

## INTERVIEW MIT THORSTEN GERKE, MATHWORKS

# Modellbasierter Wandel der Mobilität

Allein schon aus Gründen der Komplexität ist die konsequente Nutzung und Anwendung modellbasierter Entwicklung spätestens bei der Entwicklung der Fahrzeuge von morgen ein absolutes Muss. AUTOMOBIL-ELEKTRONIK hat sich mit Thorsten Gerke, Automotive Industry Manager EMEA bei Mathworks, über diverse Themen unterhalten: von Business-Aspekten über modellbasierte Entwicklung, Sensordatenfusion, Cloud-Anbindung und virtuelles Testen bis hin zur Datensicherheit.

Interviewer: Alfred Vollmer

## Herr Gerke, wie läuft das Automotive-Geschäft?

**Thorsten Gerke:** Eines vorweg: Automotive ist das größte Marktsegment von Mathworks. Unser Geschäft läuft seit vielen Jahren gut, und der Trend ist weiterhin rundum positiv. Der Hauptgrund dafür ist, dass wir über viele Jahre hinweg eine enge Partnerschaft zu automotiven OEMs und Zulieferern aufgebaut haben, und unsere Kunden benötigen zuverlässige Tools, die es ihnen ermöglichen, qualitativ sehr gute und hochwertige Produkte in immer kürzeren Entwicklungszeiten zu entwickeln.

## Was heißt das in Zahlen ausgedrückt?

**Thorsten Gerke:** Da wir ein Unternehmen mit Privateigentümern sind, veröffentlichen wir keine Geschäftszahlen. Diese Eigentümerstruktur kommt uns sehr zugute, denn weil wir keine Abhängigkeiten von Shareholdern haben, können wir viel langfristiger denken, planen und agieren. Wir analysieren sehr genau, in welche Richtungen sich die Industrie langfristig entwickeln kann und welche Prioritäten wir in den nächsten Jahren setzen wollen und definieren darauf basierend unsere Ziele und verfolgen diese dann ganz konsequent.

## Welche Schwerpunkte setzen Sie im Automotive-Bereich?

**Thorsten Gerke:** Wir richten unsere Schwerpunkte immer an den wichtigen Trends und langfristigen Entwicklungen der Industrie aus. Hauptaktivitäten der Autoindustrie beziehen sich auf die Bereiche Elektrifizierung des Antriebs, ADAS & autonomes Fahren, Funktionale Sicherheit, Embedded Security; und auch der Bereich Big Data spielt mittlerweile eine große Rolle in der Autoindustrie. Mathworks unterstützt unsere Kunden bei der Bewältigung dieser Aufgaben domänenübergreifend. Zum einen durch die modellbasierte Entwicklung mit Simulink für regelungstechnische Systeme in Verbindung mit automatischer Codegenerierung für die Entwicklung von Embedded-Software. Die eigentliche Mehrwertfunktion im Auto steckt heute in der Embedded-

Software, und somit ist es wichtig, hohe Softwarequalität bei immer kürzer werdenden Entwicklungszeiten und unter hohem Kostendruck modellbasiert zu entwickeln.

Über die verschiedenen Anwendungstrends hinweg sieht sich die Autoindustrie aber zunehmend auch mit großen Datenmengen konfrontiert. Diese stellen ein beträchtliches Potential für die Autofirmen dar, aus denen Informationen und Wissen extrahiert werden muss, um die Produktentwicklung zukünftig noch stärker an den Wünschen des Kunden zu orientieren. Hierzu entwickeln Autofirmen Algorithmen in Matlab, um diese Informationen aus den riesigen Datenmengen zu gewinnen. Ein gutes Beispiel ist der Bereich der Umfeldsensorik für ADAS und automatisiertes Fahren. Hier werden immense Datenmengen produziert, die analysiert werden müssen. Bei geschickter Nutzung beziehungsweise Auswertung dieser Daten können die Ingenieure diverse Modelle für die Vorhersage ableiten. Automatische Fahrzeugdetektion und Fußgängererkennung mittels Deep Learning sind da nur einige Bereiche, bei denen es um sehr große Datenmengen geht.

## Über welche Datenmengen sprechen wir hier?

**Thorsten Gerke:** Mittlerweile müssen die Ingenieure Daten im Petabyte-Bereich analysieren, um ihre Algorithmen zu entwickeln und entsprechend zu verifizieren. Die Datenanalytik geht dann direkt in die Regelalgorithmen über, und die Entwicklung dieser Algorithmen lässt sich aufgrund ihrer Komplexität nur noch modellbasiert durchführen. Natürlich geht es dabei nicht nur um die Daten eines Einzelsensors, sondern von mehreren Sensoren und Sensortypen, die fusioniert werden müssen.

## Wo läuft denn die Sensorfusion der Zukunft ab?

**Thorsten Gerke:** Um das automatisierte Fahren erfolgreich zu machen, müssen wir ein einwandfreies digitales Abbild der Umgebung schaffen. Das Fahrzeug muss wirklich verstehen, was drum-



**Thorsten Gerke (rechts; im Gespräch mit AUTOMOBIL-ELEKTRONIK-Chefredakteur Alfred Vollmer): „Die Automated Driving System Toolbox vereint den Bereich Perception mit der Planung des Fahrwegs und der Regelung in Form von Lenken, Bremsen und Beschleunigen.“**

herum passiert, um dann die entsprechenden Fahrmanöver planen zu können. Der zentrale Punkt dabei ist die Sensordatenfusion von Kamera-, Radar- und Lidar-Daten sowie das anschließende Fällen von Entscheidungen. Dieser Vorgang wird sehr wahrscheinlich zukünftig in einer zentralen Einheit erfolgen; diverse Kunden führen Daten von detektierten Objekten aus verschiedenen Sensoren zusammen. Die Sensoren werden in Zukunft wohl die Rohdaten direkt an das zentrale Steuergerät zur Sensordatenfusion liefern. Wir haben vor einigen Tagen die Automated Driving System Toolbox, kurz ADST genannt, in der Produktrelease R2017a auf den Markt gebracht, die Autofirmen unter anderem bei der Entwicklung der Algorithmen für die Sensordatenfusion als auch der Gesamtentwicklung automatisierter Fahrfunktionen in Verbindung mit anderen Mathworks-Produkten unterstützt.

#### **Ein weiteres Problem – auch in punkto Zeit – ist das Testen...**

**Thorsten Gerke:** Test und Verifikation sind neben der Entwicklung und dem Design der komplexen Algorithmen sicherlich mit das

technologische Fundament für das automatisierte Fahren. Um im Markt erfolgreich zu sein, müssen wir erreichen, dass die Endkunden Vertrauen in die Fahrfunktionen des Fahrzeugs haben, denn nur dann werden sie es kaufen. Die Branche muss noch einiges tun, um das Vertrauen in automatisiertes Fahren aufzubauen – und wie kann dies geschehen? Nur mit konsequentem umfassendem Testen! Sowohl mit realen Prototypen als auch mit virtuellen Prototypen mittels Simulation.

Eigentlich müssten die OEMs zur Absicherung eines vollständig automatisierten Systems mehrere hundert Millionen Testkilometer fahren, aber das ist zeitlich nicht machbar, und es wäre auch nicht wirtschaftlich. Darüber hinaus ist es im realen Test auch schwierig, dedizierte Worst-Case-Szenarien gezielt abzubilden beziehungsweise zu reproduzieren. Daher wird neben den einzelnen Fahrzeugtests die Modellierung und Simulation in einer virtuellen Umgebung eine ganz zentrale Rolle spielen. Zunächst gilt es, die Perception, also die Umfeldmodellerkennung zu verifizieren und abzusichern – und zwar noch lange bevor es zu einer direkten Verbindung zur Regelung kommt. Wir unter-

stützen in ADST die Entwickler bereits mit einem Szenariengenerator, der es erlaubt, Sensor-Fusion-Algorithmen mit Hilfe von synthetischen als simulierten Daten zu testen und abzusichern. Langfristig wird es notwendig sein, dies in einer vollständigen 3D-modellierten Simulation durchzuführen. Synthetisch gewonnene Daten mittels Simulation werden zukünftig an Bedeutung zunehmen, da man nicht jedes einzelne Fahrmanöver auf der Straße testen kann beziehungsweise für das Training neuer Detektionsalgorithmen initial gar nicht immer ausreichend Daten aus realen Fahrzeugtests zur Verfügung stehen.

Zum Beispiel hat Scania bereits zusammen mit Tier-1s komplett modellbasiert ein automatisiertes Bremssystem für ihre Trucks entwickelt, bei dem wir mit involviert waren. Wenn bei einem 40-Tonner irgendetwas nicht richtig funktioniert, kann das ganz böse enden. Da durfte wirklich nichts schiefgehen. Auch bei diesem Projekt hat sich das schrittweise Vorgehen in Kombination mit modellbasierter Entwicklung bewährt.

### Wie lässt sich der Übergang vom reinen Algorithmus zum modellbasierten Testen realisieren?

**Thorsten Gerke:** Wenn ein Algorithmus in Matlab und Simulink umgesetzt ist, erhält man automatisch eine ausführbare Spezifikation und damit einen virtuellen Prototypen, mit dem die Ingenieure direkt Simulationen mit unterschiedlichen Verkehrssimulationen durchführen können. Wenn sich dann zeigt, dass die Requirements abgebildet sind, folgt die zweite Stufe: die Integration in ein Steuergerät, also die Überleitung vom funktionalen Modell in ein Implementierungsmodell. Auch das geht



*„Wer in Zukunft seine Wettbewerbsfähigkeit steigern will, der muss auf modellbasierte Entwicklung setzen.“*

Thorsten Gerke

mit MATLAB und Simulink, und mit Hilfe der automatischen Code-Generierung entsteht dann der Seriencode für die ECU. Damit es hier zu einem wirklich nahtlosen Übergang zwischen modellbasierter Entwicklung und Implementierung im Steuergerät kommt, haben wir seit vielen Jahren die Entwicklung unserer Code-Generierung signifikant ausgebaut bis hin zur optimierten Seriencodegenerierung mittels Embedded Coder. Dies wird auch in der neuen Automated Driving System Toolbox unterstützt.

### Welche besonderen Eigenschaften hat die brandneue Automated Driving-System-Toolbox?

**Thorsten Gerke:** Die Automated Driving System Toolbox knüpft genau an dem Punkt an, den wir gerade besprochen haben, denn sie kombiniert Entwicklung und Absicherung der Perception –

also dem Umfeldmodell inklusive der Erkennung von Objekten – mit der anschließenden Planung des Fahrwegs und der Regelung in Form von Lenken, Bremsen und Beschleunigen. Sie bildet somit das wichtige Bindeglied zwischen der Funktionsentwicklung der Sensorik, beispielsweise in Form eines Computer-Vision-Algorithmus für ein Kamerasystem, der mit Hilfe der Computer Vision System Toolbox in Matlab realisiert werden kann, und den in Simulink modellierten anschließenden autonomen Regelungsfunktionen, die direkt in die Fahrzeugdynamik und somit aktiv in den Straßenverkehr eingreifen. Mit dem Release 2017a ist die Automated Driving System Toolbox jetzt verfügbar.

Bei der Perception unterstützt die Toolbox die Verarbeitung und Fusionierung von Radar-, Kamera- und Lidar-Daten auf Objektlistenebene. Auch der Test und die Absicherung erfolgen mit der Toolbox – meist mit aufgezeichneten Video- und Radar-daten, wobei die Toolbox dies entsprechend visualisiert. Für das modellbasierte Testen gibt es einen eingebauten Szenario-Generator, der virtuelle Verkehrsszenarien erzeugt, um beispielsweise einen Sensor-Fusionsalgorithmus zu testen. Neben dem Framework selbst sind auch sehr viele Beispiele ein fester Bestandteil der Toolbox, damit die Entwickler nicht bei Null anfangen müssen sondern quasi eine Starthilfe bekommen. Eigene (Fusions-) Algorithmen in C beziehungsweise C++ und Algorithmen der Toolbox lassen sich dabei auch mischen. Unsere Leitkunden haben uns dafür ein gutes Feedback gegeben.

Durch Verwendung von ADST in Verbindung mit anderen Mathworks-Produkten ist es möglich, auf Deep-Learning basierende Modelle vollständig selbst zu trainieren oder Transfer-Learning mit vortrainierten Modellen zu verwenden, die bereits Tausende von Objekten erkennen. Anwender können das Training mit GPUs auf ihrem Multi-Core Computer oder durch Skalierung in der Cloud beschleunigen.

Die auf Matlab basierende Automated Driving System Toolbox ist Teil einer sehr langfristig orientierten Entwicklung. Zum einen bietet es eine Auswahl an Algorithmen für die Entwicklung von Funktionen für das automatisierte Fahren, die von den Algorithmenentwicklern in ihr existierendes Toolset integriert werden können. Darüber hinaus gibt die Toolbox den Entwicklern fertige Beispiele an die Hand, die die Integration und Analyse von Daten durch Visualisierung sehr einfach für Teams im Bereich des automatisierten Fahrens machen. Da viele Hochschulabsolventen und Entwickler bereits intensiv mit Matlab vertraut sind, lassen sich mit der Automated Driving System Toolbox schnell konkrete Ergebnisse erzielen.

### Können es sich die OEMs und Tier-1s überhaupt noch leisten, nicht modellbasiert zu entwickeln?

**Thorsten Gerke:** Der globale Wettbewerbsdruck nimmt nicht nur in der Automobilindustrie massiv zu, und weil das Qualitätsbewusstsein der Endkunden zunimmt, während gleichzeitig die Security-Anforderungen steigen und sich die Time-to-Market



verkürzt, ist die modellbasierte Entwicklung ein zentraler Punkt geworden. Wer heute noch nicht modellbasiert entwickelt, der wird es zukünftig machen. Über die Jahre hinweg hat sich modellbasierte Entwicklung in vielen Anwendungsbereichen wie Powertrain, Body Electronics oder auch Vehicle Dynamics als wichtiger Bestandteil der Entwicklungskette etabliert und macht den Entwicklungsprozess und die Produkte kompetitiv. Nun sehen wir diese Entwicklung auch im Bereich des automatisierten Fahrens. Wer in Zukunft seine Wettbewerbsfähigkeit steigern will, der muss auf modellbasierte Entwicklung setzen.

### Welche Veränderungen beobachten Sie bei den Anwendungen der modellbasierten Entwicklung?

**Thorsten Gerke:** In der Vergangenheit kam die modellbasierte Entwicklung vor allem im Komponentenbereich zum Einsatz, quasi gemappt auf bestimmte Funktionen im Automobil. Mittlerweile spielen auch die Anwendungen im Systembereich eine immer wichtigere Rolle, weil die OEMs so weit wie möglich auf reale Fahrzeugprototypen verzichten möchten. Für abschließende Tests sind natürlich reale Prototypen erforderlich, aber in den frühen Phasen des Designs sind die realen Prototypen in der Regel noch nicht verfügbar, so dass es allein schon aus diesem Grund keine Alternativen zu virtuellen Prototypen gibt. Da die Entwicklungen in der Regel an vielen Standorten auf der Erde parallel erfolgen, müsste an all diesen Standorten jeweils ein realer Prototyp zur Verfügung stehen, so dass auch hier der virtuelle Prototyp eindeutig bessere Dienste leistet. Der Trend geht zur kompletten Virtualisierung der Fahrzeugentwicklung – sowohl auf der mechanischen als auch auf der elektronischen Seite.

Wichtig ist, dass alle Beteiligten, also alle Zulieferer und der OEM jeweils die Modelle mit beisteuern und eng zusammenarbeiten. Simulink dient dann als Integrationsplattform, um das komplette Simulationssystem aufzubauen.

### Was kommt nach dem virtuellen Test?

**Thorsten Gerke:** Ein ganz zentraler Punkt der modellbasierten Entwicklung ist die automatische Erzeugung des Codes, der auf dem Steuergerät laufen wird. Mit automatisch generiertem Code ist auch das Arbeiten im Kontext von ISO 26262 viel einfacher, zumal der Workflow damit traceable wird: bis ASIL-D. Das von ISO geforderte Back-to-Back-Testing, also die Verifizierung des generierten Codes gegen das Modell, ist bei unserem Embedded Coder genannten Codegenerator ein wichtiges Element. Eine manuelle Codierung ohne Embedded Coder ist sehr viel fehlerträchtiger.

### Wie bekommen wir die Analyse großer Datenmengen in den Griff?

**Thorsten Gerke:** Um aus der Analyse großer Datenmengen Modelle zu erstellen und Vorhersagen abzuleiten, spielt Deep-Learning,



*„Der Trend geht zur kompletten Virtualisierung der Fahrzeugentwicklung.“*

Thorsten Gerke

also maschinelles Lernen eine ganz große Rolle – vor allem dann, wenn neuronale Netze bei der Entwicklung von Algorithmen und beim Klassifizieren zum Einsatz kommen.

### Welchen Beitrag kann modellbasierte Entwicklung im Antriebsstrang leisten?

**Thorsten Gerke:** Ohne fortschrittliche Algorithmen sind die Powertrain-Entwicklungen für morgen und übermorgen nicht mehr umsetzbar. Das gilt für den Verbrennungsmotor genauso wie für E-Antriebe. Um die Anforderungen zu erfüllen, müssen teilweise im laufenden Betrieb Modelle im Steuergerät mitlaufen, die das Verhalten des Systems mit Hilfe eines Algorithmus bereits voraussberechnen, um so eine deutlich verbesserte Regelung zu erzielen. Wir haben mit Powertrain Blockset ein spezielles Produkt für diese Anwendung im Programm, das Closed-Loop-Simulationen macht, um die Performance von Algorithmen zu verbessern. Powertrain Blockset kann sowohl für Model-in-the-Loop und

Software-in-the-Loop als auch für Hardware-in-the-Loop eingesetzt werden. Die Performanz der Modelle in Powertrain Blockset wurde dafür optimiert.

Langfristig werden wir jedoch um die Elektrifizierung des Antriebsstrangs nicht umhinkommen, um die stringenten Emissionsforderungen zu adressieren. Auch in diesem Fall liefern wir die entsprechende Unterstützung, zumal wir bei der Code-Generierung nicht nur C-Code sondern auch VHDL-Code für FPGAs zur Regelung der elektrischen Maschine erzeugen können.

### Welche Bedeutung hat das Thema Security?

**Thorsten Gerke:** Security ist mit Sicherheit ein ganz wichtiges Thema geworden, besonders dann, wenn es um die Anbindung des Fahrzeugs an die Umgebung und die Cloud geht. Mit der Öffnung nach außen entsteht bekanntlich eine größere Angriffsfläche für Hacker, und mit jeder Zeile Code steigt die Gefahr, dass sich theoretisch irgendwo ein Programmierfehler einschleicht, der zu einer Schwachstelle des Fahrzeugs werden könnte. Mit dem neuen SAE-Standard SAE J3061, der den kompletten Security-Lifecycle abbildet, macht die Branche den Schritt in die richtige Richtung. Auch wir unterstützen unsere Kunden, indem wir zum Beispiel mit unserer statischen Codeanalyse-Lösung Polyspace auch Security-Checker anbieten, die gezielt nach Schwachstellen im Code suchen. Im Bereich Security wird sich in Zukunft noch einiges tun. ■

### Interviewer

Alfred Vollmer  
Chefredakteur AUTOMOBIL-ELEKTRONIK

all-electronics.de   
infoDIREKT

300ael0417