

Echtzeit-Virtualisierung und Konsolidierung von NC-Steuerungen zur dynamischen Skalierung von Hardwareressourcen

Ch. Susen, O. Petrovic, W. Herfs, Ch. Brecher, WZL, RWTH Aachen

1. Motivation

Die produzierende Industrie sieht sich zunehmendem Druck ausgesetzt, individualisierte und qualitativ hochwertige Produkte in immer kürzeren Entwicklungs- und Produktionszyklen zu fertigen, und das unter strengen Kosten und Nachhaltigkeitsvorgaben [1]. Um die hieraus resultierenden Schwankungen in Produktivitäts- und Qualitätsanforderungen zu kompensieren, rücken hochflexible und rechenintensive Steuerungsalgorithmen in den Fokus. Klassische NC-Steuerungen stoßen dabei jedoch an ihre Grenzen, da sie als monolithische Systeme nur bedingt skalierbar sind [2; 3]. Entweder muss unnötig leistungsfähige Hardware vorgehalten und Steuerungssysteme müssen grundsätzlich überdimensioniert werden, oder das Potenzial innovativer Steuerungsalgorithmen bleibt gänzlich ungenutzt [4].

Ein vielversprechender Lösungsansatz ist die Trennung von Steuerungssoftware und Steuerungshardware durch Verwendung einer Virtualisierungsschicht. Dieses etablierte Prinzip aus der Informationstechnologie ermöglicht es, Rechenleistung bedarfsgerecht zu skalieren. Durch die Konsolidierung mehrerer virtualisierter NC-Steuerungen in einer Edge-Cloud lassen sich außerdem finanzielle Skaleneffekte ausnutzen, während weiter in Echtzeit mit der Feldebene kommuniziert werden kann [5; 6]. Neben der dynamischen Skalierung von Hardwareressourcen profitieren Unternehmen von vereinfachter Rekonfiguration, erhöhter Verfügbarkeit durch erleichterte Backups oder Wiederherstellung und eine vereinfachte Durchführung von Updates der Steuerungsalgorithmen oder Sicherheitsfunktionen [7; 8].

Vor diesem Hintergrund wird in diesem Beitrag ein Konzept vorgestellt, das die Echtzeit Virtualisierung und Konsolidierung von NC-Steuerungen auf IT-Hardware in einer privaten Edge-Cloud am Produktionsstandort ermöglicht. Im Gegensatz zu bestehenden Forschungsarbeiten kommt dabei keine spezialisierte Hardware zum Einsatz; auch die Netzwerkebene wird konsequent mit Standardkomponenten realisiert, um bestehende Infrastruktur in Unternehmen weiterhin nutzen zu können. Das Ziel ist es dabei, eine skalierbare und wirtschaftliche Lösung zu schaffen, die die Nutzung neuartiger und ressourcenintensiver Steuerungsalgorithmen unterstützt.

Zunächst wird in Abschnitt 2 der aktuelle Stand der Forschung im Bereich der Echtzeit Virtualisierung und Konsolidierung von NC-Steuerungen kurz dargelegt. Daraufhin wird in Abschnitt 3 das Lösungskonzept präsentiert. Die Implementierung und die Ergebnisse der Evaluation werden in Abschnitt 4 diskutiert. Zuletzt werden die Forschungsergebnisse in Abschnitt 5 zusammengefasst und es erfolgt ein Ausblick.

2. Stand der Forschung

Die Virtualisierung und Konsolidierung industrieller Steuerungen sind ein zunehmend erforschtes Thema mit dem Ziel, flexiblere und skalierbare Steuerungssysteme zu schaffen [9]. Es wurden bereits diverse Ansätze entwickelt, die sich hinsichtlich der Architektur, der verwendeten Virtualisierungstechnologie und der Netzwerkintegration teils deutlich unterscheiden. Während frühe Arbeiten eine vollständige Verlagerung der Steuerungsfunktionen in die Cloud untersuchen, zeigen neuere Arbeiten, dass eine Edge-Cloud-Architektur vorteilhaft ist, um die Echtzeitanforderungen industrieller Anwendungen zu erfüllen. Die erreichbaren Produktivitätsvorteile, die durch die

Konsolidierung von Steuerungen mittels Virtualisierung in der Produktionstechnik erzielt werden können, werden beispielsweise von FISCHER ET AL. beschrieben [10].

Einige der ersten Forschungsarbeiten im genannten Bereich untersuchen die Möglichkeit, Steuerungen vollständig in eine entfernte Cloud auszulagern. So präsentieren SCHLECHTENDAHL ET AL. eine cloudbasierte SPS zur Steuerung eines Handarbeitsplatzes [7]. In Anlehnung an diese Arbeit entwickeln VERL ET AL. und SCHLECHTENDAHL ET AL. ein Konzept für Machine Control as a Service (MCaaS) für Fräsmaschinen, wobei die Steuerungslogik in Cloud-Dienste ausgelagert wird [4; 11; 12]. Die Kommunikation erfolgt dabei durch Wide Area Networks (WANs) mittels TCP und HTTP, wobei Mechanismen zur adaptiven Anpassung der Bandbreite erforderlich sind [12]. Diese frühen Arbeiten zeigen aufgrund der WAN-Kommunikation jedoch Einschränkungen hinsichtlich der Netzwerklatenz, des Jitters und der Sicherheit und machen eine zuverlässige Echtzeit-Steuerung aus der Cloud nicht uneingeschränkt möglich. KRETSCHMER ET AL. betrachten daher alternative Architekturen mit einer zwischengeschalteten Kommunikationserweiterung zur Reduktion des Jitters [13]. Dabei bleibt dennoch die Herausforderung bestehen, dass es selbst bei reiner LAN-Kommunikation zu hoher Netzwerkvariabilität kommt.

Neuere Forschungsansätze fokussieren sich auf die Virtualisierung und Konsolidierung von Steuerungen in einer Edge-Cloud am Produktionsstandort, um Latenz und Jitter bei der Netzwerkkommunikation zwischen virtualisierten Steuerungen und Feldgeräten zu minimieren [14; 15]. So erforschen GUNDALL ET AL. eine Architektur für virtualisierte SPS mit Container-Virtualisierung und Ethernet Time-Sensitive Networking (TSN) [16; 17]. Der Fokus liegt dabei auf der flexiblen Bereitstellung von Rechenleistung und der Resilienz des Systems. Es werden jedoch keine umfassenden Tests des Systems und der Netzwerkintegration unter Last durchgeführt. PEREZ ET AL. befassen sich ebenfalls mit virtualisierten SPS auf Commercial Off-The-Shelf (COTS)-Servern und verwenden dabei herkömmliche Netzwerktechnik aus der IT. Die Validierung beschränkt sich primär auf die Messung der Antwortzeit der Steuerungen an I/O Terminals [18].

Die Verwendung von Container-Virtualisierung für industrielle Steuerungen wird auch von WALKER ET AL. und TASCI ET AL. untersucht. Das Hauptaugenmerk sind dabei insbesondere Microservice-basierte Architekturen [19; 20]. Diese Ansätze ermöglichen eine hohe Modularität. Die Nutzung bestehender Steuerungssoftware, die nicht auf Linux basiert, ist jedoch bei der Verwendung von Container-Virtualisierung nicht ohne Anpassungen der Software möglich. NEUBAUER ET AL. verfolgen mit Software-Defined Manufacturing (SDM) eine weitergehende Vision einer selbstorganisierten Fertigung, bei der NC-Steuerungen über eine Edge-Cloud flexibel orchestriert werden [8]. Die Virtualisierung der Netzwerkanbindungen der Steuerungen wird mit VLAN-Interfaces auf dem Host umgesetzt. Die Verbindung zwischen Steuerungen und Feldgeräten erfolgt mithilfe eines TSN-fähigen Netzwerks.

Der vorliegende Beitrag unterscheidet sich hinsichtlich der Herangehensweise in mehreren wesentlichen Punkten von diesen bestehenden Forschungsansätzen:

Harte Echtzeitanforderungen: Während sich einige Arbeiten auf SPS oder Motion-Control-Anwendungen mit weniger strikten Zykluszeiten konzentrieren, adressiert das hier vorgestellte Konzept die Virtualisierung von NC-Steuerungen mit typischen Zykluszeiten des Lagereglers zwischen 0,5 ms und 2,5 ms und harten Echtzeitanforderungen [21; 22]. Daher ist ein deterministisches Verhalten der Virtualisierungslösung und des physischen sowie virtuellen Netzwerks essenziell.

Hypervisor-basierte Virtualisierung: Im Gegensatz zu Arbeiten, die auf Container-Virtualisierung aufbauen, wird hier Hypervisor-Technologie verwendet, um eine uneingeschränkte Nutzung

bestehender Steuerungssoftware zu ermöglichen. Container-basierte Ansätze würden bei Steuerungssystemen, die nicht auf Linux basieren, eine Anpassung der Software erfordern und bieten im Allgemeinen eine weniger starke Isolation zwischen den Steuerungen.

Verzicht auf spezialisierte Netzwerkhardware: Während in einigen Arbeiten TSN-fähige Netzwerkhardware verwendet wird, wird hier untersucht, inwiefern sich gewöhnliche Ethernet-Netzwerke mit Quality of Service (QoS)-Mechanismen zur Erfüllung der Echtzeitanforderungen der NC-Steuerungen eignen. Dies ermöglicht eine Weiterverwendung bestehender Infrastruktur in Unternehmen.

Konsolidierung mehrerer Steuerungen: Bestehende industrielle Lösungen zur Echtzeit-Virtualisierung, die z.B. in der Robotik zur Anwendung kommen, konzentrieren sich auf die Virtualisierung einer einzelnen Steuerungsanwendung. Dieser Beitrag untersucht hingegen die parallele Ausführung der echtzeitkritischen Steuerungen mehrerer Maschinen auf einer gemeinsamen Hardwareplattform.

Evaluation der Netzwerkkommunikation: Während sich bestehende Arbeiten auf die Messung von Scheduling-Latenzen innerhalb der Steuerung beschränken oder lediglich die Antwortzeit zur Validierung der Netzwerkkommunikation heranziehen, erfolgt hier eine umfassende Evaluation der gesamten Architektur.

Zusammenfassend zeigt die bestehende Forschung, dass die Virtualisierung und Konsolidierung industrieller Steuerungen technisch grundsätzlich möglich sind. Hinsichtlich der Virtualisierung und Konsolidierung von NC-Steuerungen besteht jedoch unter der Prämisse der Nutzung bestehender Steuerungssoftware und Netzwerkinfrastruktur in produzierenden Unternehmen weiterer Forschungsbedarf, der in diesem Beitrag adressiert werden soll.

3. Konzept

Das hier vorgestellte Konzept sieht vor, NC-Steuerungen in einer Edge-Cloud am Produktionsstandort auf Standard-IT-Hardware zu virtualisieren und mehrere Steuerungen parallel auf einer gemeinsamen Plattform zu konsolidieren. Abbildung 1 zeigt schematisch das Konzept für die Struktur des gesamten neuartigen Steuerungssystems. Die Steuerungssoftware wird als virtualisierte NC-Steuerung auf einer Hypervisor-basierten Plattform ausgeführt. Außerdem können parallel nicht-echtzeitkritische Anwendungen, wie CAD- oder CAM-Systeme in der Edge-Cloud betrieben werden. Ergänzend laufen auf den Edge-Servern Orchestrierungs- und Überwachungsanwendungen zur Verwaltung der Steuerungen und zur Überwachung der Systemauslastung. Als Hypervisor kommt quelloffene Software aus der IT zum Einsatz, die so modifiziert und konfiguriert wird, dass die Echtzeitanforderungen der NC-Steuerungen erfüllt werden können. In den Schaltschränken der Maschinen verbleiben lediglich I/Os und Umrichter für die Sensoren, Aktoren und Antriebe der Maschinen. Diese Feldgeräte sind unter Verwendung optionaler Gateways zur Protokollwandlung mit dem bestehenden Netzwerk des Produktionsunternehmens verbunden. Über Wireless Access Points erhalten Thin Clients und Mobilgeräte Zugang zu diesem Netzwerk, um kostengünstige und flexible Human Machine Interfaces (HMIs) bereitzustellen. Das Unternehmensnetzwerk besteht neben diesen Wireless Access Points aus einer Reihe von Ethernet-Switches. Demnach wird hier die vereinfachende Annahme getroffen, dass das Unternehmensnetzwerk komplett auf Layer 2 arbeitet. Eine Adaption des Konzepts auf Netzwerke in größeren Unternehmen, die Routing auf Layer 3 nutzen, ist jedoch grundsätzlich realisierbar, da auch auf Layer 3 entsprechende QoS-Mechanismen existieren.

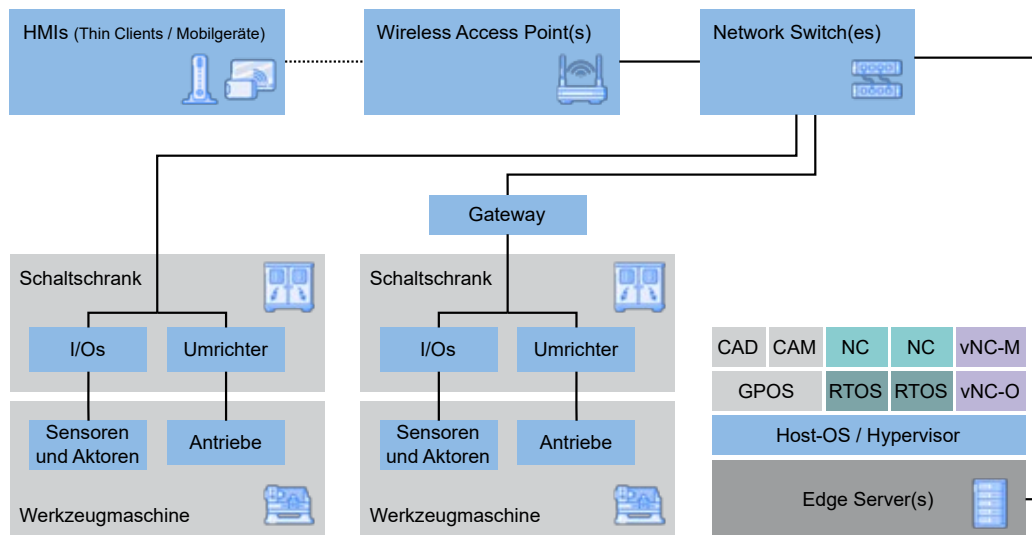


Abbildung 1: Schematische Darstellung des Konzepts für das Gesamtsystem zur Echtzeit-Virtualisierung und Konsolidierung von NC-Steuerungen

Das Konzept basiert konsequent auf Standardkomponenten aus der IT für Server und Netzwerk. Die Kommunikation zwischen den virtualisierten Steuerungen und den Maschinen erfolgt über Ethernet, wobei QoS-Mechanismen zur Priorisierung der übertragenen Steuerungsdaten eingesetzt werden. Die Trennung des Datenverkehrs unterschiedlicher Steuerungen erfolgt mithilfe von VLANs. Die Steuerungssoftware muss für den Betrieb in den virtuellen Maschinen nicht angepasst werden, sodass bestehende Lösungen der Maschinenhersteller wiederverwendet werden können.

4. Implementierung und Evaluation

Zur Validierung des Konzepts wurde eine prototypische Implementierung durchgeführt. Als Edge-Server dient ein Shuttle XPC Cube Barebone SW580R8, ausgestattet mit einem Intel Core i9-10900K Prozessor (10 Kerne, 3,7 GHz), 128 GB RAM und einer Intel X540 Netzwerkkarte mit SR-IOV-Unterstützung. Als Hypervisor kommen QEMU und KVM auf einem Linux-Hostsystem mit PREEMPT_RT-Patch (Version 6.1.38-rt-12) zum Einsatz, wobei CPU-Kerne für die Steuerungen isoliert werden und den Threads der virtuellen CPUs Echtzeitprioritäten zugewiesen werden. Die Netzwerkkarte stellt mithilfe von SR-IOV Virtual-Functions (VFs) für jede Steuerung eigene logische Schnittstellen bereit, die per PCIe-Passthrough direkt den virtuellen Maschinen (VMs) zugewiesen werden. VLANs und QoS Mechanismen nach IEEE 802.1Q und IEEE 802.1p werden genutzt, um Steuerungsdatenverkehr zu priorisieren und zu isolieren. Das Unternehmensnetzwerk wird mithilfe von drei MikroTik CRS310 Switches modelliert. Die Verbindungen zwischen den Switches und zwischen Server und Switches unterstützen eine Datenrate von 10 Gbits⁻¹. Als Feldbus wird für die prototypische Implementierung EtherCAT verwendet. Die Ports der Switches, an die die Feldgeräte angeschlossen werden, unterstützen theoretisch eine Datenrate von 2,5 Gbits⁻¹, welche aber mit den Feldgeräten nicht ausgeschöpft werden kann. Die Übertragung auf andere Feldbussysteme erfordert, wie oben erwähnt, ggf. Gateways zur Wandlung auf standardkonforme Ethernet-Frames.

Auch in den VMs kommt Linux mit PREEMPT_RT-Patch zum Einsatz. Die Konfiguration des Kernels weicht leicht von der auf dem Host ab. Detaillierte Informationen zu den Kernelkonfigurationen werden hier im Sinne der Komplexität und Länge bewusst nicht im Detail erläutert. Die Konfiguration des Gesamtsystems folgt etablierten Vorgehensweisen zur Echtzeitoptimierung von Linux und KVM. Für weitere technische Details sei diesbezüglich auf die Vorarbeit in [23] verwiesen. Alle virtualisierten Steuerungen nutzen den IgH EtherCAT-Master zur Kommunikation mit ihren Feldgeräten. Der Treiber für die VFs der Netzwerkkarte wurde für die

Verwendung mit diesem EtherCAT-Master modifiziert, sodass die EtherCAT-Frames der Steuerungen automatisch in die korrekten hardwareseitigen Warteschlangen zur Priorisierung geschrieben werden können. Für Messungen des Echtzeitverhaltens wird der EtherCAT-Master von selbstentwickelten C++-Applikationen direkt verwendet. Zur Evaluation der virtualisierten NC-Steuerung wird zusätzlich LinuxCNC in einer VM installiert und über einen Glue Layer mit dem EtherCAT-Master verbunden.

Als Demonstrationsmaschine wird ein 3D-Drucker verwendet. Dieser 3D-Drucker wurde so modifiziert, dass sich die Antriebe über Beckhoff EL7041 EtherCAT-Klemmen für Schrittmotoren regeln lassen. Dabei übernimmt die virtualisierte NC-Steuerung die Positionsregelung, während höherfrequente Regelkreise auf den Klemmen laufen, wie es auch bei einem Servomotor und zugehörigen Umrichtern in der Praxis üblich ist. Die Klemmen unterstützen Distributed Clocks zur Synchronisation untereinander und mit dem Master und gewährleisten so eine isochrone Regelung trotz geringfügigen Jitters, der potenziell im Netzwerk auftritt.

Um auch den Aspekt der Konsolidierung mehrerer Steuerungen abzubilden und die Echtzeitfähigkeit des Gesamtsystems messtechnisch untersuchen zu können, werden zusätzlich eigens entwickelte EtherCAT-Slaves auf Basis von Infineon XMC4800 Entwicklungsboards mit EtherCAT-Slave Controller (ESC) verwendet. Diese Slaves ermöglichen die Ermittlung unterschiedlicher Indikatoren für das Echtzeitverhalten des Gesamtsystems, nämlich die Erfassung der Antwortzeit auf externe Ereignisse, die Bestimmung der Rahmenverzögerung (Frame Delay) und die Messung der Genauigkeit der Uhrensynchronisation (Sync0-Offset). Die Slaves messen dabei zeitkritische Abläufe direkt in Hardware und übertragen die Ergebnisse per TCP an eine Anwendung zur Auswertung. Mit Frame Delay ist hier die Zeit gemeint, die zwischen dem Versand eines EtherCAT-Frames durch eine virtualisierte Steuerung und der Ankunft im Feld beim Slave vergeht. Der Sync0-Offset bezeichnet die Differenz zwischen dem Auftreten des Sync0-Interrupts beim Master und beim Slave und gibt Aufschluss über die Genauigkeit der Uhrensynchronisation. Zur Erfassung der Rahmenverzögerung und der Genauigkeit der Uhrensynchronisation wird hier ein zweikanaliges Messverfahren verwendet, bei dem der Versandzeitpunkt eines EtherCAT-Frames von der Steuerung über eine RS232 Schnittstelle signalisiert wird. Der Slave vergleicht diesen Referenzzeitpunkt mit dem tatsächlichen Empfang des Frames bzw. dem Auftreten des Sync0-Interrupts, um präzise Latenz- und Synchronisationswerte zu ermitteln. Die Antwortzeit wird hingegen als Zeitspanne zwischen einem extern generierten Ereignis (PWM-Signal) und der Reaktion der Steuerung gemessen. Alle Messungen erfolgen mit einer Zeitauflösung im einstelligen Mikrosekundenbereich und ermöglichen so eine detaillierte und hochaufgelöste Analyse der Echtzeitfähigkeit des gesamten virtualisierten Steuerungssystems.

Insgesamt laufen auf dem Edge-Server vier virtuelle Maschinen. Davon lastet eine das physische und virtuelle Netzwerk sowie den Server mithilfe der Anwendungen *iperf3* und *stress-ng* permanent aus. Die Netzwerklast mit *iperf3* wird bidirektional im Zusammenspiel mit einem weiteren Server erzeugt und sorgt dafür, dass die Warteschlangen niedriger Priorität aller Switches durchgehend gefüllt sind. Die anderen drei VMs führen echtzeitkritische Steuerungsaufgaben aus. Dabei führen zwei Steuerungen mit je einem EtherCAT-Slave eine Antwortzeitmessung durch. Die andere Steuerung ist an zwei EtherCAT-Slaves angeschlossen. Davon ermittelt einer die Rahmenverzögerung und der andere die Genauigkeit der Uhrensynchronisation. Alle Messungen erfolgen bei einer eingestellten Zykluszeit des Masters von 1 ms. Der beschriebene Messaufbau ist schematisch in Abbildung 2 dargestellt.

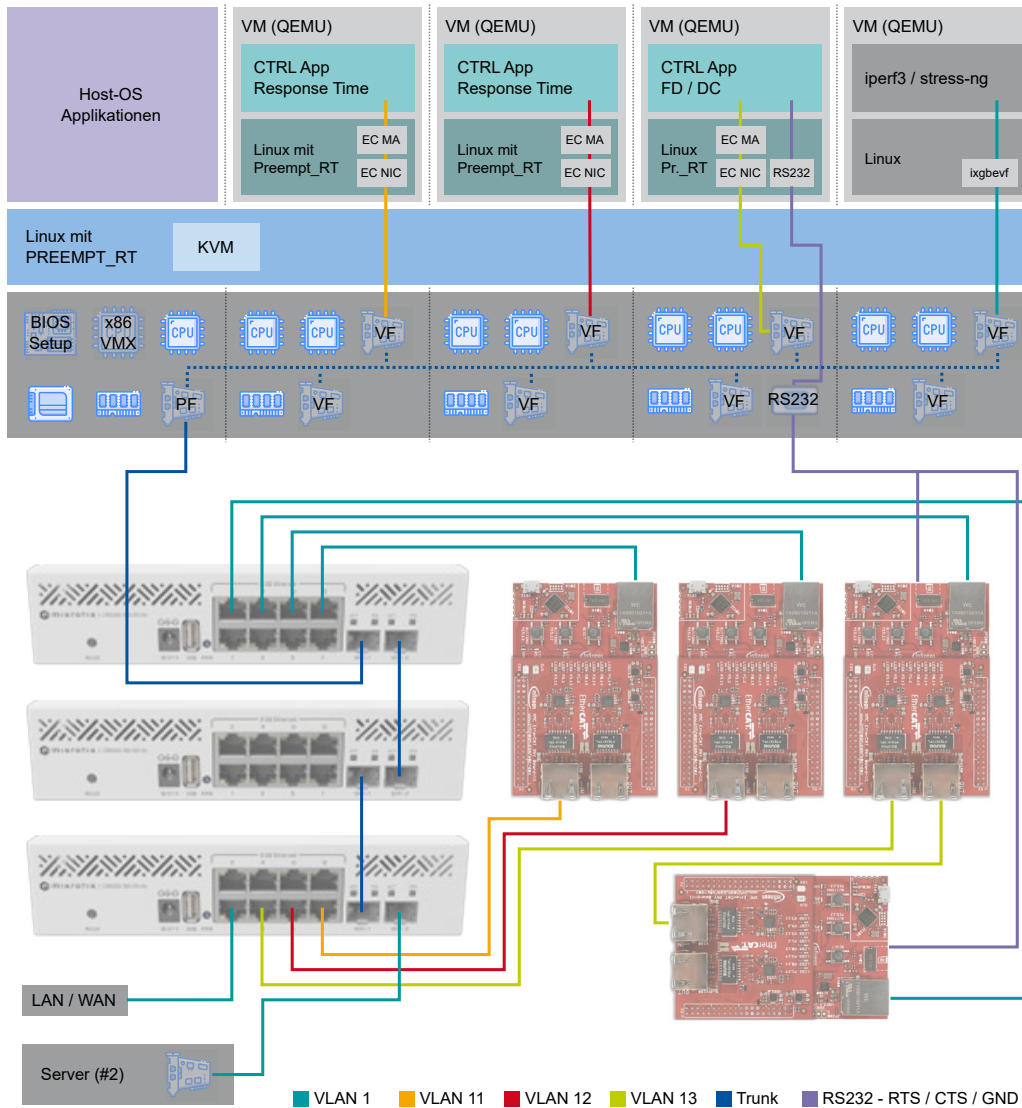


Abbildung 2: Schematische Darstellung des Messaufbaus zur Validierung der Echtzeitfähigkeit des Gesamtsystems

Parallel zu den Messungen mithilfe der EtherCAT-Slaves wird die Scheduling-Latenz in den VMs und auf dem Host kontinuierlich mit *cyclictest* erfasst. Alle Messungen werden für eine Dauer von 25 h durchgeführt. Die Ergebnisse der Messung der Scheduling Latenz sind in Form von Histogrammen in Abbildung 3 visualisiert. Dabei liegt der Durchschnitt der Latenz auf dem Host bei unter 2 μ s und in den VMs bei knapp über 4 μ s. Das in den über 24 h gemessene Maximum beträgt auf dem Host knapp 34 μ s und über alle VMs hinweg knapp 78 μ s. Die maximalen und durchschnittlichen Scheduling Latenzen, die in den Steuerungs-VMs auftreten, sind für die hier angestrebte Anwendung der Lageregelung mit einem Takt von 0,5 ms bis 2,5 ms akzeptabel. Zu berücksichtigen ist jedoch, dass die Scheduling-Latenzen in den VMs um ca. 40 μ s höher sind als auf dem Host.

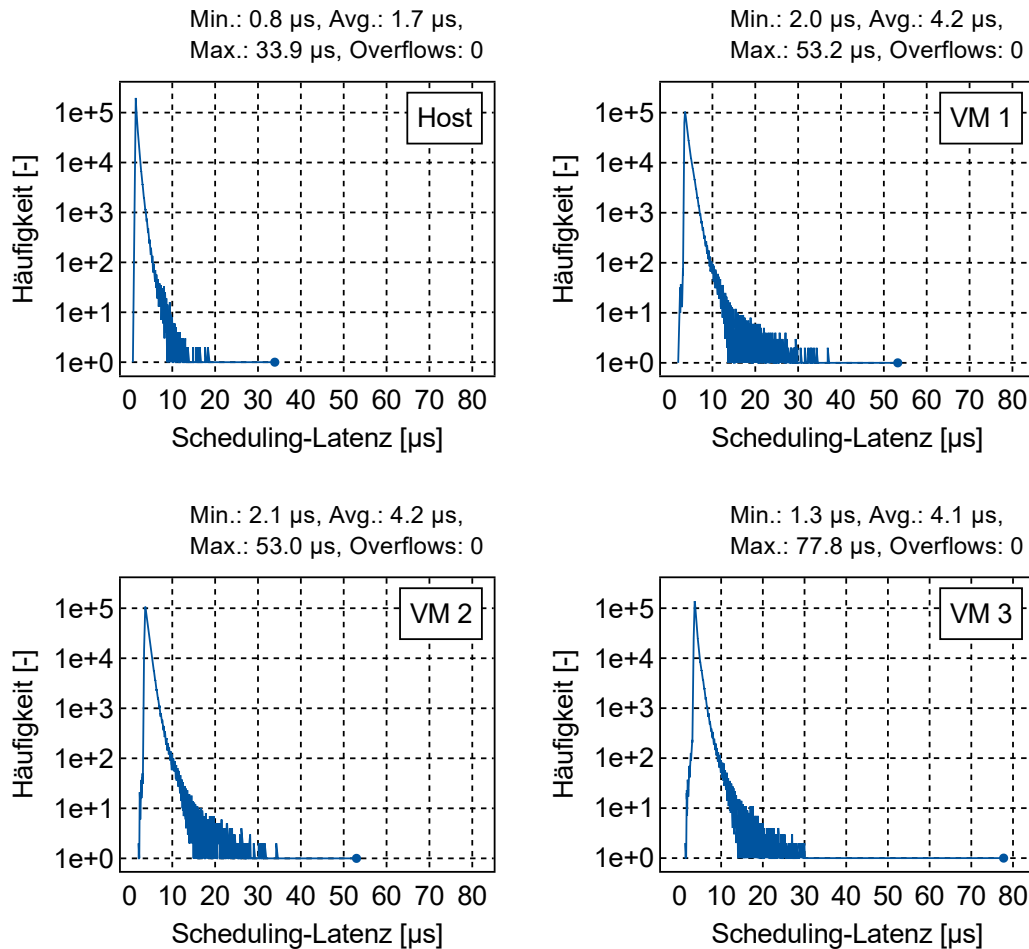


Abbildung 3: Mit cyclictest ermittelte Scheduling-Latenz in allen drei Steuerungs-VMs und auf dem Host

Die Ergebnisse der Messungen der Antwortzeit sind für die beiden VMs und Slaves in Abbildung 4 dargestellt. Dabei zeigen die Plots die Antwortzeit als prozentuale Verteilungsfunktion. Die drei unterschiedlichen Kurven stellen das Minimum, Maximum und den Durchschnitt aus jeweils 200 Messungen dar. Diese Aggregation von je 200 Messwerten wurde vorgenommen, um den Mikrocontroller der Slaves nicht mit dem Versand je eines Pakets pro Antwortzeitmessung zu überlasten. Die Antwortzeit wird mithilfe eines PWM-Signals mit einer Frequenz von 400 Hz (abwechselnd 2,5 ms hohes und 2,5 ms niedriges Spannungspegel) ermittelt. Gemessen wird die Verzögerung zwischen den steigenden Flanken dieses Signals vom Slave und den steigenden Flanken der Antworten des Masters. Theoretisch sollten die Minima und Maxima der Antwortzeit bei einer und zwei Zykluszeiten liegen. Im Bestfall durchläuft der Frame den Slave unmittelbar nach Auftreten des externen Ereignisses. Der Master liest die Information dann sofort zu Beginn des nächsten Zyklus ein und reagiert mit der entsprechenden Antwort im nächsten Frame. Im schlechtesten Fall hat der Frame den Slave bereits unmittelbar vor Auftreten des Ereignisses durchlaufen. Das vom Slave gesetzte Bit, also die steigende Flanke, kommt dann erst im nächsten Zyklus beim Master an. Demnach erfolgt die Reaktion des Masters erst im übernächsten Zyklus. Sind das externe Ereignis und die Steuerung nicht synchronisiert, treten auch alle Werte zwischen dem theoretischen Minimum und Maximum auf. Es ergibt sich dann bei Betrachtung der Antwortzeit über der Zeit ein Sägezahnmuster, das zwischen einer und zwei Zykluszeiten der Steuerung oszilliert. Jegliche Latenzen im Gesamtsystem aus Rechenhardware, Netzwerkhardware und Software können diese theoretischen Werte sowohl verringern als auch vergrößern.

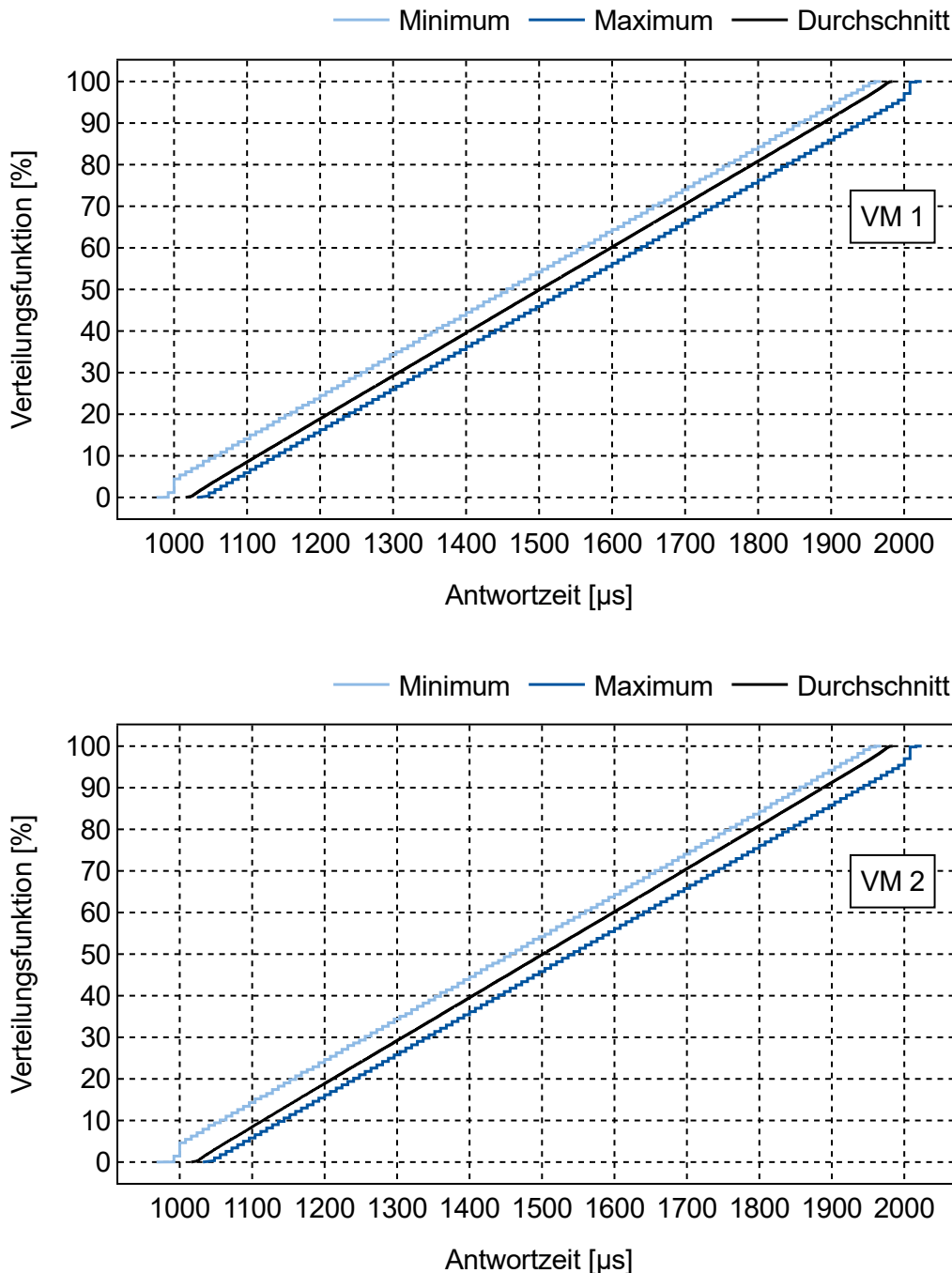


Abbildung 4: Ergebnisse der Messung der Antwortzeit mit den ersten beiden VMs

Das gemessene Minimum der Reaktionszeit in den gesamten 25 h liegt für die erste Steuerungs-VM bei 976 μs und für die zweite Steuerungs-VM bei 968 μs . Die Maxima liegen für beide Steuerungen bei 2024 μs . Demnach weichen die gemessenen Werte nach unten maximal 32 μs und nach oben maximal 24 μs von den theoretisch erwarteten Werten ab. Während der gesamten Messung der Antwortzeit der beiden Steuerungen blieb keine einzige Reaktion einer der beiden Steuerungen aus.

Die Ergebnisse der Messung der Rahmenverzögerung sind in Abbildung 5 dargestellt. Hierbei veranschaulicht der Plot in der Abbildung die Rahmenverzögerung über der Zeit. Die drei unterschiedlichen Kurven stellen das Minimum, Maximum und den Durchschnitt aus jeweils 1000 Messungen, also einer Sekunde, dar. Das Minimum des Frame Delays in den gesamten 25 h liegt bei 4 μs . Das Maximum beträgt 76 μs . Weiter beträgt der Durchschnittswert für die meisten Datenpunkte zwischen 32 μs und 36 μs . Im Rahmen weiterer Messungen zur Validierung hat sich gezeigt, dass

der maßgebliche Teil der Rahmenverzögerung durch Effekte innerhalb des Servers verursacht wird, wie z.B. die Auslastung der PCIe Bandbreite. Die Switches, die für die kritischen Verbindungen untereinander eine Datenrate von 10 Gbits^{-1} unterstützen, tragen nur einen geringen Teil zur Rahmenverzögerung bei. Für Anwendungsfälle, bei denen eine eher kleine Zykluszeit des Lagereglers von $500 \mu\text{s}$ oder weniger erforderlich ist, kann diese zusätzliche Verzögerung eventuell hinderlich sein. Im Allgemeinen ist sie jedoch erst einmal nicht grundsätzlich als beeinträchtigend für die Virtualisierung und Konsolidierung von NC-Steuerungen anzusehen. Darüber hinaus ist es denkbar, die Rahmenverzögerung beispielsweise durch Verwendung eines Servers mit höherer PCIe Bandbreite weiter zu reduzieren.

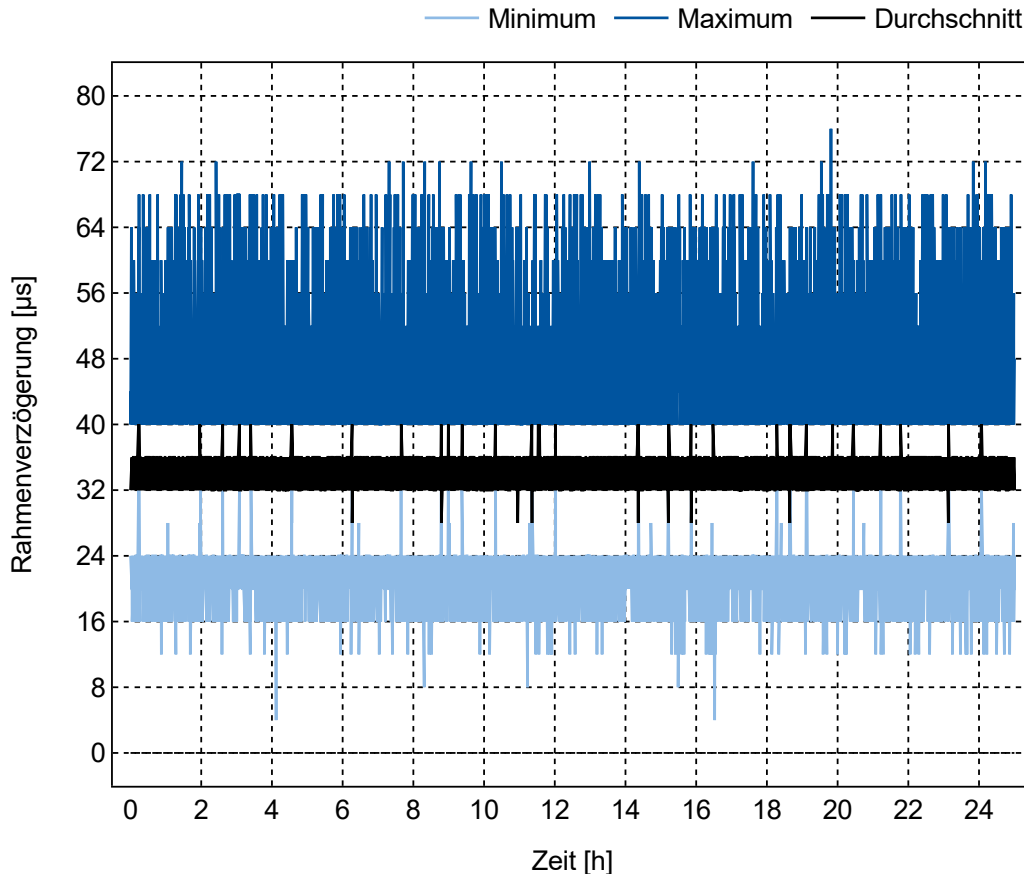


Abbildung 5: Ergebnisse der Messung der Rahmenverzögerung mit der dritten VM

Die Ergebnisse der Messung der Genauigkeit der Uhrensynchronisation sind in Abbildung 6 visualisiert. Hier zeigen die drei Plots Minimum, Maximum und Durchschnitt der Differenz zwischen dem Zeitpunkt des Auftretens des Sync0-Interrupts auf dem Slave und dem Versand des Frames durch den Master aus jeweils 1000 Zyklen. Der Sync0-Shift ist für den Slave auf $570 \mu\text{s}$ eingestellt, sodass eine konstante Linie bei $570 \mu\text{s}$ das ideale Ergebnis in einem System ohne Latenz und Jitter wäre. Auch eine konventionelle IPC basierte Steuerung ohne Virtualisierung und ohne zwischengeschaltetes Netzwerk zeigt nicht dieses ideale Verhalten. Bei Verwendung von EtherCAT sind die Slaves untereinander bis auf wenige Nanosekunden genau synchronisiert. Weil die Sync-Interrupts auf den Slaves eine zeitlich hochpräzise und deterministische Erfassung der Eingänge und Schaltung der Ausgänge ermöglichen, ist es kein Problem, dass der Master und der Zeitpunkt des Versands des EtherCAT-Frames durch diesen einem gewissen Jitter unterliegen. Es können handelsübliche Netzwerkcontroller verwendet werden, und der Master muss den EtherCAT-Frame lediglich ausreichend früh versenden, sodass neue Prozessdaten vollständig in den Speicher der ESCs aller Slaves kopiert werden können, bevor der Sync-Interrupt auftritt. Darüber hinaus muss es der Frame auch zurück zum Master schaffen, bevor dieser im nächsten Zyklus die neuen Daten aus dem

Frame ausliest. Aus diesem Grund müssen die Rahmenverzögerung und der Fehler der Uhrensynchronisation wesentlich kleiner sein als die angestrebte Zykluszeit. In praktischen Tests hat sich eine obere Grenze von 10 % der Zykluszeit als akzeptabel herausgestellt.

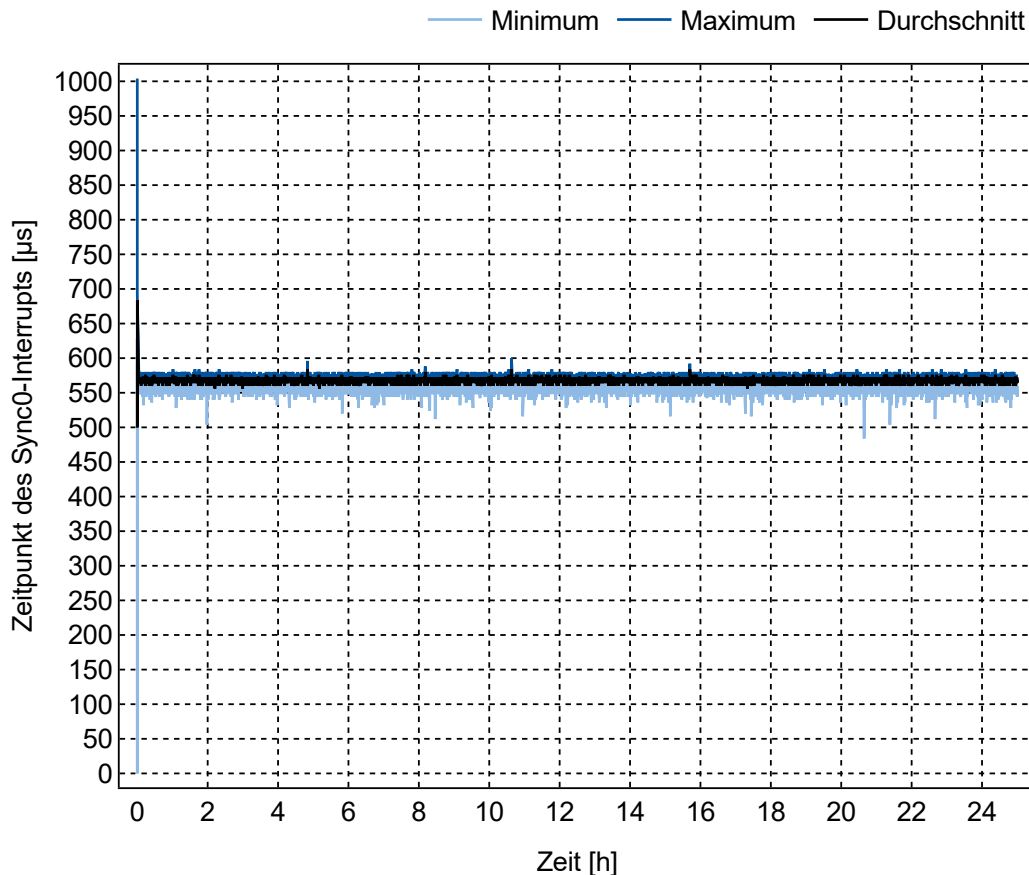


Abbildung 2: Ergebnisse der Messung der Genauigkeit der Uhrensynchronisation mit der dritten VM

Unmittelbar zu Beginn der Messung schwankt der Fehler in der Uhrensynchronisation noch zwischen 0 µs und 1000 µs, also um eine ganze Zykluszeit. Dies liegt darin begründet, dass Master und Slave zu Beginn noch kein gemeinsames Zeitverständnis haben. In der initialen Phase der Synchronisation muss jede Uhr im System mit der Referenzuhr synchronisiert werden. Dieser Vorgang benötigt hier ca. eine Minute. Nach Abschluss dieser Phase bewegt sich der Durchschnitt durchweg nahe dem erwarteten Wert von 570 µs. Der gemessene Minimal- und Maximalwert liegen bei 484 µs und 600 µs, was einer Abweichung vom erwarteten Wert von 86 µs nach unten und 30 µs nach oben entspricht. Diese Werte betragen unter 10 % der Zykluszeit und sind akzeptabel. Während der 25-stündigen Messung kam es kein einziges Mal zu einem Fehler bei der Synchronisation zwischen Master und Slaves.

Neben diesen theoretischen messtechnischen Untersuchungen der Echtzeitfähigkeit wurde auch eine Validierung mithilfe der bereits erwähnten Demonstrationsmaschine durchgeführt. Dazu wurde ein HMI auf Basis eines Tablets realisiert, was einen RDP-Server innerhalb der VM nutzt, um die Bedienoberfläche von LinuxCNC zugänglich zu machen. So kann die Maschine beispielsweise zum Einfahren manuell durch einen Maschinenbediener gesteuert werden. Das Tablet-basierte HMI ist schematisch in Abbildung 7 dargestellt.

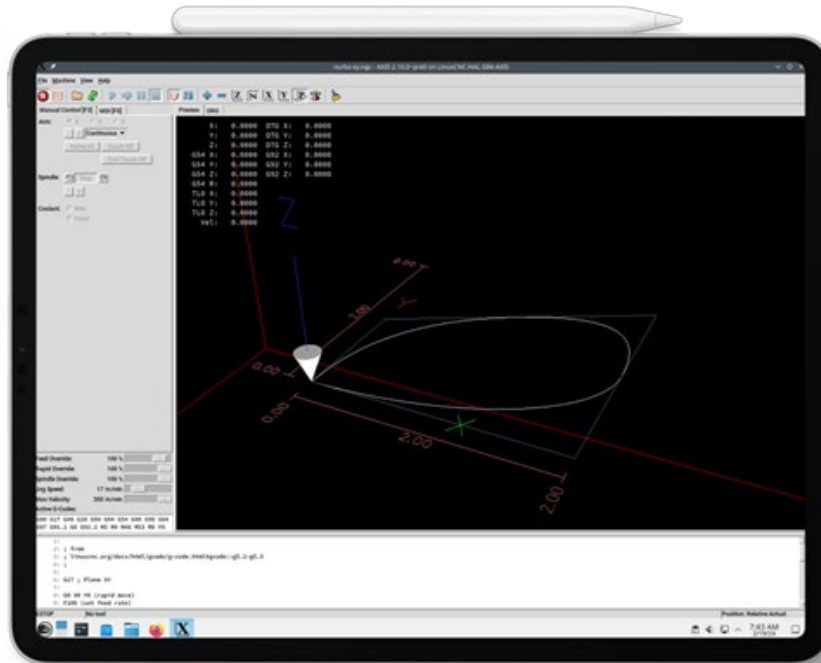


Abbildung 7: Exemplarische Darstellung eines Tablet-basierten HMIs zur Bedienung virtualisierter NC-Steuerungen

Mit diesem HMI wurden zur Validierung unterschiedliche NC-Programme geladen und auf fehlerfreie Ausführung geprüft. Auch dies geschah unter den zuvor beschriebenen Rahmenbedingungen und unter Auslastung des Servers und Netzwerks. Die NC-Steuerung lief dabei parallel zu zwei weiteren Steuerungen. Es wurde außerdem ein 24-stündiger Test durchgeführt, in dem immer wieder ein Kreis in der X-Y-Ebene abgefahren wurde. Während dieses Tests wurde sichergestellt, dass es zu keiner Auslösung des Watchdogs der Slaves kommt und dass keine Verzögerung der Prozessdatenkommunikation auftritt. Darüber hinaus wurde die Phase-Locked Loop (PLL) von LinuxCNC über ein Skript überwacht, um den Jitter im System zu beobachten und zu kontrollieren, dass die Uhrensynchronisation zwischen Steuerung und Slaves ordnungsgemäß funktioniert. Während dieser Versuche konnte kein fehlerhaftes Verhalten beobachtet werden. Durch Deaktivierung von QoS im Netzwerk wurde auch die Gegenprobe angestellt. Ohne QoS kam es nach wenigen Minuten zu einer ausreichend hohen Verzögerung der Kommunikation, die dazu führte, dass der Watchdog der Slaves auslöste. Die Steuerung schaltete die Antriebe in diesem Moment sofort stromlos. Diese praktischen Versuche weisen die Funktionsfähigkeit der virtualisierten NC-Steuerung nach.

5. Zusammenfassung und Ausblick

Mit dem vorgestellten Konzept zur Echtzeit-Virtualisierung und Konsolidierung von NC-Steuerungen auf Standard-IT-Hardware in einer Edge-Cloud konnte gezeigt werden, dass sich mehrere virtualisierte Steuerungen parallel auf einer gemeinsamen Hardwareplattform betreiben lassen, während die Echtzeitanforderungen eingehalten werden. Die messtechnische Evaluation bestätigt, dass Zykluszeiten im Bereich von 1 ms und eine präzise Uhrensynchronisation mit Abweichungen von weniger als 10 % der Zykluszeit realisiert werden können. Auch die Antwortzeiten auf externe Ereignisse sowie die Rahmenverzögerung bleiben im akzeptablen Bereich, selbst bei hoher Last im Netzwerk und auf dem Server. Die durchgängige Priorisierung und Isolierung der Steuerungsdaten im Ethernet-Netzwerk mit QoS-Mechanismen erweist sich als entscheidender Faktor für die Echtzeitfähigkeit. Durch Hypervisor-Virtualisierung wird die Nutzung unveränderter

NC-Software der Maschinenhersteller ermöglicht, was eine Integration in bestehende Unternehmensinfrastrukturen begünstigt.

Ein wesentlicher Beitrag des Konzepts liegt auch in der konsequenten Verwendung von COTS-Hardware auf der Netzwerkebene. Die Evaluation zeigt, dass sich die Echtzeitanforderungen einer NC-Steuerung auch ohne TSN fähige Spezialhardware erfüllen lassen, sofern die Netzwerkkonfiguration mittels VLANs und QoS auf die Priorisierung von Steuerungsdaten ausgerichtet wird. Dies ermöglicht die Weiternutzung bestehender Netzwerkinfrastruktur in produzierenden Unternehmen und reduziert die Hürde für die Einführung virtualisierter Steuerungssysteme erheblich.

Für die weitere Forschung ergeben sich mehrere potenzielle Anknüpfungspunkte. Insbesondere die Integration rechenintensiver Algorithmen aus den Bereichen der künstlichen Intelligenz und datengetriebenen Prozessoptimierung könnte von der flexiblen Bereitstellung der Hardwareressourcen profitieren. Darüber hinaus bietet die Kombination der Hypervisor-basierten Virtualisierung mit Container-Technologie interessante Möglichkeiten zur Reduktion des System-Overheads und zur Steigerung der Modularität. Auch die Übertragung der Konzepte auf mobile Anwendungen wie fahrerlose Transportsysteme und autonome Roboter ist vielversprechend und erfordert weitere Untersuchungen, insbesondere im Hinblick auf drahtlose Kommunikation. Schließlich sind Themen wie funktionale Sicherheit, Redundanzmechanismen und Ausfallsicherheit im Umfeld virtualisierter Steuerungen vertiefend zu betrachten, um den industriellen Einsatz in sicherheitskritischen Anwendungen zu ermöglichen.

Hinweis zur Herkunft der Inhalte

Dieser Beitrag stellt eine Zusammenfassung zentraler Inhalte der Dissertation „*Echtzeit Virtualisierung und Konsolidierung von NC-Steuerungen zur dynamischen Bereitstellung von Hardware für rechenintensive Steuerungsaufgaben*“ von Christoph Susen dar, die im Februar 2025 an der Fakultät für Maschinenwesen der RWTH Aachen University eingereicht wurde.

Danksagung

Das IGF-Vorhaben „*ViRTNC – Echtzeit Virtualisierung von NC-Steuerungen zur dynamischen Bereitstellung von Hardwareressourcen für rechenintensive Steuerungsaufgaben*“, IGF Projekt Nr. 22716 N/2, der Forschungsvereinigung Programmiersprachen für Fertigungseinrichtungen e. V. (FVP) wurde im Rahmen des Programms zur Förderung der industriellen Gemeinschaftsforschung (IGF) vom Bundesministerium für Wirtschaft und Klimaschutz (BMWK) aufgrund eines Beschlusses des Deutschen Bundestages gefördert. Die Autoren dieses Beitrags möchten sich herzlich für die erfahrene finanzielle Unterstützung bedanken.

6. Literaturverzeichnis

- [1] Stark, J.: PLM and Its Environment. In: Stark, J. (Hrsg.): Product Lifecycle Management (Volume 1). (Reihe: Decision Engineering). Cham: Springer International Publishing, 2020, S. 35–70.
- [2] Riedel, O.; Lechler, A.; Verl, A.: Control Architecture for Automation. In: Nof, S. Y. (Hrsg.): Springer Handbook of Automation. (Reihe: Springer Handbooks). Cham: Springer International Publishing, 2023, S. 357–378.

- [3] Tasci, T.; Walker, M.; Lechler, A.; Verl, A.: Towards Modular and Distributed Automation Systems – Leveraging Software Containers for Industrial Control: Automation 2020: VDI Verlag, 2020, S. 915–928.
- [4] Lechler, A.; Schlechtendahl, J.: Steuerung aus der Cloud. In: Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M. ten (Hrsg.): Handbuch Industrie 4.0 Bd.1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, S. 61–74.
- [5] Arogundade, O. R.; Palla, K.: Virtualization Revolution: Transforming Cloud Computing with Scalability and Agility. In: IARJSET. 10. Jg., 2023, Nr. 6.
- [6] Sprenger, M.; Krause, H.-M.; Gezer, V.; Cornel, M.; Grotepass, J.; Demir, C.; Höckner, S.; Brüggemann, C.; Miersch, G.; Ziegenfuss, U.; Ruskowski, M.; Hamm, S.; Bedarf, D.; Runge, T.; Pfeiffer, M.; Althoff, S.: Industrial Edge Cloud - Whitepaper SF-1.1. URL: https://smartfactory.de/wp-content/uploads/2021/11/SF_Whitepaper-Industrial-Edge-Cloud-WEB.pdf [Stand: 12.02.2025].
- [7] Schlechtendahl, J.; Kretschmer, F.; Lechler, A.: I4.0 - Totale Vernetzung von der Klemme bis zur Cloud. In: Industrie 4.0 Management, 2015, S. 8–11.
- [8] Neubauer, M.; Reiff, C.; Walker, M.; Oechsle, S.; Lechler, A.; Verl, A.: Cloud-based evaluation platform for software-defined manufacturing. In: at - Automatisierungstechnik. 71. Jg., 2023, Nr. 5, S. 351–363.
- [9] Susen, C.; Storms, S.; Herfs, W.: Virtualisierung in der Steuerungstechnik/Virtualization in control technology – Virtualization and consolidation of programmable logic controllers. In: wt Werkstattstechnik online. 112. Jg., 2022, 11-12, S. 792–796.
- [10] Fischer, M.; Müller, C.; Kolb, C.: Virtualisierung in der Produktionstechnik: atp magazin, 2020, S. 32–34.
- [11] Verl, A.; Lechler, A.; Wesner, S.; Kirstädter, A.; Schlechtendahl, J.; Schubert, L.; Meier, S.: An Approach for a Cloud-based Machine Tool Control. In: Procedia CIRP. 7. Jg., 2013, S. 682–687.
- [12] Schlechtendahl, J.; Kretschmer, F.; Lechler, A.; Verl, A.: Communication Mechanisms for Cloud based Machine Controls. In: Procedia CIRP. 17. Jg., 2014, S. 830–834.
- [13] Kretschmer, F.; Friedl, S.; Lechler, A.; Verl, A.: Communication extension for cloud-based machine control of simulated robot processes. In: 2016 IEEE International Conference on Industrial Technology (ICIT): IEEE, 2016, S. 54–58.
- [14] Susen, C.; Horn, B. von; Petrovic, O.; Herfs, W.: Network Aspects of Virtualized Industrial Control from the Edge Cloud Enabled by COTS Hardware. In: 2023 7th International Conference on Automation, Control and Robots (ICACR): IEEE, 2023, S. 130–139.
- [15] Susen, C.; Storms, S.; Brecher, C.: Towards numerical control of machine tools from the edge cloud. In: at - Automatisierungstechnik. 71. Jg., 2023, Nr. 5, S. 390–402.
- [16] Gundall, M.; Glas, C.; Schotten, H. D.: Introduction of an Architecture for Flexible Future Process Control Systems as Enabler for Industry 4.0. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA): IEEE, 2020, S. 1047–1050.
- [17] Gundall, M.; Glas, C.; Schotten, H. D.: Feasibility Study on Virtual Process Controllers as Basis for Future Industrial Automation Systems. In: 2021 22nd IEEE International Conference on Industrial Technology (ICIT): IEEE, 2021, S. 1080–1087.

- [18] Perez, J. D.; Walzl, J.; Prenzel, L.; Steinhorst, S.: How Real (Time) Are Virtual PLCs? In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA): IEEE, 2022, S. 1–8.
- [19] Tasci, T.; Melcher, J.; Verl, A.: A Container-based Architecture for Real-Time Control Applications. In: 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC): IEEE, 2018, S. 1–9.
- [20] Walker, M.; Tasci, T.; Lechler, A.; Verl, A.: Analysis of Real-Time Execution Models for Container-Based Control Applications. In: Kiefl, N.; Wulle, F.; Ackermann, C.; Holder, D. (Hrsg.): Advances in Automotive Production Technology – Towards Software-Defined Manufacturing and Resilient Supply Chains. (Reihe: ARENA2036). Cham: Springer International Publishing, 2023, S. 14–24.
- [21] Berns, K.; Köpper, A.; Schürmann, B.: Technische Grundlagen Eingebetteter Systeme. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.
- [22] Rudolf, T.; Brecher, C.: Adaptierbare Parametrierung von Diagnosesystemen durch Verwendung digitaler Antriebssignale in der Prozessüberwachung: Lehrstuhl für Werkzeugmaschinen / Werkzeugmaschinenlabor WZL der RWTH Aachen. Apprimus-Verl. and Zugl.: Aachen, Techn. Hochsch., Diss, 2014.
- [23] Susen, C.; Schoofs, N.; Petrovic, O.; Herfs, W.: Clock Synchronization for Virtualized Numerical Control from the Edge Cloud using EtherCAT and COTS Hardware. In: Procedia CIRP. 130. Jg., 2024, S. 1856–1862.