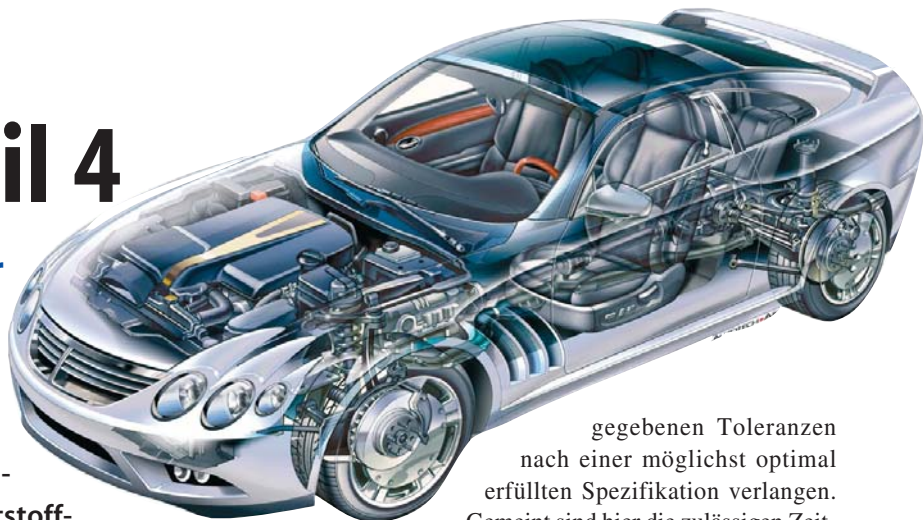


Das Virtuelle Fahrzeug – Teil 4

Entwicklung und Optimierung der Elektronik von Einspritzsystemen

An heutige Antriebssysteme werden seitens der Fahrzeughersteller und deren Zulieferer ständig steigende Anforderungen nach mehr Effizienz, geringerem Kraftstoffverbrauch, hohem Fahrkomfort und der sich zukünftig verschärfenden Schadstoffemission gestellt. Dieser Artikel beschreibt eine Entwicklungsmethodik, die mit Hilfe so genannter Robust-Design-Techniken eine qualitative Optimierung und Absicherung der Systeme sicherstellt. Die Anwendung dieser Methode wird beispielhaft anhand eines elektrischen Einspritzsystems demonstriert.



gegebenen Toleranzen nach einer möglichst optimal erfüllten Spezifikation verlangen. Gemeint sind hier die zulässigen Zeitintervalle für die Einschaltzyklen der Einspritzung und auch die auftretenden Flanken. Sie garantieren, dass der Injektor auch optimiert und innerhalb des geforderten Zeitintervalls exakt einspritzt. Des Weiteren stellt sich auch die Frage der frühzeitig verifizierten Diagnose unter Berücksichtigung der Bauteiltoleranz. Diese Frage ist für den Entwickler nicht einfach zu beantworten. In der frühen Entwicklungsphase liegt noch kein Prototyp vor, der für eine Verifikation herangezogen werden könnte. Auch der Hardware-Prototyp lässt keine gezielte Untersuchung des Worst-Case-Verhaltens bei Komponententoleranzen zu. Jede Hardware-Implementierung des Steuergerätes stellt lediglich eine spezielle Konfiguration hinsichtlich ihrer Toleranzen dar. Entwickler bedienen sich bei der unumgänglichen Untersuchung an simulationsbasierten Methodiken. Die Anwendung so genannter Robust-Design-Techniken ist nicht unbedingt an eine Simulation gebunden. Bei der Anwendung sind spezielle Schritte dieser Methodik notwendig, die sich praktisch mit Hardware-Prototypen gar nicht durchführen lassen oder nur unter sehr großem Zeit- und Kostenaufwand appliziert werden können.

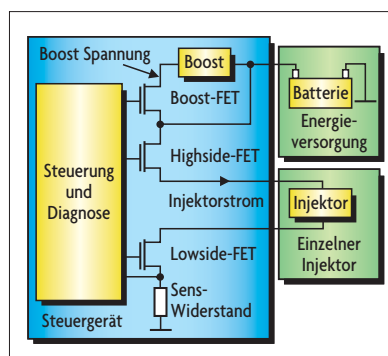
Von Thorsten Gerke und Frank Lehmann

Aktuelle Antriebssysteme enthalten eine komplexe Elektronik, die sowohl Hardware als auch Software-Bestandteile beinhaltet. Diese Systeme müssen teilweise widersprüchliche Ansprüche optimal erfüllen. Typische Anforderungen an heutige Implementierungen sind hohe Leistung bei möglichst minimalem Verbrauch, hoher Fahrkomfort, geringe Schadstoffemission, lange Lebensdauer und Ausfallsicherheit sowie einfache Diagnostizierbarkeit.

Ein wichtiger Bestandteil des gesamten Antriebssystems ist das Einspritzsystem und dessen elektrische Steuerung. Eine Applizierung der obi-

gen Kriterien auf dieses System hat einen signifikanten Einfluss auf das Gesamtergebnis in Bezug auf die Qualität des Antriebssystems. Bild 1 stellt schematisch den elektrischen Teil eines Einspritzsystems dar, der z.B. bei Direkteinspritzsystemen zum Einsatz kommt. Das System besteht aus zwei Teilen. Zum einen aus den physikalischen Endstufen, Treibern und der Spannungsversorgung. Diese Teile liefern die Stromzufuhr für den Injektor. Zum anderen aus der Applikations-Software, welche die Steuerung der Einspritzung übernimmt und die Diagnose durchführt. Hierbei wird über den Highside-Gate-Treiber ein Power-MOSFET, der das Injektorsystem mit Strom versorgt, mit einem PWM-Signal getaktet. Um zu Beginn der Einspritzphase genügend Strom in kurzer Zeit bereitzustellen, ist dem System noch ein Boost zugeschaltet. Dieser liefert eine 60-V-Versorgungsspannung und wird über einen weiteren Power-MOSFET kurzzeitig zugeschaltet.

Die besondere Schwierigkeit bei dieser Entwicklung sind die gestellten Anforderungen der OEMs, die bei vor-



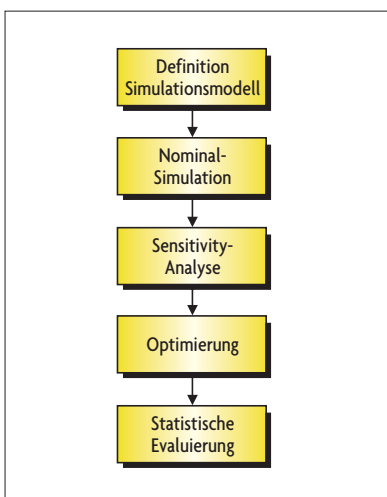
! Bild 1. Prinzip der Einspritzung.

Die Robust-Design-Entwicklungsmethodik

Die Robust-Design-Methodik stellt sicher, dass die Implementierung eines Systems unter allen möglichen Variationen innerhalb der Spezifikation einwandfrei funktioniert. Toleranzen von Komponenten durch Fertigung oder

auch Temperatureinflüsse werden in diese Betrachtungen mit einbezogen. Eine der bekanntesten Methoden aus diesem Bereich ist „Design for Six-Sigma“.

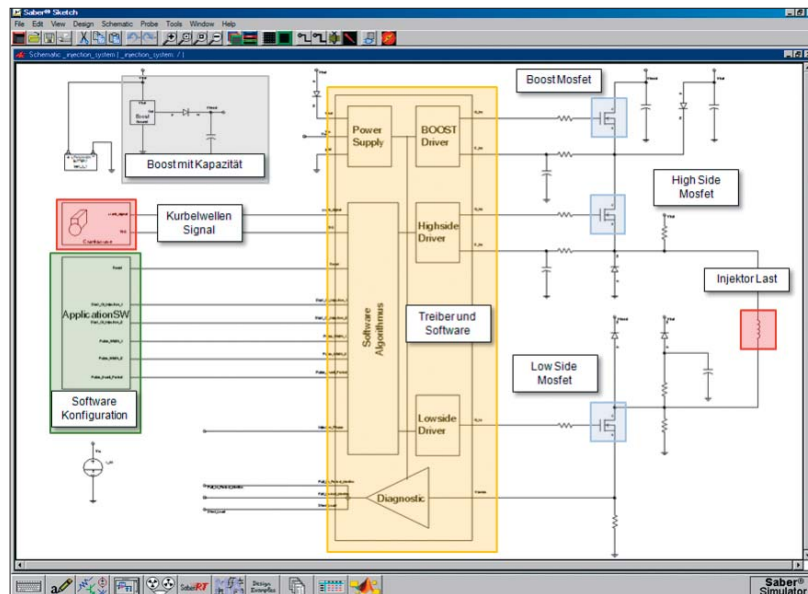
Die Robust-Design-Methodik beinhaltet fünf aufeinander aufbauende Stufen (Bild 2). Im ersten Schritt entsteht aus der Definition des Systems und des zugehörigen virtuellen Prototyps das Simulationsmodell. An dieser Stelle wird festgelegt, auf welche Aspekte des Systems sich die Untersuchungen erstrecken. Definiert wird, ob temperaturabhängige Effekte berücksichtigt oder vernachlässigt werden. Ebenso wird festgelegt, wie die Modellierung des analogen Systemverhaltens aussieht und in welchem Frequenzbereich gearbeitet wird. Die Komponentenparameter, welche einen signifikanten Einfluss auf das Systemverhalten haben, ergeben die Parame-



! Bild 2. Die Robust-Design-Methodik enthält fünf Stufen.

ter, die für die Simulation mit Toleranzen zu beaufschlagt sind. Diese Anforderungen müssen genau geprüft werden. Sie nehmen erheblichen Einfluss auf die Genauigkeit der darauffolgenden Systemsimulation. An dieser Stelle werden meist auf Grund unzureichender Physikkenntnisse Fehler gemacht. Resultierend daraus können unnötig umfangreiche Modelle simuliert werden, die unnötige Effekte enthalten. Nach gewissenhafter Aufstellung aller notwendigen Parameter wird das Systemmodell aufgebaut.

Das vorliegende Modell wird im nächsten Schritt einer Nominalanalyse



! Bild 3. Simulationsmodell eines Einspritzsystems.

unterzogen. Sie simuliert das System ohne Berücksichtigung von Bauteiltoleranzen und zeigt auf, welche Probleme sich eventuell schon im Nominalbetrieb offenbaren. Direkt zu Beginn Toleranzen in Betracht zu ziehen, wäre wenig sinnvoll. Abhängig vom Fokus der Untersuchung können hier Analysen im Zeit- oder im Frequenzbereich durchgeführt werden. Um die Leistung des Systems zu optimieren, ist es notwendig, zu wissen, an welchen „Stellschrauben“ im Design gedreht werden muss. Hierzu wird im nächsten Schritt eine Sensitivity-Analyse durchgeführt. Sie ermittelt den Einfluss eines jeden Designparameters – z.B. des Widerstandes – auf die Designleistung. Für den Entwickler ist so erkennbar, welche Parameter zur Optimierung geeignet sind und welche hinsichtlich ihrer Toleranzen sensitiv behandelt werden müssen. Der nächste Schritt ist die Optimierung des Systemverhaltens durch gezielte Variationen der Komponentenparameter. Anschließend ist die Simulation der Bauteiltoleranzen notwendig. Liegt das Systemverhalten noch außerhalb der Spezifikation, müssen die Schritte erneut evaluiert werden.

■ Anwendung der Robust-Design-Methodik auf ein Einspritzsystem

Gemäß der Robust-Design-Methodik werden zunächst die Grundanforde-

rungen an das Simulationsmodell gestellt. Die Endstufen sind durch Verhaltensmodelle der Power-MOSFETs abgebildet. Der Boost wird durch ein Verhaltensmodell berücksichtigt. Die Ausgangsleistung ist hier eine Funktion der Batteriespannung. Digitale Modelle abstrahieren die Logik der Treiber. Die Software wird durch ihren Algorithmus abgebildet. Eine Berücksichtigung der gesamten Software wäre nicht sinnvoll. Das Motorsteuergerät verfügt über viele Funktionalitäten, die für die Betrachtung des Einspritzsystems nicht von Interesse sind. Die Injektorlast wird als verlustbehaftete Induktivität modelliert. Bild 3 zeigt das Einspritzsystemmodell im Systemsimulator Saber. Nach der Modelldefinition und dem Aufbau erfolgt die eigentliche Simulation des nominellen Systemverhaltens. So kann der Stromverlauf während der Einspritzphasen bewertet werden. Bild 4 zeigt den Stromverlauf durch den Injektor. Der Verlauf der Pegel lässt erkennen, dass es sich um eine dreistufige Stromregelung handelt. Der Boost ist dabei nur während der ersten Phase durchgeschaltet, um zügig den gewünschten Strompegel zu erreichen.

Im Nominalbetrieb zeigt das System das gewünschte Verhalten und keine abnormalen Auffälligkeiten. Zur weiteren Betrachtung werden

die Toleranzen der verbauten Komponenten hinzugezogen, die aus den Datenblättern der Hersteller entnommen werden können. Abhängig von den Komponenten wird sowohl ein normal als auch gleich verteiltes statistisches Verhalten berücksichtigt. Ein Bewertungskriterium für die elektronische Einspritzung ist die Zeit, die beim ersten elektrischen Einspritzimpuls benötigt wird, um 15 A für den Injektorstrom zu erreichen. Laut Vorgabe darf der Zeitraum nicht größer als 370 µs sein. Bei einer statistischen Un-

tersuchung des Einspritzsystems unter Berücksichtigung der Bauteiltoleranzen ist dieser Effekt mit in Betracht zu ziehen.

Bild 5 zeigt das Ergebnis einer statistischen Untersuchung für die Zeit, die zur Erreichung des Stromschwellwertes im Injektor benötigt wird. Zur Aufarbeitung dieser Daten wurde eine Monte-Carlo-Analyse durchgeführt, bei der simultan die Parameterwerte aller mit Toleranzen beaufschlagten Simulationsmodelle geändert werden. Dabei werden die Werte gemäß der ausgewählten Verteilungsfunktionen „gewürfelt“. Um eine ausreichende Abdeckung zu erreichen, sind 2000 Durchläufe appliziert worden. In jedem Durchlauf wird eine Zeitbereichssimulation durchgeführt. Das Ergebnis sind Kurvenscharen mit jeweils 2000 Einzelkurven für jedes Signal im System. Die aufkommende Datenflut wird effizient und automatisch im Post-Processing-Tool von Saber ausgewertet. Es fällt auf, dass viele Werte oberhalb des maximal zulässigen Werts liegen. Die vorliegende Implementierung genügt in diesem Punkt nicht den Anforderungen der Spezifikation. Damit stellt sich die Frage, welche Änderungen in die Implementierung eingepflegt werden müssen, damit das System in den zulässigen Bereich gelangt. Hierzu bedient sich der Entwickler des dritten Schritts der Robust-Design-Methodik, der Sensitivity-Analyse. Die Sensitivität s einer Systemgröße (in diesem Fall z.B. die Zeit zur Erreichung des Stromschwellwertes) berechnet sich dabei wie folgt:

$$s = \frac{\partial m}{\partial p} \times \frac{p}{m}$$

Wenn z.B. eine Sensitivität von 1 vorliegt, verursacht eine fünfprozentige Änderung des betreffenden Parameters p um seinen Nominalwert herum eine fünfprozentige Änderung in der betrachteten Messgröße m . Die Sensitivitäts-Analyse erlaubt die Identifizierung der wichtigen Stell-schrauben im System. Außerdem liefert sie ein Maß für die Adjustierung

der betreffenden Design-Parameter. Der gesamte Prozess für die Definition der Sensitivitäts-Analyse ist durchgehend mittels intuitiver GUIs grafisch unterstützt. Für das Einspritzsystem werden unter anderem die MOSFETs der Endstufen, die Kondensatoren als auch die Kapazität am Boost mit in Betracht gezogen. Die Tabelle liefert eine Übersicht der berechneten Sensitivitäten. Es ist erkennbar, dass der serielle Widerstand ESR der Boost-Kapazität die größten Auswirkungen hat. Die Bahnwiderstände der Transistoren der Endstufen weisen eine hohe Sensitivität auf. Da sie am technischen Limit angekommen sind, eignen sie sich nicht als weitere Optimierungsgröße. Deutlich ist auch der Einfluss der Boost-Kapazität auf die Zeit zur Erreichung der Stromschwelle.

Es bietet sich an, eine nähere Untersuchung des Boost-Kondensators in Bezug auf ESR und Kapazität durchzuführen. Um deren Einfluss näher zu

Instanz	Parameter	Nominal	Sensitivität
Boost-Kapazität	ESR	0,5 Ω	0,98
Boost-MOSFET	R _{Dson}	12,7 mΩ	0,86
High-Side-MOSFET	R _{Dson}	12,7 mΩ	0,86
Low-Side-MOSFET	R _{Dson}	12,7 mΩ	0,86
Boost-Kapazität	C	380 µF	0,71

Berechnete Sensitivitäten verschiedener Systemgrößen

quantifizieren, werden im dritten Schritt beide Parameter mit Hilfe von Variationsschleifen untersucht. Bild 6 zeigt die Kurvenscharen für den Stromverlauf. Deutlich erkennbar ist hier der Einbruch der Spannung am Boost. Durch eine Verschachtelung der Variationsschleifen für ESR und Kapazität lassen sich hieraus Bereiche extrahieren. In diesen müssen die Werte eines Kondensators für den Boost liegen, damit die gewünschte Leistung erreicht wird. Für eine abschließende Implementierungsvalidierung sind die optimierten Boost-Kondensator-Werte erneut mit dem Gesamtsystem unter Berücksichtigung der Toleranzen zu simulieren. Hier bietet sich an, die gesamte Berechnung auf einem CPU-Cluster durchzuführen und die Durchläufe zu parallelisieren. Saber bietet hierzu die so genannte Distributed-Iterative-Analysis (DIA) an. Sie er-

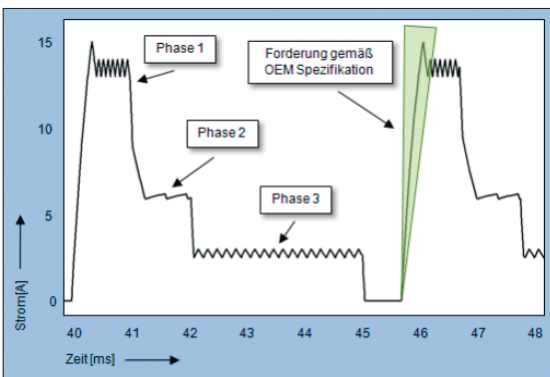


Bild 4. Injektorstrom im Nominalfall.

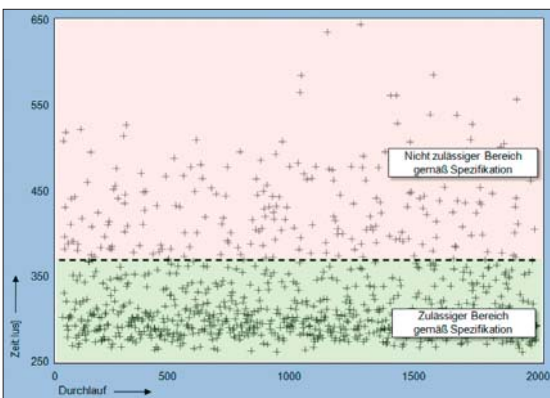


Bild 5. Monte-Carlo-Simulation zur Untersuchung der Parameterstreuung.

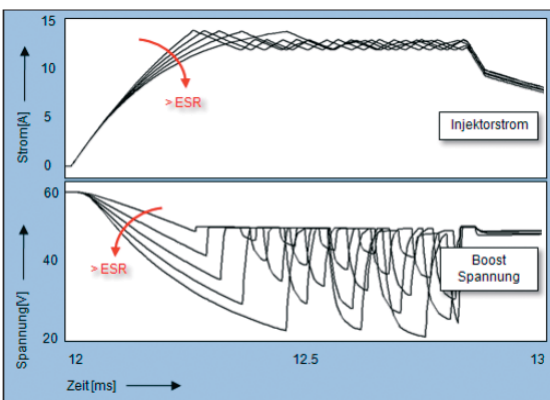


Bild 6. Injektor-Strom und Boost-Spannung bei Variation des ESR.

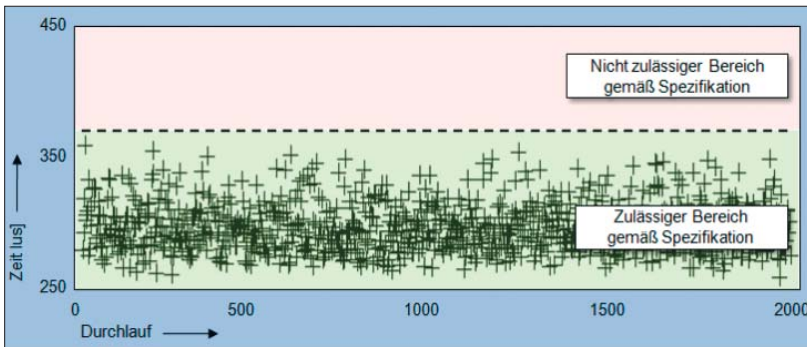


Bild 7. Optimiertes System mit finaler statistischer Evaluierung durch Einbeziehung der Toleranzen.

möglicht parallele Simulationen. In diesem Fall wurden die Simulationen auf einen Cluster mit 20 CPUs verteilt. Der zusätzliche Gewinn an Rechenleistung entspricht einem Faktor von ca. 19. **Bild 7** zeigt die Ergebnisse des optimierten Systems. Dargestellt ist hier die Zeit, die für das Erreichen der Stromschwelle benötigt wird. Nun befinden sich alle Punkte innerhalb des zulässigen Bereichs gemäß der Spezifikation.

■ Weiterführende Untersuchungen

In Abhängigkeit der Applikation kann es notwendig sein, weiterführende Untersuchungen durchzuführen, z.B. nach dem Systemverhalten im Fehlerfall. Hierzu bietet Saber die Umgebung Testify an. Sie erlaubt die Definition der Fehlerfälle und Bewertungskriterien sowie deren automatische Auswertung. Typische Fehlerfälle sind z.B. Kurzschluss zur Masse oder zur Versorgungsspannung. Der integrierte Algorithmus für die Diagnose bietet die komfortable Möglichkeit, unterschiedliche Fehlerfälle zu induzieren. Dies gibt Aufschluss über die Robustheit der Diagnose. Es bleibt festzuhalten, dass die Robust-Design-Methode in Verbindung mit dem Einsatz von virtuellen Prototypen eine effiziente und flexible Entwicklungsmethode darstellt. Sie ermöglicht es, die Qualität der Implementierung gezielt zu steuern. Die Identifikation der kritischen Stellschrauben im System als auch seine statistische Untersuchung sind Möglichkeiten, die im Rahmen einer reinen Hardware-Prototypen-Entwicklung nicht zur Verfügung stehen. Es bietet sich an, sowohl die Vorteile der simulationsbasierten Entwicklung als auch der Hardware-Proto-

typen-basierten Methode gemeinsam zu nutzen. *fr*

Literatur

- [1] Goodwin, B.; Patton, J.; Poole, J.: Modeling and Simulating a Variable Valve Timing System for Robust Design. SAE World Congress 2008.
- [2] Goodwin, B.: Designing Automotive Subsystems Using Virtual Manufacturing and Distributed Computing. SAE World Congress 2008.
- [3] Gerke, T.; Schanze, C.; Bollati, D.: Model based Robust Design Methodology for FlexRay Design Systems. FlexRay Product Day Congress 2007.
- [4] Gerke, T.: Das Virtuelle Fahrzeug, Teil 1: Entwurf von Vernetzungstopologien (In-Vehicle Networks). *Elektronik automotive* 2008, H. 5, S. 46ff.
- [5] Gerke, T.: Das Virtuelle Fahrzeug, Teil 2: Frühzeitige Validierung von Bordnetzen und Energiemanagement. *Elektronik automotive* 2008, H. 6, S. 30ff.
- [6] Gerke, T.; Lagerquist, S.: Das Virtuelle Fahrzeug, Teil 3: Robuste Entwicklung von Kabelbaumsystemen. *Elektronik automotive* 2008, H. 8, S. 58 bis 61.

Dipl.-Ing. Thorsten Gerke

studierte an der Universität Duisburg Maschinenbau und Mechatronik und begann dann als Support- und Applikations-Ingenieur bei Avanti-Systems. Von 2002 bis 2006 war er bei Synopsys in den Bereichen Applikation und Presales für Automotive-Lösungen zuständig. Seit 2006 leitet er das technische Marketing der Saber-Produktfamilie in Europa, qualifiziert Produkthanforderungen und arbeitet an der zukünftigen Strategie des Saber-Produktportfolios.

thorsten.gerke@synopsys.com

Dipl.-Ing. Frank Lehmann

studierte Elektrotechnik an der Hochschule für Technik, Wirtschaft und Kultur (HTWK) in Leipzig und begann dann als Entwicklungsingenieur bei Continental (damals SiemensVDO). Er betreute die Steuergeräte-Entwicklung im Bereich Powertrain und war zuständig für die Entwicklung von Simulationsmodellen. Seit 2007 ist er bei Synopsys als Applikationsingenieur tätig und zuständig für den technischen Kundensupport der Saber-Produktfamilie.

frank.lehmann@synopsys.com