu schützen. Ein gemeinsam verursachter Ausfall ist ein Szenario, bei dem eine Grundursache eine Abweichung in beiden Kanälen verursacht, sodass beide ausfallen. Gängige Lösungsansätze für dieses Problem sind die Verwendung unterschiedlicher Sensorsätze in verschiedenen Kanälen, die Verwendung unterschiedlicher Implementierungen in kritischen Teilen (wie Math Library, Fusionsalgorithmus und ähnliche Elemente), die Diversifizierung der Hardware-Plattform und andere Koppelfaktoren.

Fail-operational-Systeme gewährleisten

Vernetzte Fahrzeuge haben seit den 1980er Jahren hinsichtlich Design, Infrastruktur und Technologie einen weiten Weg zurückgelegt. Heute schickt sich dieses Konzept an, zukunftsfähig zu werden – mit deutlichem Schwerpunkt auf Sicherheit. Der Denkprozess von KPIT für autonome Fahrzeuge umfasst die Wiederverwendung verfügbarer Integritätsmechanismen von ausfallsicheren Systemen als Grundlage für den Aufbau betriebssicherer Systeme. Während Softwaresysteme zum Erzielen einer hohen Diagnoseabdeckung heute jederzeit verfügbar sind, müssen diese zukünftig in eine robuste Konstruktion integriert werden, um die autonomen Fahrfunktionen zu gewährleisten. ■ (eck)

www.kpit.com

Literatur

[1] https://www.brookings.edu/research/securing-the-future-of-driverless-cars/

[2] https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[3] https://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/

[4] https://www.forbes.com/sites/samabuelsamid/2022/02/08/automated-driving-must-be-fail-operational/?sh=24ef7b2a527f

Sabahudin Husic ist Subject Matter Expert Autonomous Driving bei KPIT Technologies.