

Wie Sigmatek mit seinem weiterentwickelten Engineeringtool »Lasal« dabei hilft, Applikationsprogrammierer zu entlasten und gleichzeitig die Qualität, Wiederverwendbarkeit sowie Wartungsfreundlichkeit von Code zu maximieren

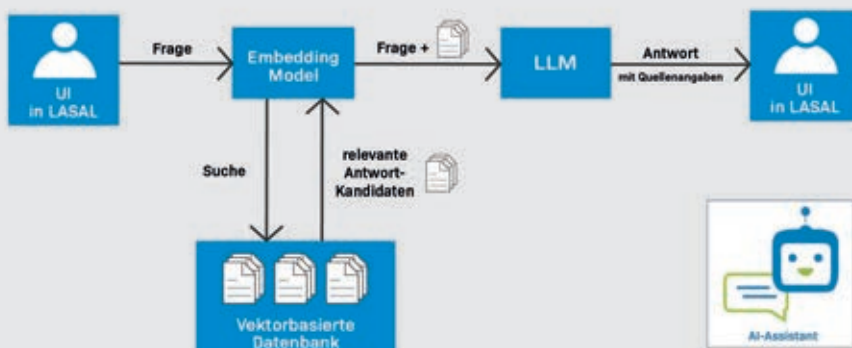
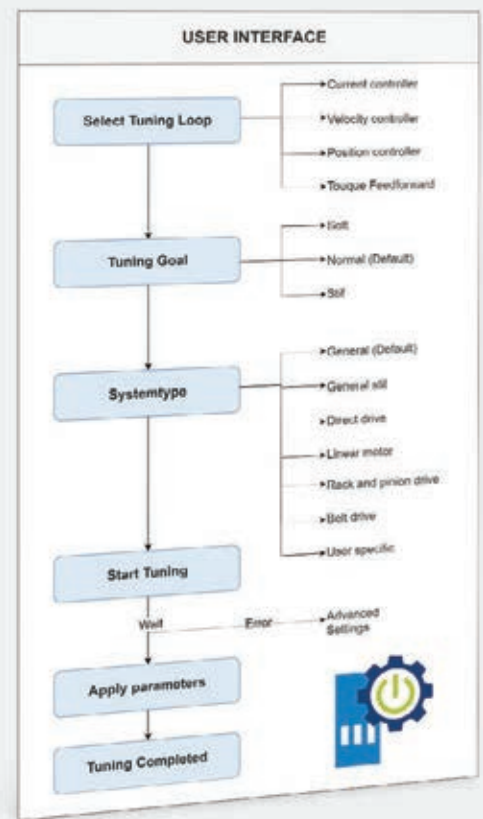
Lasst doch die Software machen!

Mit »Lasal« führt Sigmatek bereits seit über 25 Jahren ein umfassendes, objektorientiertes All-in-one-Engineeringtool zur Programmierung von Ablaufsteuerung, Safetyaufgaben, Visualisierung sowie Motion erfolgreich im Programm. Und der Salzburger Automatisierungstechnikhersteller entwickelt das hochperformante Softwarewerkzeug permanent weiter, um einerseits Engineeringprozesse im Maschinenbau zu beschleunigen und andererseits Entwicklern repetitive Arbeiten abzunehmen. Auf der kommenden »SPS«-Fachmesse in Nürnberg werden die jüngsten »Lasal«-Innovationen vorgestellt. Neu sind eine optionale Erweiterung zur statischen Codeanalyse, eine Autotuning-Funktion zur automatischen Regleroptimierung von Servoantrieben sowie ein KI-Assistent für mehr Effizienz im täglichen Workflow. Das Ziel: Einfacher coden in einer immer komplexer werdenden Automatisierungswelt. Von Ingrid Traintinger

Der Arbeitsalltag von Softwareentwicklern im Maschinenbau wird von steigender Komplexität, Skalierbarkeit, wachsender Dynamik und einer Unmenge an Daten bestimmt. „Wir wollen unseren Kunden eine durchgängige, komfortable Entwicklungsumgebung zur Verfügung stellen, die mit der Aufgabe wächst. Entwickler sollen von Routinearbeiten entlastet und in der Programmierung unterstützt werden. So können sie sich auf kundenspezifische Funktionalitäten konzentrieren, die dem Unternehmen Wettbewerbsvorteile bieten“, erklärt Alexander Melkus, Geschäftsführer von Sigmatek, den Ansatz von »Lasal«. Im Laufe der Jahre sind viele Komfortfunktionen hinzugekommen wie beispielsweise »Versionsverwaltung«, »Matlab Connector«, »Hardware-Editor« oder »Scripting mit Python«. Seit 2009 umfasst Sigmateks Engineering-Suite

mit dem »Lasal SafetyDesigner« ein nahtlos integriertes Tool, mit dem Anwender sicherheitsrelevante Anwendungen komfortabel konfigurieren können. Auch HTML5-Visualisierung wurde ins Tool integriert, und der »Lasal Machine Manager« vereinfacht das Handling bei Mehr-CPU-Lösungen. Ein Herzstück sind die umfangreichen Libraries, die im Laufe der Zeit enorm gewachsen sind und einsatzbereite Funktionsbausteine und Packages für eine Vielfalt an Maschinenfunktionen bereitstellen. Neueren Datums sind die Anbindung von modellbasierter Entwicklung mittels Digitalem Zwilling. Fortschrittliche Features erweitern den Funktions-

vermeiden. Die Funktion »Statische Code-Analyse« (»SCA«), macht es nun möglich, Softwarecode automatisiert zu prüfen, ohne dafür den Code auszuführen. So lassen sich potenzielle Fehler und Verstöße gegen interne Codierungsstandards und Best-Practices (z.B. PLCOpen) frühzeitig erkennen. Die Einhaltung eines definierten Coding-Stils ermöglicht eine problemlose Zusammenarbeit in größeren Teams – ohne Auswirkungen auf Wartbarkeit und Lesbarkeit des Quellcodes. Das ist ein wichtiger Faktor bei langlebigen Maschinen und Systemen. Zudem lässt sich mit dem Einsatz der »SCA«-Funktion die Einhaltung branchenspezifischer Richtlinien



Oben: Bei der automatischen Regleroptimierung übernimmt die neue »Autotuning«-Funktion die mathematische Auswertung nach festgelegten Kriterien vollautomatisch.

Links: Schematisches Funktionsprinzip des neuen, in »Lasal Class« integrierten »AI-Assistant«.

umfang des etablierten All-in-one-Engineering-tools »Lasal«, bieten in puncto Usability noch mehr Komfort und beschleunigen die Entwicklung. Gleichzeitig können Programmierexperten tief in das System einsteigen.

Fehler frühzeitig ausmachen

Mit der steigenden Komplexität der Automatisierung und somit den Anforderungen an die Software wird die Qualitätssicherung von Code immer wichtiger. Nur so lassen sich Skalierbarkeit und Wartbarkeit über Jahre hinweg gewährleisten. Umso später Softwarefehler erkannt werden, umso teurer fällt deren Behebung in der Regel aus. Daher lohnt es sich, bereits in der Entwicklungsphase häufige Fehlerquellen auszumachen bzw. von vornherein zu

und Standards sicherstellen. In »Lasal Class« zur Steuerungsprogrammierung überprüft die optionale Erweiterung »SCA« den in Structured Text erstellten Steuerungscode bzw. dessen Codebasis auf mögliche Laufzeitfehler und undefiniertes Verhalten, wie nicht initialisierte Variablen, Retournierung eines Pointers auf eine lokale Variable, implizite Konvertierungen usw. Dazu analysiert eine Vielzahl an unterschiedlichen, gleichzeitig laufenden Prüfregele die verschiedensten Aspekte des Steuerungs-codes und identifiziert eventuelle Qualitäts-mängel oder nicht eingehaltene Richtlinien. Dabei bietet die Software dem Anwender sehr flexible Konfigurationsmöglichkeiten: Jede der über 50 vordefinierten Regeln für Variablen, Typen, Funktionen und Methoden kann einzeln parametrisiert werden. Das Tool kann auch

Metriken berechnen. Metriken sind Messgrößen, die verschiedene Aspekte der Codebasis bewerten. Zudem sind auch komplett konfigurierbare Namenskonventionen möglich. Die Regeln, Metriken oder Namenskonventionen können an- und ausgeschaltet werden – sowohl individuell als auch in Gruppen. »



„Entwickler sollen von Routinearbeiten entlastet und in der Programmierung unterstützt werden. So können sie sich auf kundenspezifische Funktionalitäten konzentrieren, die dem Unternehmen Wettbewerbsvorteile bieten.“

Alexander Melkus,
Geschäftsführer
von Sigmatek.

„KI-gestützte Tools treiben die Automatisierung voran – sie übernehmen Routineaufgaben, machen spezifisches Wissen unmittelbar verfügbar und erzeugen Code automatisch.“

Maximilian Leschanowsky, Software-Entwicklungsleiter bei Sigmatek



Fehler augenscheinlich machen

Durch die automatisierten Basisprüfungen können sich Reviewer auf Korrektheit der Logik und Architektur konzentrieren. Die Analyse in einem eigenen Ausgabefenster zeigt Fehlermeldungen klar an, die – ähnlich wie beispielsweise bei der Compiler-Ausgabe in »Lasal« – via Doppelklick an die entsprechende Stelle im Code führen. Konfigurierbare, farblich gekennzeichnete Fehler-Prioritätsstufen wie »Trace«, »Info«, »Warning« und »Error« bieten eine schnelle Übersichtlichkeit im Problemfall. Die Fehlerausgabe kann zudem zeilenweise per Code-Kommentar unterdrückt werden. Durch Black-, White- und Ignore-Listen bei den entsprechenden Regeln kann die Prüfung noch besser an individuelle Kundenwünsche angepasst werden. Auch eigene Tabellen mit Dateien und Ordnern, die von der Prüfung ausgenommen werden sollen, sind möglich. „Statische Code-Analyse ist eine Art Sicherheits-

Bei der statischen Codeanalyse in »Lasal« werden die Fehlermeldungen in einem eigenen Ausgabefenster angezeigt. Durch Doppelklick auf eine Meldung wird an die entsprechende Stelle im Code gesprungen. Konfigurierbare, farblich gekennzeichnete Prioritätsstufen – »Trace«, »Info«, »Warnings« und »Error« – schaffen Übersichtlichkeit.

netz: triviale, aber oft aufwendig zu findende Fehler lassen sich verhindern, das Team wird entlastet, Produktivität und Softwarequalität steigen und das nicht nur bei der Erstellung, sondern auch bei der Wartung“, bringt Maximilian Leschanowsky, Software-Entwicklungsleiter bei Sigmatek, die Vorteile auf den Punkt.

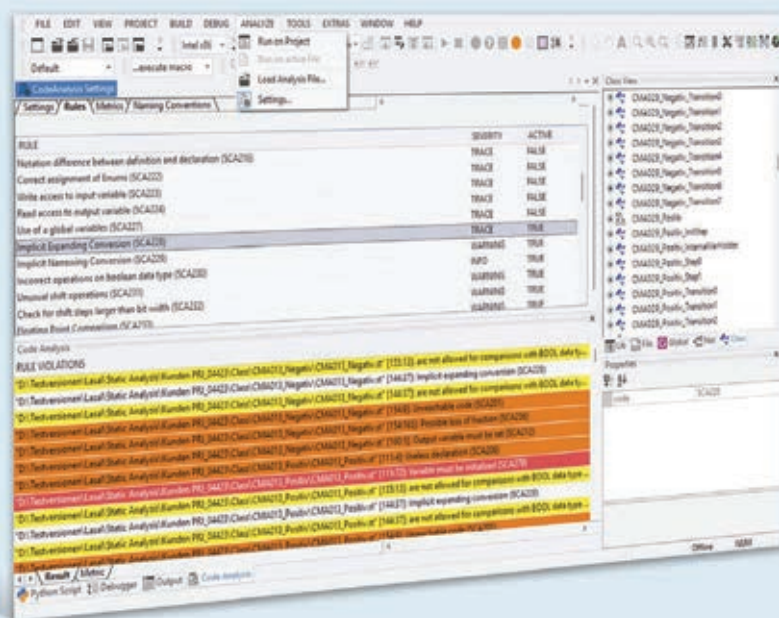
Autotuning: Automatische Regleroptimierung

In der Antriebstechnik beschleunigt und vereinfacht die neue »Autotuning«-Funktion in »Lasal« die Inbetriebnahme von Servoachsen enorm. Statt aufwändigem Parametrieren und mehrfachem Nachjustieren von Strom-, Geschwindigkeits- und Positionsregler, um Motoren und Achsen auf das mechanische System abzustimmen, erledigt die »Autotuning«-Funktion für die Sigmatek-Servoantriebe der »MDD 2000«-Serie die Regleroptimierung auf Knopfdruck. Die Parametrierung läuft vollständig automatisiert in der SPS und im Drive selbst ab und umfasst die gesamte Regelkette: die Parameter für den Strom-

regler, den Geschwindigkeitsregler, den Positionsregler sowie die Drehmomentvorsteuerung. Während der »MDD 2000«-Servoantrieb die notwendigen Messdaten aufzeichnet, übernimmt die SPS die mathematische Auswertung. Dazu zählen »Diskrete Fourier-Transformationen« (»DFT«) inklusive Berechnung von Bode-Diagrammen mittels »Sweep Frequency Response Analysis« (»SFR«), sowie die Einstellung, Optimierung und Validierung der Reglerparameter nach festgelegten Kriterien vollautomatisch.

Tuning mit vielen Facetten

Für Strom- und Drehzahlregler werden Startwerte aus den hinterlegten Motorparametern berechnet. Anschließend wird die Verstärkung automatisch so lange variiert, bis die Tuningziele erreicht sind. Beim Geschwindigkeitsregler lässt sich die Optimierung durch schrittweise Anpassungen der Verstärkung und Integrationszeit erzielen. Auch der Positionsregler kann automatisch eingestellt werden. Seine Verstärkung wird aus der Dynamik des Drehzahlreglers abgeleitet.



Nach demselben Prinzip wird die Drehmomentvorsteuerung bestimmt, sodass Schwingungen am Achssystem in Amplitude und Dauer minimiert werden. Ergänzend dazu lässt sich das Massenträgheitsmoment des Systems wahlweise über Frequenzganganalysen oder über das Aufschalten einer konstanten Beschleunigung bestimmen. Für den Anwender ist »Autotuning« komfortabel über ein Visualisierungsobjekt (VOV) zugänglich. Der Anwender wählt den zu tunenden Regelkreis, legt den Tuningbereich fest und startet den vollautomatischen Durchlauf. Das Tunen übernimmt der Algorithmus ohne weiteres Zutun des Benutzers. Um die Transparenz zu erhöhen, können sämtliche Messdaten als CSV-Dateien exportiert werden. Frequenzdaten lassen sich so direkt in gängigen Analyse-Tools wie »Excel«, »Matlab« oder im »Lasal Class Data Analyzer« darstellen.



Für den Anwender sind die neuen »Autotuning«-Funktionen für Sigmateks »MDD 2000«-Servoantriebe komfortabel über ein Visualisierungsobjekt zugänglich.

Synchrones Tuning mehrerer Achsen

In komplexen Maschinen müssen oftmals mehrere Antriebe parallel abgestimmt werden. Hier sorgt eine übergeordnete Klasse in »Lasal Class« dafür, dass alle beteiligten Achsen synchron starten, ihre Messungen durchführen und anschließend die neuen Parameter erhalten. Sollte während des Tuningprozesses ein Fehler (z.B.: Überschreitung des festgelegten Tuningbereichs) auftreten, wird das Tuning aller Achsen sofort abgebrochen, sodass ein sicherer Zustand gewährleistet ist. Mit »Autotuning« steht ein leistungsfähiges Tool zur Verfügung, mit dem Maschinen schneller startklar sind, die Regelqualität steigt und Bedienfehler vermieden werden.

KI-Assistent für mehr Effizienz

Ganz neu in der »Lasal«-Welt von Sigmatek ist der »AI-Assistent«. Mit Einsatz KI-basierter Tools wird die Entwicklungseffizienz gesteigert und das ist in puncto Workflow und Iterationszyklen entscheidend. Im ersten Schritt agiert der im SPS-Tool »Lasal Class« integrierte »AI-Assistent« als intelligente Wissensdatenbank, die mit Sigmatek-spezifischem Know-how gefüttert und trainiert wird: Hardware- und Software-Dokumentationen, Libraries mit Klassen, Funktionen, Packages sowie Best-Practices und FAQs. Der Anwender bekommt eine zentrale Anlaufstelle für optimierte Informationsbeschaffung und Zugriff auf Doku-

mentationen und Bibliotheken. Sigmatek nutzt »Agentic RAG« (»Retrieval-Augmented Generation«), um die richtigen Informationen in der Wissensdatenbank zu finden und diese mit Hilfe eines hochmodernen Sprachmodells (LLM/Large Language Model) in eine für den User nützliche Antwort umzuwandeln. Wichtig dabei ist die höchstmögliche Genauigkeit. Wenn der »AI-Assistent« zu keinem Ergebnis kommt, teilt er das mit und »halluziniert« nicht einfach. Bei seinen Antworten bietet er Links zur Informationsquelle für mehr Details und schlägt Bibliotheksfunktionen vor. Der Anwender kann rund um die Uhr Fragen stellen und erhält die Antwort in seiner jeweiligen Sprache.

Ausblick in die KI-geprägte Zukunft

Mit dem Einsatz Künstlicher Intelligenz verfolgt Sigmatek das Ziel, Anwender bei der Umsetzung von Basisfunktionen perfekt zu unterstützen. »Lasal Class«-Maschinenanwendungen bestehen in der Regel mindestens zu 50% aus Library-Klassen, die importiert und parametrisiert werden, um die gewünschte Funktion zu erfüllen. Dieser Prozess kann mit dem »AI-Assistent« beschleunigt werden, indem er die für die zu bewältigende Aufgabe am besten geeignete Funktionalität vorschlägt. Ein mögliches Zukunftsszenario könnte so aussehen: Stellt ein User dem »AI-Assistent« beispielsweise die Frage: »Wie kann ich ein

Alarmsystem umsetzen?«, sollte er als Antwort die vorgefertigten Klassen in der Library – beispielsweise »_AlarmXBuffer« – vorgeschlagen bekommen. Passen sie für den Entwickler, gibt dieser das Kommando zum Laden der Library und Importieren der entsprechenden Klassen – konkret »_AlarmXBuffer«, »_AlarmXPara«, und zum Visualisieren »_AlarmDisplayList«. Der »KI-Assistent« erklärt zudem bei Bedarf die Parametrierung im Detail. »Wir stehen am Anfang einer spannenden Reise. KI-gestützte Tools treiben die Automatisierung voran – sie übernehmen Routineaufgaben, machen spezifisches Wissen unmittelbar verfügbar und erzeugen Code automatisch. Das Ergebnis sind kürzere Entwicklungszeiten, höhere Qualität und eine kosteneffiziente, zuverlässige Automatisierung«, resümiert Maximilian Leschanowsky – nicht ohne zu betonen: »Entscheidend ist, dass vertrauliches Firmen-Know-how geschützt bleibt. Denn gerade dieses spezifische Wissen ist der eigentliche Wettbewerbsvorteil und darf nicht zur Allgewinnware werden.« (TR)

Zur Autorin: Ingrid Traintinger leitet die Marketing-Kommunikation bei Sigmatek in Lamprechtshausen.

»SPS«-Messe: **Halle 7, Stand 270**

INFOLINK: www.sigmatek-automation.com