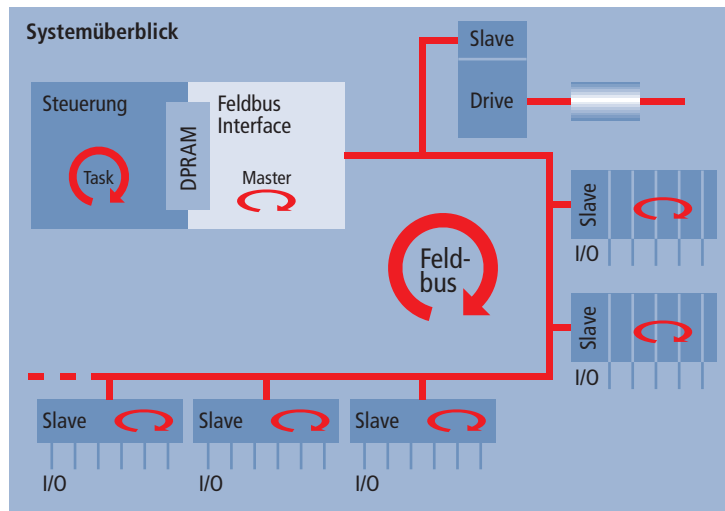


Die Leistungsfähigkeit von Feldbussystemen in der Fertigung wird gerne anhand der erreichbaren Kommunikations-Zykluszeiten beschrieben. Aufgrund dieser Kennzahlen wird dann bereits die Echtzeitfähigkeit der unterschiedlichen Systeme beurteilt. Differenzierte Ansätze berücksichtigen darüber hinaus noch die Wiederholgenauigkeit der Kommunikationszyklen. Meist werden jedoch weitaus kritischere Parameter, wie etwa das Synchronisierungsverhalten, sowohl der zentralen Steuerungsapplikationen als auch der unterlagerten, dezentralen Applikationsteile, mit dem Feldbuszyklus außer acht gelassen.

Performance von Feldbussystemen in der Fertigung

Zykluszeit ist nicht alles

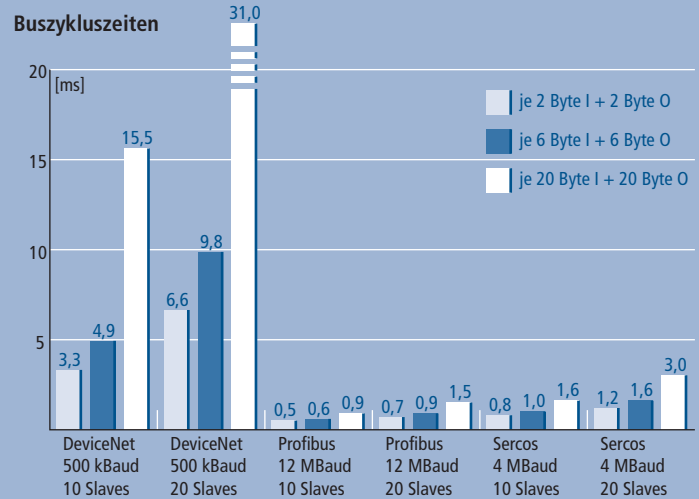


Weitere entscheidende Kenngrößen sind das Verhältnis von Applikations- und Kommunikations-Zykluszeiten, sowie Softwarelaufzeiten und -jitter. Feldbussysteme in der Fertigung unterscheiden sich zwar in ihrer Topologie – für die Untersuchung der Performance-Kenngrößen ist diese jedoch zweitrangig: Die Unterschiede bei den physikalischen Laufzeiten von Bus, Ring und Baumstrukturen sind vergleichsweise gering. Selbstverständlich sind die Baudraten und vor allem die Protokolleffizienz dagegen von Bedeutung. Bei den zyklischen Systemen gehen diese vor allem in die Zykluszeit ein.

Systemaufbau

Betrachtet wird ein Steuerungssystem mit dezentralen Feldbusgeräten. Die Steuerung (z. B. ein IPC oder eine SPS) führt ein oder mehrere Anwendungsprogramme (z. B. NC- oder SPS-Tasks) zyklisch aus. Das Feldbus Interface ist als Steckkarte mit eigenem Prozessor ausgeführt. Der Austausch der Prozessdaten mit der Steuerung erfolgt über einen gemeinsamen Speicherbereich (DPRAM), auf den wechselseitig zugegriffen wird.

Das einfachste – und auch am weitesten verbreitete – Kommunikationsprinzip für Feldbussysteme ist der zyklische Datenaustausch zwischen einem zentralen Feldbus-Master und vielen dezentralen Feldbus-Slaves. Für die vorliegende Betrachtung ist dabei zunächst unerheblich, ob der Datenaustausch und der Buszugriff



dabei per Polling (wie etwa bei Profibus), per Zeitscheibenverfahren (wie etwa bei Sercos) oder per Schieberegisterprinzip (wie etwa bei Interbus) erfolgt. Selbst das CAN basierte DeviceNet, das auf Buszugriffsebene dem Multi-Master Prinzip folgt, ist auf Protokollebene fast immer ein Polling-System und kann daher in der Regel als zyklischer Feldbus betrachtet werden. Auch ein CANopen System, das typisch auf Multi-Master-Art ereignisgesteuert kommuniziert, kann im streng zyklischen Modus betrieben werden – etwa zur Ansteuerung von Achsen.

Für das betrachtete allgemeine Feldbusystem übernimmt das Feldbus Interface der Steuerung die Aufgabe des Masters, der den Buszyklus kontrolliert. Die dezentralen Feldbusgeräte sind dementsprechend Slaves. Sie antworten nur auf Anforderung durch den Master und können deshalb neue Eingangsdaten nur im Takt des Buszyklus kommunizieren.

Zykluszeit Feldbus

Die Zykluszeit des Feldbusystems ist die bekannteste Kenngröße. Sie hängt in erster Linie von der Baudrate und von der Protokolleffizienz ab. Das Diagramm „Buszykluszeiten“ gibt einen Überblick über typische Zykluszeiten einiger Feldbussysteme in Abhängigkeit von Baudrate, Anzahl Teilnehmer und Anzahl der jeweils ausgetauschten Datenbytes. Bei diesen Berechnungen mussten naturgemäß einige Annahmen gemacht werden: So wurde bei DeviceNet eine Buslast von 80 % für reinen Polling Betrieb zugrunde gelegt. Bei Profibus wurde von einer Verzögerung zwischen zwei Zyklen (etwa für Zugriff auf das DPRAM) von 0,3 ms ausgegangen, außerdem mussten die Protokoll-Zeitkonstanten (etwa die Station Response Time) für typische Geräte abgeschätzt werden.

Durchlaufverzögerung Master/Firmwarezyklus Master

Es wird von einem intelligenten Feldbus Interface auf der Steuerung ausgegangen, das über einen eigenen Prozessor verfügt. Dieser Feldbusmaster kopiert das Prozess-Ausgangsabbild via Feldbus zu den Slaves und sammelt die Eingangsdaten ein. Das Eingangs-Prozessabbild wird zum DPRAM der Steuerung kopiert. Aus Gründen der Datenkonsistenz kopieren die meisten Implementierungen das Eingangs-Prozessabbild erst dann ins DPRAM der Steuerung, wenn es vollständig ist. Bei einem zyklischen Feldbusystem wird also erst der komplette Feldbuszyklus abgewartet. Das Umkopieren der Eingangsdaten kann sich auch dadurch verzögern, dass zunächst auf die Freigabe des DPRAM gewartet werden

muss. Typische Verzögerungszeiten in Feldbus Mastern liegen in der Größenordnung von 0,1 bis 3 ms.

Zykluszeit Steuerung („Task“)

Die Feldbus Performance sollte sinnvollerweise in vernünftiger Relation zur Performance der dazugehörigen Steuerung stehen. Eine sinnvolle Kenngröße ist hier die Zykluszeit der entsprechenden Steuerungstask für typische Anwendungen.

Bei PC-basierten Steuerungssystemen werden, aufgrund der großen Leistungsfähigkeit der PC Prozessoren Task-Zykluszeiten von deutlich unter 1 ms erreicht. Dies macht vor allem für schnelle Aufgaben Sinn – wie etwa die Achsregelung. Für typische SPS-Anwendungen werden meist Zykluszeiten in der Größenordnung von 2...10 ms gewählt.

Bei „klassischen“ SPS-Lösungen findet man dagegen nach wie vor typische Zykluszeiten in der Größenordnung von 10...20 ms.

Durchlaufverzögerung Slave/Firmwarezyklus Slave

Auch auf den Slave Geräten arbeitet bei den meisten Feldbussystemen ein Prozessor mit entsprechender Firmware. Diese wird zyklisch aufgerufen, fragt dann die Eingänge ab und kopiert die Ausgangsdaten zu den Ausgängen.

Bei modularen Slave Geräten, wie etwa dem Beckhoff Busklemmensystem, gibt es einen lokalen Bus, der die I/O-Baugruppen mit dem eigentlichen Feldbusgerät, dem Buskoppler, verbindet. Dieser lokale Bus ist zwar sehr schnell, benötigt aber dennoch je nach Konfiguration 0,1... 2 ms für ein komplettes Update.

Aber auch bei Geräten mit feststehender Konfiguration kann der lokale Firmwarezyklus nicht vernachlässigt werden – nur in dessen Zyklusraster werden die Eingänge abgefragt bzw. die Ausgänge aktualisiert.

Schnittstellen: Synchronisierung oder Freilauf

Bei dem betrachteten typischen Feldbussystem haben wir es mit einem Mehrprozessorsystem mit vielen Schnittstellen zu tun. Die Programme auf den verschiedenen Prozessoren laufen in der Regel zyklisch, seltener auch ereignisgesteuert (interruptgetrieben) ab. Am einfachsten ist es, die Prozesse im Freilauf zu betreiben: Entkoppelt durch einen gemeinsamen Speicherbereich können sowohl zyklische als auch ereignisgesteuerte Prozesse zeitlich weitgehend unabhängig voneinander abgearbeitet werden. Auch der Feldbus selbst kann hier als verteilter gemeinsamer Speicherbereich interpretiert werden.

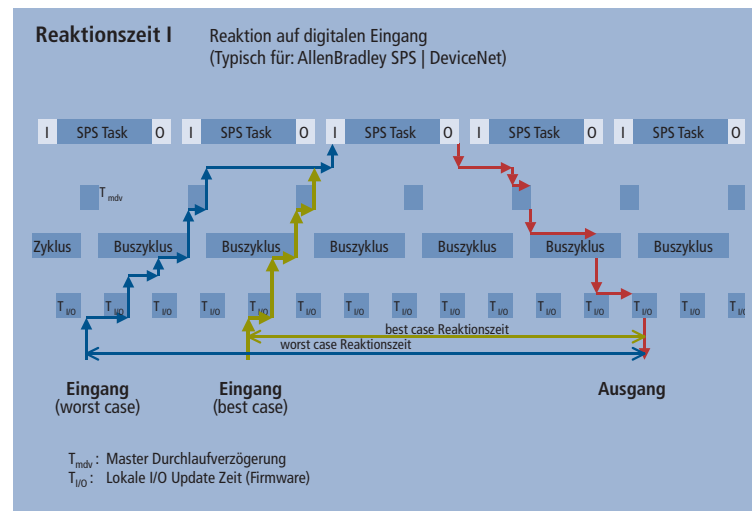
Alternativ können die unterschiedlichen Prozesse auch synchronisiert werden. Auf Slave-Seite bedeutet dies, dass der Firmwarezyklus bzw. der lokale Buszyklus mit dem Feldbuszyklus abgestimmt ist. Die Ausgänge einer I/O Baugruppe werden dann stets unmittelbar nach Empfang der neuen Ausgangsdaten aktualisiert. Die Eingangsdaten werden bei diesem Firmwarezyklus gelesen, aber erst beim nächsten Feldbuszyklus zur Steuerung kommuniziert.

Besonders interessant ist die Synchronisierung des Feldbus Masters mit der Steuerung. Bei schnellen Bussystemen wechseln sich Feldbuszyklus und Steuerungstask jeweils ab, sodass die Task jedes Mal aktuelle Eingangsdaten erhält. Die Ausgänge werden direkt nach Abschluss der Task mit dem nächsten Feldbuszyklus geschrieben. Bei sehr schnellen Tasks (z. B. NC Tasks zur Achsregelung) oder bei relativ langsamen Bussen ist es auch möglich, die Ein- und Ausgangsdaten am Task-

anfang zu kopieren und dann den Buszyklus zu starten. Dann sind die Eingangsdaten zwar einen Buszyklus alt, die Task muss jedoch nicht auf den Bus warten. Die Auswirkungen der Kenngrößen auf die Reaktionszeit und den Determinismus des Systems werden im Folgenden anhand verschiedener Beispielkonfigurationen untersucht.

Szenario I: durchgehender Freilauf, „mittelschnelles“ Bussystem

In diesem Szenario gleicht die Buszykluszeit der Taskzykluszeit. Der Feldbuszyklus ist gegenüber der Steuerungstask freilaufend. Auch der lokale I/O-Zyklus ist nicht mit dem Buszyklus synchronisiert.

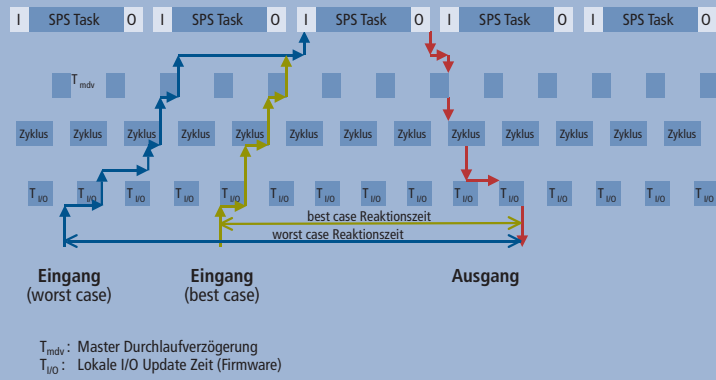


Untersucht wird nun die Reaktionszeit auf ein beliebiges Eingangssignal. Im worst-case Fall tritt dieses gerade auf, nachdem der lokale I/O-Zyklus die Eingänge abgefragt hat (bzw. passiert den Hardware-Eingangsfiler gerade dann). Nun muss zunächst ein lokaler Firmwarezyklus abgewartet werden, bis die Firmware des dezentralen I/O-Knotens Kenntnis vom Eintreten des Ereignisses hat. Sie kann diese Information aber erst beim nächsten Abfragen der Eingangsdaten durch den Feldbus mitteilen – im worst case muss darauf einen ganzen Feldbuszyklus lang gewartet werden.

Nach Abschluss des Feldbuszyklus befindet sich die Eingangsinformation nun im Feldbus-Master, also in der Anschaltbaugruppe der Steuerung. Der Master kopiert die Daten ins DPRAM der Steuerung und gibt das DPRAM wieder frei – benötigt hierfür die Master-Durchlaufverzögerung. Bei einem freilaufenden System hat nun im worst case die Steuerung gerade mit der Steuerungstask begonnen, wenn die Master-Baugruppe das DPRAM freigibt, d. h. die Eingangsdaten werden von dieser Task nicht mehr erreicht. Es dauert nun also eine komplette Steuerungstask, bis die Eingangsdaten von der Steuerung wahrgenommen und verarbeitet werden. Im worst case muss davon ausgegangen werden, dass auch der Weg zu den Ausgängen unglücklich verläuft: Der Start des nächsten Masterzyklus wird von der SPS-Task gerade verpasst, sodass ein Feldbuszyklus abgewartet werden muss. Nun werden die Ausgangsdaten übertragen und verpassen gerade den lokalen I/O-Zyklus.

Reaktionszeit II

Reaktion auf digitalen Eingang
(Typisch für: Siemens SPS | Profibus, AB | ControlNet)



Wie aus der Grafik ersichtlich, summiert sich die Reaktionszeit bei einem solchen System bei ungünstlichem Timing auf > 4 Buszykluszeiten oder entsprechend viele SPS-Task-Zeiten. Selbst der Jitter zwischen Idealfall und worst case – ein Maß für den Determinismus des betrachteten Systems – liegt im Bereich von 2 SPS-Tasks. Das zeigt, dass der systemeigene Determinismus des gewählten Feldbuses in diesem Fall vernachlässigbar ist.

Szenario II: durchgehender Freilauf, schnelleres Bussystem

Nun wird versucht, das System durch einen schnelleren Feldbus reaktionsschneller zu machen: Der „mittelschnelle“ Feldbus aus Szenario I wird durch einen schnellen Feldbus ersetzt, der gegenüber dem ersten Fall nur noch halb so lange Zykluszeiten aufweist. Weiterhin bestehen bleibt der Freilauf des gesamten Systems.

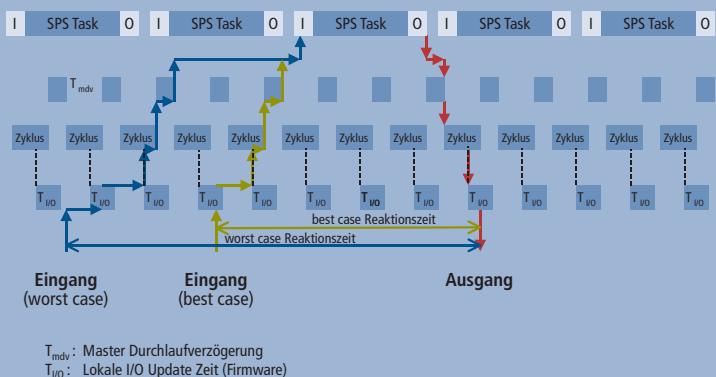
Beim Vergleich der beiden Grafiken wird deutlich, dass sich die Reaktionszeit zwar im Mittel um etwa 20-25 % verbessert hat – der „best case“ ist jedoch in beiden Fällen gleich. Insgesamt ein eher enttäuschendes Ergebnis – immerhin war der Bus nun doppelt so schnell. Weiter ist zu sehen, dass die Eingangsdaten einiger Buszyklen nicht verarbeitet werden – sie werden vom nächsten Buszyklus überschrieben, da zwischenzeitlich kein Taskstart erfolgt ist. Ebenso transportiert nur jeder zweite bis dritte Buszyklus neue Ausgangsdaten. Der Feldbus ist zwar schneller, aber teilweise ineffizient.

Szenario III: Master freilaufend, Slave synchronisiert

Das Ergebnis soll nun durch Einführung von Synchronisierungsmechanismen weiter verbessert werden. Zunächst wird der Slave-Firmwarezyklus bzw. das lokale

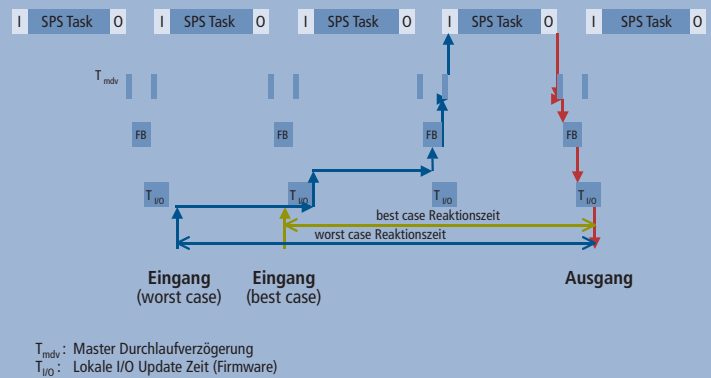
Reaktionszeit III

Reaktion auf digitalen Eingang
(Typisch für: Siemens SPS | Profibus I/O)



Reaktionszeit IV

Reaktion auf digitalen Eingang



I/O-Update mit dem Feldbuszyklus synchronisiert. Das führt nur auf Ausgangsseite zu einer Verbesserung des Reaktionsverhaltens: Die Ausgänge werden stets unmittelbar nach Erhalt neuer Daten aktualisiert. Die Verbesserung entspricht im Mittel der halben Dauer eines lokalen Firmwarezyklus auf der I/O-Baugruppe. Dementsprechend verbessert sich auch der Determinismus des Systems.

Szenario IV: schneller Bus, voll synchronisiert

Nun wird auch der Feldbuszyklus mit der Steuerungstask synchronisiert. Das ausgangsseitige Reaktionsverhalten ist optimal, die Ausgänge werden auf schnellstmögliche Weise deterministisch gesetzt. Allerdings vergehen eingangsseitig immer noch mehrere Taskzyklen (bzw. Buszyklen), bis die Eingangsinformation verarbeitet werden kann. Der Determinismus des Systems ist zwar gut (er entspricht der Taskzeit), die Reaktionszeit bleibt aber insgesamt nach wie vor unbefriedigend. Interessant ist auch, dass sich die Reaktionszeit durch den schnellen Bus (1 ms statt 5 ms) nur wenig verbessert hat.

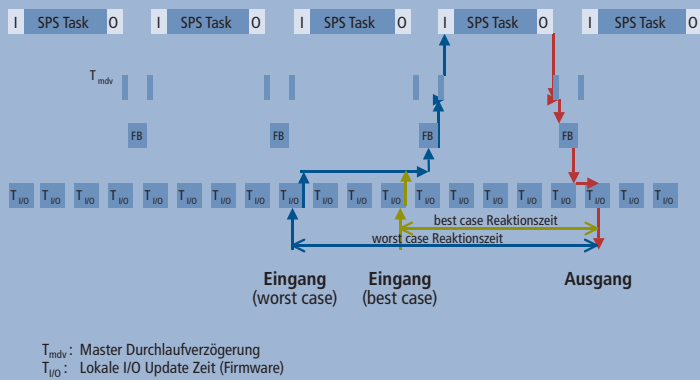
Szenario V: masterseitig synchronisiert, freilaufender Slave

Der Freilauf der Slave Firmware gegenüber dem Buszyklus ist zwar ausgangsseitig leicht nachteilig – eingangsseitig werden nun jedoch auch Informationen erreicht, die zwischen den „gespreizten“ Buszyklen auftreten. Bei Zeitverhältnissen, wie in der Grafik dargestellt – Task langsam gegenüber lokalem Firmwarezyklus – verbessert sich vor allem die Reaktionszeit.

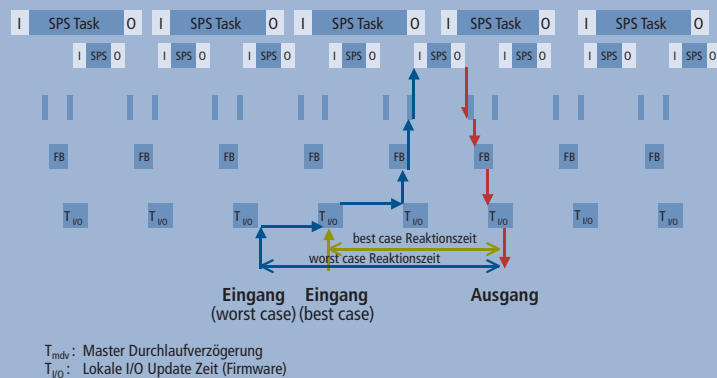
Szenario VI: Multitasking System, durchsynchronisiert

Eine weitere Verbesserung der Reaktionszeit – vor allem aber des Determinismus – kann nur noch durch Verkürzung der Taskzykluszeit erreicht werden. Bei gege-

Reaktionszeit V Reaktion auf digitalen Eingang



Reaktionszeit VI Reaktion auf digitalen Eingang Möglich mit: Beckhoff TwinCAT | Profibus | Sercos



bener CPU-Leistung ist das am einfachsten durch ein Multi-Tasking Steuerungssystem zu erreichen: Die zeitkritischen Aufgaben werden in einer kurzen Task programmiert, die mit dem (schnellen) Feldbussystem synchronisiert wird. Alle anderen Applikationsteile können dann in langsameren Tasks realisiert werden. Dieses Szenario weist optimale Reaktionszeiten bei gleichzeitig optimalem Determinismus auf. Es ist dies die übliche Systemarchitektur des TwinCAT Steuerungssystems.

Szenario VII: ereignisgesteuerte Kommunikation

Bislang wurden zyklische Bussysteme betrachtet. Einige Systeme – etwa CANopen oder auch DeviceNet – können jedoch auch im ereignisgesteuerten Modus betrieben werden. Bei DeviceNet heißt diese Betriebsart „Change of State“. Die Änderung eines Eingangs wird als Ereignis gewertet und selbsttätig kommuniziert, ohne Anforderung durch einen Master.

Messergebnisse für ein CANopen System mit ereignisgesteuerter Kommunikation zeigen das trotz vergleichsweise geringer Baudrate (500 kBaud) die Reaktionszeit einem vergleichbaren Profibus System recht nahe kommt. Allerdings weist das Profibus System diese Reaktionszeit für alle Eingänge auf, während beim CANopen System bei starker Busbelastung mit Verzögerungen zu rechnen ist.

Ergebnisse

Die Untersuchung der verschiedenen Szenarien führt zu folgenden Ergebnissen:

- Die Leistungsfähigkeit eines Feldbus vernetzten Steuerungssystems kann nicht anhand einer einzigen Kenngröße bestimmt werden.
- Die reine Zykluszeit des gewählten Bussystems ist in der Regel zweitrangig. Selbst zur Beurteilung des Echtzeitverhaltens des Bussystems taugt die Zy-

kluszeit nur bedingt: In vielen Fällen ist das Synchronisationsverhalten entscheidend.

- Wichtiger als die Zykluszeit des Bussystems ist in aller Regel die Zykluszeit der Steuerung: eine langsame Steuerung wird auch durch einen schnellen Feldbus nicht wirklich leistungsfähig.
- Für die Feldbuszykluszeit gilt im Allgemeinen: „schnell genug“ heißt schneller als die Zykluszeit der Steuerung.
- Synchronisierung verbessert zunächst den Determinismus des Systems, nicht unbedingt die Reaktionszeit.
- Bussysteme mit ereignisgesteuerter Kommunikation weisen eine sehr kurze Reaktionszeit auf.
- Eine optimale Reaktionszeit bei geringem Jitter wird nur bei sorgfältiger Abstimmung des Systems erzielt. Hierzu ist die Kenntnis der verschiedenen zeitlichen Faktoren erforderlich – sowie ein System, dass ein solches „Tuning“ erlaubt.
- Eine kurze Reaktionszeit ist dabei nur eines von mehreren Kriterien zur Bestimmung der Systemleistung. Bei vielen Applikationen mag der Determinismus des Systems entscheidend sein.

Bei der Betrachtung der Messergebnisse ist zu berücksichtigen, dass der in den Scope-Bildern ausgewiesene Jitter den Unterschied zwischen minimaler und maximaler Reaktionszeit auf externe Ereignisse darstellt. Dieser Jitter kann niemals kleiner als die Taskzykluszeit werden. Er entspricht nicht dem Jitter einer zyklischen, über den Bus geschlossenen Regelschleife – etwa bei der Antriebsregelung. Dieser ist in etwa um die Taskzykluszeit kürzer.

Zusammenfassung

Für die Beurteilung der Leistungsfähigkeit eines Feldbussystems ist die Buszykluszeit nur eine unter vielen Kenngrößen. Weitere wichtige Werte sind das Synchronisationsverhalten sowie die Durchlaufzeiten durch die Feldbusanschlüsse auf Master und Slave-Seite. Für die Auswahl eines geeigneten Bussystems ist dabei die Kenntnis der Steuerungszykluszeiten von großer Bedeutung. Anhand von einigen typischen Szenarien lassen sich die Zusammenhänge der Kenngrößen darstellen. Dabei wird deutlich, dass die Buszykluszeit nur in erster Näherung Aussagen zu erreichbaren Reaktionszeiten und Determinismus – und damit zu entscheidenden Echtzeiteigenschaften – zulässt. Die Synchronisationsmechanismen moderner Feldbussysteme sind auch außerhalb der Antriebskommunikation sinnvoll einzusetzen. Dabei verbessert ein vollständig durchsynchronisiertes Kommunikationssystem zwar den Determinismus, nicht zwingend jedoch die Reaktionszeit. Für optimale Auswahl bzw. Tuning des Bussystems ist in jedem Fall die Analyse der Echtzeitanforderungen wichtig: Je nachdem, ob in der zu lösenden Applikation Reaktionszeit oder Determinismus im Vordergrund stehen, sind die Parameter unterschiedlich zu wählen.