

Semiautomatisch generierte Element- und Prozesssimulationsmodelle für PLS-Tests

Semi automatic generated element and process simulation models for PLC-tests

M.Sc. **Mike Barth**, Helmut-Schmidt-Universität, Hamburg,
Dipl.-Phys. **Peter Weber**, ABB AG Forschungszentrum, Ladenburg,
Prof. Dr.-Ing. **Alexander Fay**, Helmut-Schmidt-Universität, Hamburg,
Dr.-Ing. **Jürgen Greifeneder**, ABB AG Forschungszentrum, Ladenburg.

Kurzfassung

Im Zuge des stetig ansteigenden Anlagenkomplexitätslevels einerseits und des zunehmenden Qualitäts- und Zeitdrucks andererseits wird die frühzeitige Identifizierung von Fehlern im Steuerungscode zu einem maßgeblichen Wettbewerbsaspekt. Eine Lösung bietet die Verwendung von Anlagensimulationen in frühen Projektphasen, welche jedoch mit möglichst geringem Zeitaufwand erstellt werden können müssen. Der vorliegende Beitrag beschreibt daher einen Ansatz, bei dem die notwendigen Steuerungs- und Anlagendaten direkt in eine in Microsoft Excel implementierte Simulationsumgebung importiert werden, auf deren Basis unterschiedliche Tests durchgeführt werden können. Der Artikel fokussiert dabei insbesondere auf die Funktionsweise sowohl des Anlagensimulationsmodellgenerators als auch die der Simulationsumgebung.

Abstract

In the course of rising plant complexity levels on the one hand and increasing pressure regarding quality, and time on the other hand, the early identification of faults in the control code becomes a relevant competition aspect. A solution to this is given by using plant simulation in early project phases. Such a plant simulation however should be generated with as low expenditure in time as possible. Therefore this article describes an approach that imports all necessary control and plant data directly into the simulation environment. The latter is implemented in Microsoft Excel and is the basis for the execution of different tests. The focus of this article is on the functionality of the plant simulation model generator as well as on the simulation environment.

1. Einleitung

Bevor eine Anlage der verfahrens- und prozesstechnischen Industrie in Betrieb genommen werden kann, muss eine Vielzahl an Tests durchgeführt werden, welche den späteren reibungslosen Betrieb sicherstellen sollen. Hierunter fällt unter anderem der so genannte Factory Acceptance Test (FAT), im Zuge dessen die automatisierungstechnischen Komponenten unmittelbar vor ihrer Auslieferung durch den Systemlieferanten, unter Mitwirkung des Kunden, getestet werden. Die Komplexität moderner Anlagen bedingt, dass diese Tests zunehmend umfangreicher und dadurch zeitintensiver werden. Da eine längere Testdauer eine spätere Inbetriebnahme zur Folge haben kann, besteht der Wunsch, die Automatisierungsapplikationen bereits entwicklungsbegleitend zu überprüfen [1]. Diesem kann nachgekommen werden, indem die zu dieser Projektphase noch nicht verfügbaren Anlagenkomponenten (Sensoren, Aktoren, Behälter, Rohrleitungen etc.) durch virtuelle Modelle ersetzt werden. Eine Technik, welche dies leisten kann, die Simulation, wird deshalb zunehmend in die Testaktivitäten integriert.

Eine mit mehreren Projektengineeringen aus unterschiedlichen Anwendungsbereichen der Prozessleittechnik (PLT) durchgeführte Analyse [2] hat ergeben, dass bislang der Einsatz von Simulationswerkzeugen in größerem Umfang primär am zusätzlichen, vor allem für die Modellerstellung notwendigen Zeitaufwand scheitert. Dementsprechend wird ein solches Werkzeug nur dann akzeptiert werden, wenn es möglich ist, den projektierten Steuerungscode sowie alle zugehörigen Bedienkomponenten unverändert und ohne zusätzliche Instrumentierung in einer virtuellen Umgebung zu testen und das Simulationsmodell, das die physikalische Anlage modelliert, weitestgehend automatisiert zu generieren.

Ein vorgezogener Testzeitpunkt hat zur Folge, dass sich die Anforderungen an die Simulationsart im Laufe des Projektes ändern [3]: Während zu Beginn der Applikationsentwicklung – aufgrund der fehlenden Peripherie – weitestgehend der Ansatz der Systemsimulation [4] zum Einsatz kommt, findet sich am Ende der Applikationsentwicklung der „Hardware-in-the-loop“-Ansatz (HIL) [5]. Der Übergang vom einen zum anderen, bzw. die damit verbundene Substitution der virtuellen Steuerungskomponenten durch reale Hardware, erfolgt i.d.R. sequentiell, wobei Tests in jedem Zwischenstadium möglich sein müssen. Hieraus folgen zwei weitere Anforderungen an eine Simulationsumgebung: sie muss skalierbar in Bezug auf den Simulationsumfang und flexibel in Abhängigkeit von der Testvariante sein. In diesem Artikel soll die interne Funktionsweise des in [2] allgemein vorgestellten Simulationswerkzeuges erläutert sowie die hinter der automatischen Modellgenerierung stehende Methodik beschrieben werden.

2. Definition der Betriebsarten und Simulationsmodelle

Das Anwendungsgebiet des hier vorgestellten Werkzeugs ist die Simulation des verfahrenstechnischen Anlagenverhaltens zum Zwecke von Steuerungscodetests. Hierzu gehört die Implementierung der Bedienoberfläche ebenso wie eine Integration der grundlegenden physikalischen Zusammenhänge. Da der Funktionsumfang des Werkzeugs auf die Unterstützung von Testaktivitäten optimiert ist, sollte es nicht als Ersatz für die Erstellung von Operatortrainings-Werkzeugen etc. angesehen werden. Im Hinblick auf eine Modellierung gemäß der in Kapitel 1 zitierten kundenseitigen Anforderungen sowie der Fokussierung auf Steuerungscode im FAT wird zunächst eine weitestgehend statische Parametrisierung [6] implementiert. Dies bedeutet, dass werkstofftechnische und chemische Parameter zunächst als Konstanten implementiert wurden, die sich nicht in Abhängigkeit anderer Zustandsgrößen ändern. Neben dem fließenden Übergang von der reinen Systemsimulation hin zur HIL-Simulation unterstützt das Werkzeug sowohl die Fokussierung auf Tests einzelner Objekte (vertikale oder objektweise Simulation) als auch die Simulation mehrerer durch physikalische Zusammenhänge gekoppelte Objekte (horizontale Simulation). Bei der objektweisen oder vertikalen Simulation werden nur das jeweilige Objekt und der zugehörige Teil des Steuerungscode berücksichtigt. Als Beispiel sei die Aktivierung von Alarmmeldungen im Controller in Abhängigkeit eines Sensorwertes oder die Aktivierung eines Ventilfeedbacksignals in Abhängigkeit der Ventilstellung genannt.

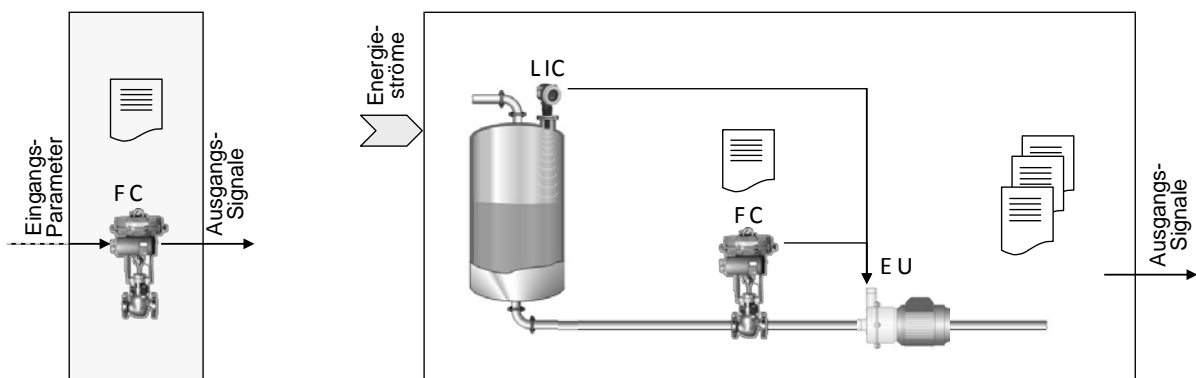


Bild 1: Beispiel zu objektweiser / vertikaler (links im Bild) und horizontaler (rechts im Bild) Simulation

Im Gegensatz hierzu bildet die horizontale Simulation das Zusammenspiel mehrerer durch Informations-, Energie- und Materialströme gekoppelter Objekte ab. Sie kann einzelne

Funktionseinheiten bis hin zur gesamten Anlage umfassen, was insbesondere beim Testen von Schrittketten notwendig ist, die auf verschiedene Aktoren wirken und Informationen von unterschiedlichen Sensoren beziehen. Der in Bild 1 gezeigte Ausschnitt einer Prozessanlage mit einem geregelten Ventil, einer Pumpe und einem Tank demonstriert deutlich die Notwendigkeit der horizontalen Simulation für den Test der Verriegelungslogik der Pumpe: Erst durch Berücksichtigung des Feedbacksignals des Ventils kann ein „Trockenlaufen“ der Pumpe (d.h. ein Förderversuch ohne Fluid) verhindert werden.

3. Aufbau und Funktionsweise der Simulationsumgebung

Nachdem in Kapitel 2 auf die Grundlagen der zur Verfügung stehenden Simulationsmodi und Modellierungsparameter eingegangen wurde, soll an dieser Stelle auf die eigentliche Funktionsweise der semiautomatischen Modellgenerierung sowie die modulare Struktur des entwickelten Werkzeuges eingegangen werden.

Die Excel-Simulationsumgebung basiert auf einem Bibliothekskonzept. Dies bedeutet, dass je nach Anlagenkonfiguration Instanzen der am Prozess beteiligten Objekte (Aktoren, Sensoren, Behälter, Wärmetauscher, Materialflusselemente, etc.) angelegt werden müssen. Im Gegensatz zu einer herkömmlichen Modellierung erfolgt dieser Schritt automatisiert, indem die PLS-Engineeringumgebung als Datenquelle verwendet wird. Aus der dort angelegten funktionalen Struktur werden die einzelnen Objekte, welche für die Simulation als virtuelle Modelle vorhanden sein müssen, in Excel geladen. Dabei wird für jedes Objekt eine Zeile im Tabellenblatt reserviert. Als konkretes Beispiel kann das in Bild 2 gezeigte Objekt „FIC101“ herangezogen werden. Diese Funktion steht für die Messung und Regelung eines Durchflusses innerhalb eines Rohrsegments. Für das hieraus resultierende Simulationselement bedeutet dies die Instanziierung einer mechanischen Regelventilcharakteristik gemäß einem Übertragungsgliedverhalten – für ein Ventil bietet sich i.d.R. ein lineares bzw. PT1-Verhalten an. Des Weiteren müssen die hierfür benötigten Ein- und Ausgangssignale, die physikalische Berechnung eines Durchflusses in Abhängigkeit von an der Wirkkette des Massenstroms beteiligten Elementen (in Bild 1-rechts Tank, Ventil, Pumpe) sowie eine Bereitstellung von optionalen Eingriffs- und Parametrisierungsmöglichkeiten seitens des Testingenieurs implementiert werden.

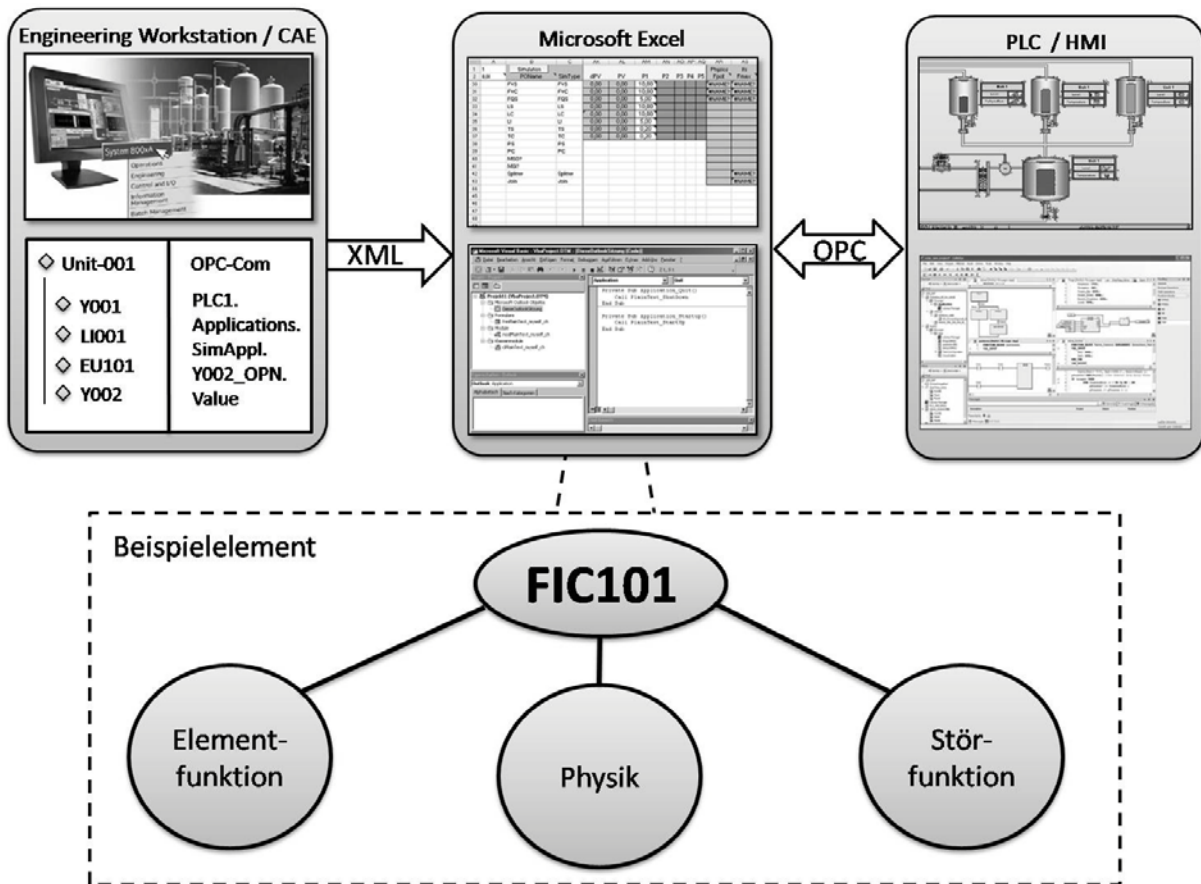


Bild 2: Testumgebung des Simulationswerkzeuges

Die Modularisierung der Simulationsumgebung erfolgt entsprechend dieser Aufgabengebiete in die Bereiche:

- (1) Objektdefinition – Import der PLS-Objekte in Excel und Zuweisung des entsprechenden Simulationstyps.
- (2) Prozessdefinition – Definition der Material- Informations- und Energieströme auf Basis von CAE-Daten bzw. manuell.
- (3) Störfunktionen – Aufschaltung von Störfunktionen zur Simulation von hardware- oder übertragungstechnischen Fehlern.
- (4) Elementverhalten – Vorgabe des mechanischen Verhaltens einzelner Objekte anhand von vordefinierten Übertragungsgliedern.
- (5) Physikkalkulation – Berechnung von Materialströmen und Energieübergängen in Abhängigkeit aufeinander einwirkender Objekte.
- (6) Signalverarbeitung – Empfang- und Sendemodul für Aktor- und Sensorsignale.

Nachdem das FIC101-Objekt in Excel importiert wird, werden für die zuvor genannten Module Standardfunktionen aus einer hinterlegten Bibliothek geladen. Die Zuweisung der Funktionselementcharakteristik des PLS zum entsprechenden, für das Objekt benötigten Simulationstyp erfolgt über eine in Excel hinterlegte „Mapping-Sektion“, welche einmalig definiert werden muss. Die für die Parametrisierung der Objekte erforderlichen Daten, wie z.B. Prozessobjektyp, EA-Signale, Messbereiche, Grenzwerte, etc., können direkt aus dem Engineering-Werkzeug (hier dem ABB Engineering-System 800xA) ausgelesen werden. Ähnlich wie die Objekte, wird auch die Kommunikation zwischen Simulationsumgebung und Prozessleitsystem automatisiert initialisiert – vgl. hierzu Kapitel 5. Nachdem zu diesem Zeitpunkt die Definition der Objekte (Modul 1), die Zuweisung des Elementverhaltens (Modul 4) und die Rangierung der benötigten E/A-Signale (Modul 6) abgeschlossen sind, ist der Test einzelner Objekte, vgl. Kapitel 2, an dieser Stelle bereits möglich.

Die Aufschaltung einer Störfunktion dient sowohl dem Test von mechanischen als auch signalübertragungstechnischen Problemen und kann für sämtliche, von der Simulationsumgebung an das PLS übertragenen Signale gleichermaßen geschehen. Zu diesem Zweck sind vorimplementierte, aus der Regelungstechnik bekannte Signalvarianten, wie z.B. eine Sinuswelle, ein Rechteckimpuls, ein Sägezahn, eine Rampenfunktion, etc. verfügbar (Modul 3), die online, also während der laufenden Simulation durch den Testingenieur aktiviert oder deaktiviert werden können. So ist es wiederum am Beispiel des Regelventils möglich, das Rückmeldesignal über die jeweilige Ventilstellung durch einen Konstantwert „0“ zu ersetzen, um so entweder einen Kabelbruch oder einen mechanischen Defekt zu simulieren.

In Erweiterung zu den bisher durchgeführten Simulationen werden bei einer objektübergreifenden Simulation zusätzlich die Prozessdefinition (Module 2) und die Physikkalkulation (Modul 5) benötigt. Für die Generierung des gerichteten Material-, Energie-, und Informationsflusses (Modul 2) wird dabei auf Informationen zurückgegriffen, welche aus einem Planungssystem (CAE-System) ausgelesen werden können. Hierzu bieten sich Datenaustauschformate wie beispielsweise CAEX [7] an. Im Anschluss können die eingetragenen Prozessinformationen durch den Testingenieur noch geändert werden.

Im Gegensatz zum Simulationslauf findet die Generierung des Prozessmodells „offline“, d.h. vor Testbeginn und damit nicht Performance-kritisch statt. Aus diesem Grund können die für die Generierung des Simulationsmodells notwendigen Algorithmen in der Excel unterlagerten Programmierumgebung VBA (Visual Basic for Applications) hinterlegt und ausgeführt werden. Die instanziierten und in die entsprechenden Zellen der Physikkalkulation (Modul 5) eingetragenen Formeln zur elementübergreifenden Berechnung der physikalischen Werte,

z.B. Materialfluss, statische und dynamische Drücke, Füllstände, Temperaturen, Wärmeübergänge etc., werden durch den Modellgeneratoralgorithmus automatisch manipuliert. Zu Beginn der Manipulation detektiert das sich an der Materialflussrichtung orientierende Programm ein Startelement aus der Gruppe der Primärobjekte. Primärobjekte sind definiert als passive, (keine Aktoren), den Materialfluss verzweigende, zusammenführende, unterbrechende, beginnende oder beendende Elemente. Tabelle 1 zeigt Beispiele mit entsprechender Zuweisung:

Tabelle 1: Primärelemente im Materialfluss

Beeinflussungsart: Materialfluss	Primärelement
beginnen	Quelle
beenden	Senke
beginnen / beenden	Tank
verzweigen	Rohrverzweigung
zusammenführen	Rohrzusammenführung

Hiermit werden virtuelle Materialflusslinien bis zum Erreichen eines weiteren Primärobjekts gebildet. In diesem Zusammenhang ist es nicht notwendig, einen in sich geschlossenen Materialfluss als Basis zu haben. Quelle und Senke können jeweils offene Enden oder Übergänge zu anderen Anlagenteilen modellieren, was die Möglichkeit des Testens einzelner dem PLS-Aufbau folgender Einheiten ermöglicht. Die Definition der einzelnen Materialflusslinien wird in vollem Umfang vom Generator-Algorithmus verwaltet. Es ist möglich, dass eine Anlage aus mehreren voneinander getrennten Materialflüssen besteht. Dies würde sowohl unterschiedliche Quellen, als auch Senken, bzw. Kreisläufe, bedeuten. In diesem Fall sucht der Algorithmus alle möglichen Startpunkte und legt entsprechend viele Materialflusslinien an.

4. Berechnungsmethodik und Abarbeitungsreihenfolge

Im Gegensatz zum Algorithmus, der für die Generierung des Simulationsmodells eingesetzt wird, findet die Berechnung aller Zeitwerte – online bei laufender Simulation – direkt im Excel-Tabellenblatt statt. Dieser verwendete Ansatz erlaubt Zeitschrittweiten von <100 Millisekunden bei gegebener Echtzeitreue und liegt damit innerhalb einer für die verfahrenstechnische Industrie benötigten Auflösung.

Als für die Simulationsumgebung gewählte Berechnungsmethode wurde das Verfahren nach Euler-Cauchy [8] umgesetzt. Dies beruht auf Gradienten, die für jeden Zeitwert berechnet und aufaddiert werden. Hierfür wird je eine Zelle für die Kalkulation des sich einstellenden Deltas und eine für den Absolutwert der Prozessvariablen reserviert. In Bild 3 wird links, unabhängig von den verwendeten Einheiten, die prinzipielle Funktionsweise der erläuterten Berechnungsmethodik dargestellt. Im rechten Bildteil sind qualitativ drei unterschiedliche Prozesswertverläufe dargestellt, die sich durch die Wahl unterschiedlicher Zeitschrittweiten ergeben. Die Unterschiede machen deutlich, dass im vorliegenden Werkzeug, wie auch in der Simulationstechnik im Allgemeinen, ein Gleichgewicht von geforderter Genauigkeit zu vertretbarem Rechenaufwand gefunden werden muss. Dieser Zusammenhang spiegelt sich ebenfalls in der Abarbeitungsproblematik von Excel wider, die im Folgenden erläutert wird.

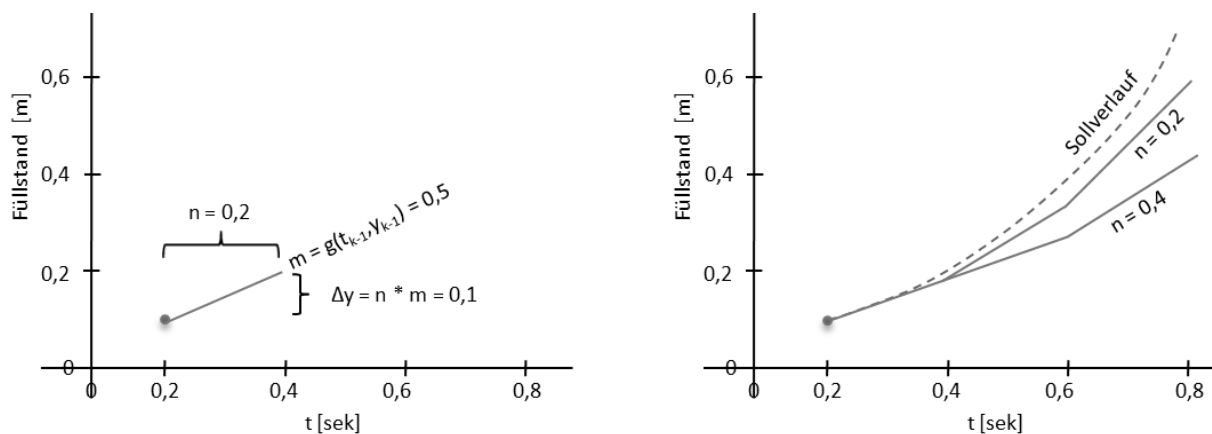


Bild 3: Explizites Euler-Verfahren für die Numerik in Excel

Der in Excel hinterlegte Berechnungsalgorithmus bearbeitet die Zellen im jeweiligen Arbeitsblatt von links oben nach rechts unten. Dadurch kann es vorkommen, dass Formeln, die auf eine Variable zurückgreifen, deren Zeitwert für den aktuellen Kalkulationsschritt erst im Anschluss daran berechnet wird, mit „alten“ Werten kalkuliert werden. Die Simulation hat bei einigen Ergebnissen demnach eine Wertedifferenz, die der Änderung innerhalb eines Zeitschrittes entspricht. Diesem Problem wird im vorliegenden Fall durch die Wahl eines minimalen, nur durch die Berechnungsgeschwindigkeit von Excel begrenzten Zeitschrittes entgegengewirkt.

Die Berechnung der Zeitwertänderungen und Absolutwerte in verschiedenen Zellen bedingt, dass Zellreferenzen auf die eigene Zelle verweisen und resultiert in so genannten Zirkelbezügen. In der Programmierung ist dieser Effekt durch die Implementierung von Endlosschleifen bekannt. Den Endlosschleifen kann durch eine Ausstiegsbedingung

entgegnet werden. Dies wird in Excel durch eine vorab zu definierende Anzahl von Berechnungsiterationen pro Kalkulationsschritt (hier auf den Wert „1“ gesetzt) erreicht.

5. Kommunikationsaufbau und -Initialisierung

Entsprechend der Signalrangierung der realen Anlage, die während des Detail-Engineerings erstellt wird, muss auch die virtuelle Anlage mit den Soft-Controllern (Systemsimulation) bzw. den realen Controllern (HIL) gekoppelt werden. Im Unterschied zur physikalischen „Signalverdrahtung“ wird das Simulationsmodell nicht über die Ein- und Ausgänge der EA-Module, sondern direkt mit den internen Variablen verbunden, die ein Abbild des EA-Systems im Speicher des Controllers darstellen. Hierbei wird jedes Signal der simulierten Sensoren und Aktoren mit den entsprechenden EA-Variablen verbunden. Da die Implementierung dieser virtuellen Signalrangierung, abhängig vom Simulationsmodell, einen erheblichen, mit der realen EA-Rangierung vergleichbaren Aufwand darstellen kann, ist eine automatisierte Generierung der Signalverbindungen erforderlich. Dies stellt nicht nur eine notwendige Voraussetzung zur Akzeptanz einer virtuellen Testumgebung bei den Anwendern dar, sondern trägt auch wesentlich zur korrekten Funktionsweise der Simulationsumgebung bei und verhindert somit, dass die Testumgebung selbst zum Testobjekt mutiert.

Die Grundlage dieses Generierungsprozesses bilden die Signalschnittstellen der einzelnen Steuerungs-Funktionen, welche im Engineering-System vorliegen. Diese werden in Excel eingelesen und anhand von Abbildungsregeln, die im Wesentlichen auf Namenskonventionen basieren, automatisch mit den entsprechenden Signalen des Simulationsmodells „verdrahtet“. Es besteht die Möglichkeit, alle Informationen manuell aufzubereiten; in den Fällen, in denen keine automatische Abbildung möglich ist, ist dies auch erforderlich.

Nachdem die Kommunikationsverbindungen in MS Excel definiert sind, können Signaldaten via OPC zwischen allen am Prozess beteiligten Komponenten ausgetauscht werden. Die OPC-Kommunikation findet erstmals initial, zu Beginn einer Testsequenz statt; während der Simulation jeweils nach der Berechnung eines durch den Benutzer definierbaren Zeitschrittes. Da der Kommunikationsaufwand einen erheblichen Anteil der CPU-Auslastung darstellen kann, müssen Maßnahmen zur Performance-Optimierung getroffen werden. In Untersuchungen während der Entwicklungsphase hat es sich als zielführend erwiesen, Datenänderungen nicht mittels eines im Sinne der Abarbeitung aufwändigen "Polling"- (Abfrage) Mechanismus zu lesen. Stattdessen wird die Übertragung aktiv durch die jeweils betroffenen Komponenten mitgeteilt („Subscription-Basis“). Auch innerhalb einer

Netzwerkconfiguration, in der mehrere Controller mit dem Simulationsmodell über Ethernet verbunden werden, stellt die in dieser Weise optimierte OPC-Kommunikation keine kritische CPU-Belastung dar.

Beim Einsatz realer Controller, die bereits mit ihren EA-Modulen konfiguriert sind, muss der Controller einen Simulationsmode für die Signale unterstützen. Die Problematik hierbei ist, dass die EA-Module bereits Prozesswerte liefern und somit eine Methodik zur Störgrößenaufschaltung vorgesehen werden muss. Im Simulationsmode kann daher das Überschreiben der via OPC übertragenen Signaldaten durch Daten der realen Ein/Ausgänge des Controllers verhindert werden. Hierzu bietet der zur Verifikation verwendete ABB Controller "AC 800M" die Möglichkeit, einzelne Signale zu simulieren, was eine stufenweise Transition von simulierten Anlagenkomponenten zu realen Anlagenteilen ermöglicht, vgl. Kapitel 2. Die erforderliche Konfiguration der Signalsimulation wird ebenfalls automatisch erstellt und an die angekoppelten Controller übertragen. Am Ende einer Testsequenz wird die Signalsimulation automatisch wieder aufgehoben.

6. Zusammenfassung

Mit der Entwicklung einer modellbasierten Simulationsumgebung zur Unterstützung des Applikationstests von Prozessleitsystemen hat ABB in Kooperation mit der HSU einen wichtigen Schritt in Richtung frühzeitige Fehlererkennung unternommen. Die hieraus entstehenden Vorteile in Bezug auf höhere Qualität, frühere Produktionsstarts und höhere Effektivität durch zielgenauere Implementierung bilden die Grundlage eines erfolgreichen Projektes des Systemlieferanten. Durch den modularen Aufbau, den fortgeschrittenen Automatisierungsgrad sowie die Flexibilität in Bezug auf die Anbindung an bestehende Prozessleitsysteme hat sich der im Zuge einer Universitätskooperation entwickelte Prototyp der Simulationsumgebung in Tests gut bewährt. Auch bestätigt das positive Feedback der beteiligten Anwender den gewählten Ansatz.

Literatur

- [1] Stetter, R.; Erben, M.: Automatisches Testen von SPS-Steuerungssoftware; Steuerungstechnik aktuell 2008 – Software; S. 31-35; Oldenbourg Industrieverlag, 2008
- [2] Barth, M.; Weber, P.; Fay, A.; Greifeneder, J.: Modellbasierte Prozesssimulation für Steuerungstests auf Excel-Basis – Mechatronik, VDI Fachkongress, Wiesloch 2009.

- [3] Drath, R.; Weber, P.; Mauser, N.: Virtuelle Inbetriebnahme – ein evolutionäres Konzept für die praktische Einführung ; Automation 2008 - S.73.

- [4] Abel, D.: Integration von Advanced Control in der Prozessindustrie; Rapid Control Prototyping; Wiley-VCH Verlag Weinheim, 2008.

- [5] Pritschow, G.; Rüdele, H.; Röck, S.: Eigenschaften und Anwendungen von Hardware-in-the-Loop-Simulation in der Steuerungstechnik - Simulationstechnik in der Produktion; VDI-Verlag 2006, S. 32...42. (VDI Fortschritt-Berichte Reihe 2 Nr. 658).

- [6] Kramer, U.; Neculau, M.: Simulationstechnik - S.98 - Carl Hanser Verlag München Wien 1998.

- [7] International Electrotechnical Commission: IEC PAS 62424 - Grafische Darstellung von PLT-Aufgaben und Datenaustausch zu Engineering-Systemen.

- [8] Kahlert, J.: Simulation technischer Systeme: Eine beispielorientierte Einführung – Kapitel 3; S.112; Vieweg +Teubner Verlag, 2004