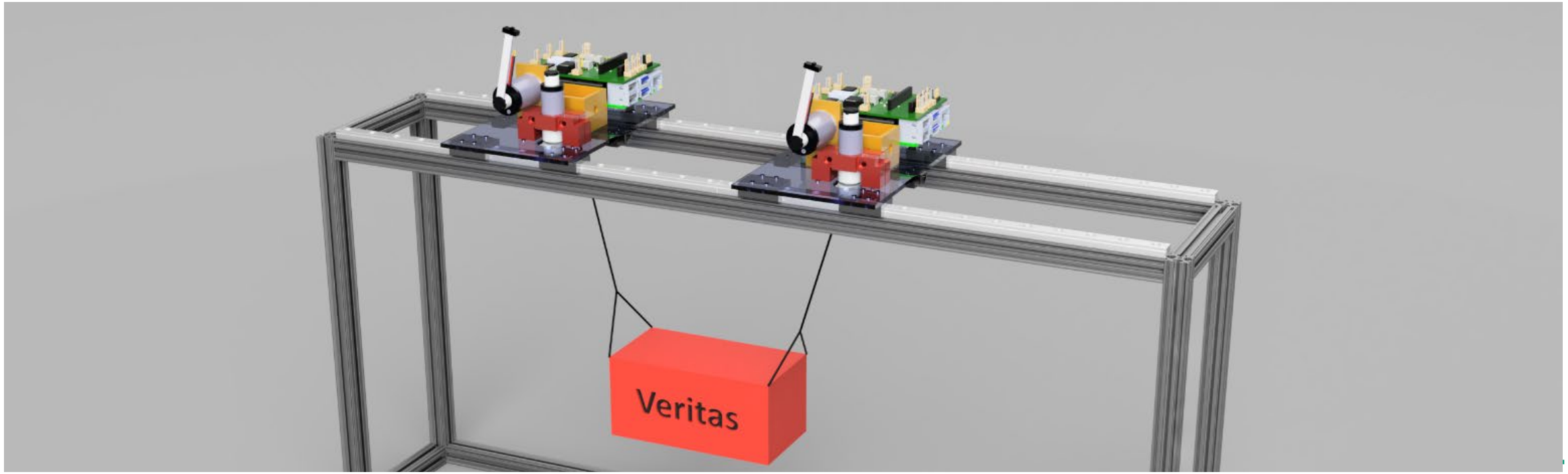
MODELLBASIERTE ENTWICKLUNG VERTEILTER SYSTEME MIT DEM ROBOT OPERATING SYSTEM ROS

Dirk Mayer

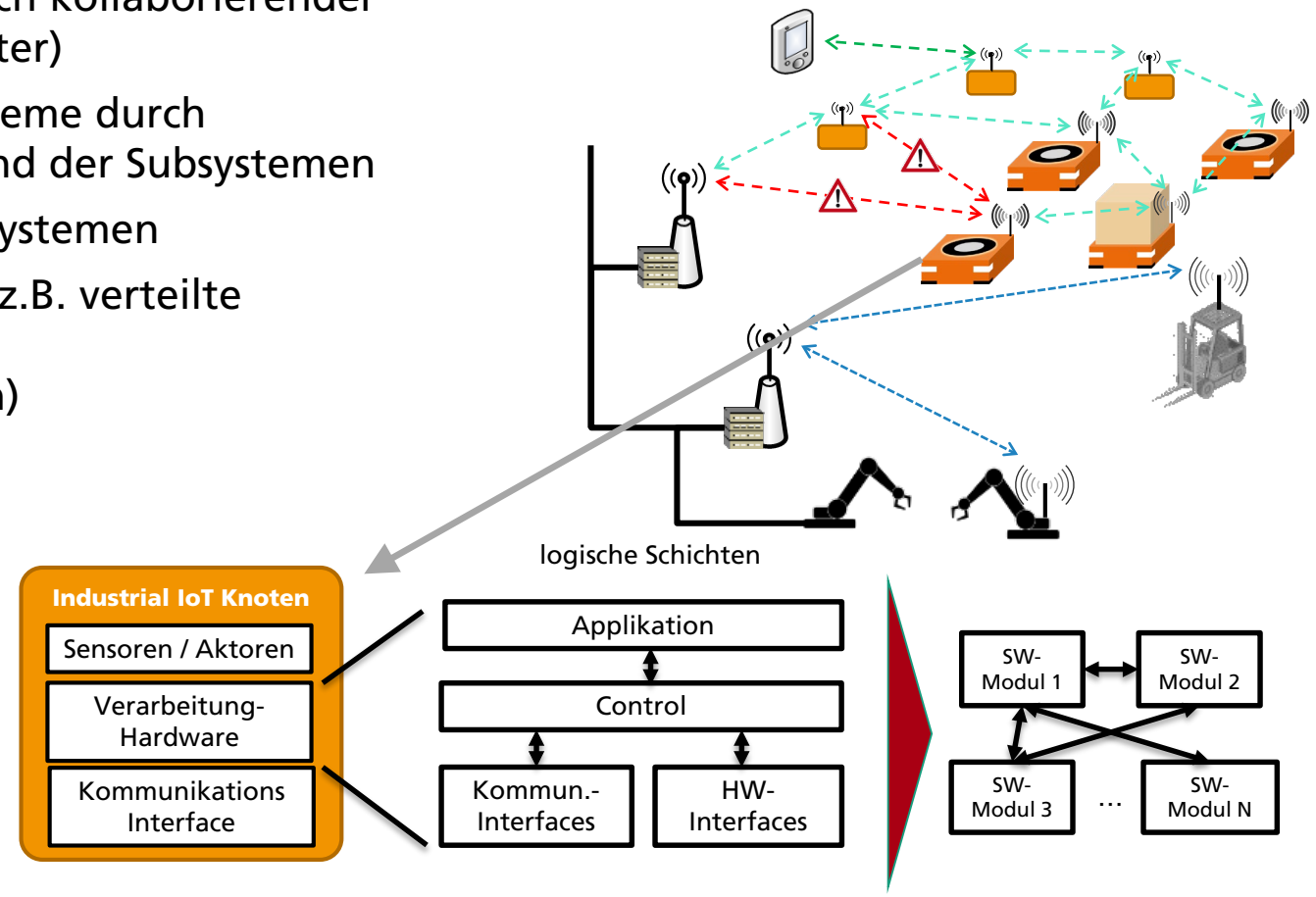


Fraunhofer IIS


Durchgängige Digitalisierung der Produktion
Flexibilisierung der Produktion
Funkvernetzung als Bindeglied
Zunehmende Komplexität der Systeme
Einsatz verteilter smarter Systeme
1. Neue Konzepte für flexible Produktion und smarte Maschinen
2. Dezentralisierung und Verteilung komplexer Aufgaben auf mehrere Subsysteme und dynamische Kollaboration der Subsysteme
3. Dynamische und agile Kollaboration smarter Systeme, selbst optimierend und lernend
4. Temporäre Schwankungen der Performance der Kommunikation

Vision

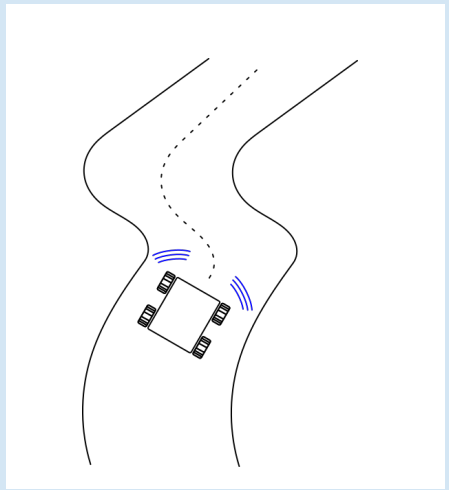
- Flexible Produktionsprozesse mittels dynamisch kollaborierender Subsysteme (z.B. Kollaboration mobiler Roboter)
- Vereinfachung des Managements der Subsysteme durch Selbstorganisation von SW/HW Submodule und der Subsystemen
- Agile Aufgabenverteilung zwischen den Subsystemen
- Co-Management der verteilten Anwendung (z.B. verteilte Regelung) und Kommunikationsinfrastruktur (Koordination Anforderungen vs. Fähigkeiten)



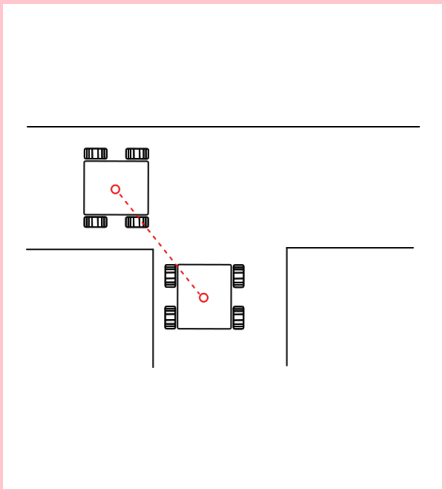
Vorteile

- Flexibilisierung der Produktionsprozesse
- Verringerung des Umrüstaufwands
- Erhöhung der Resilienz und Performance

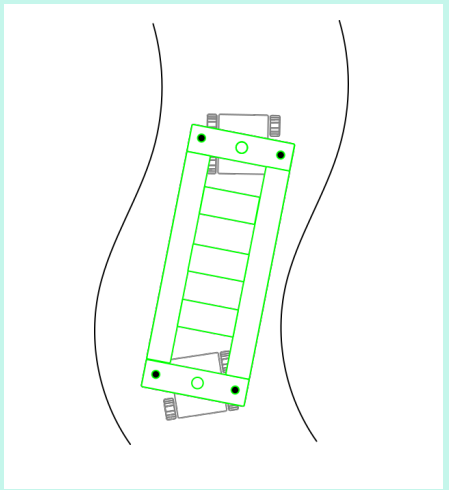
Kopplung cyberphysikalischer Systeme



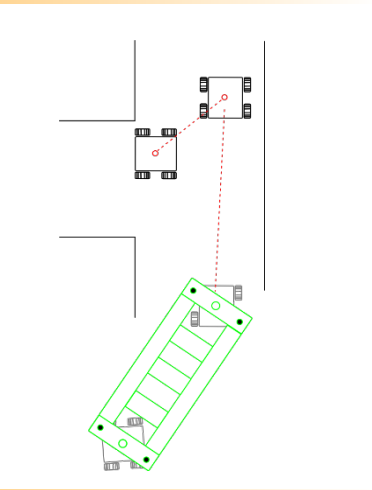
Unabhängig



Kopplung über Informationsaustausch, Kommunikationssystem



Kopplung über physikalische Systeme



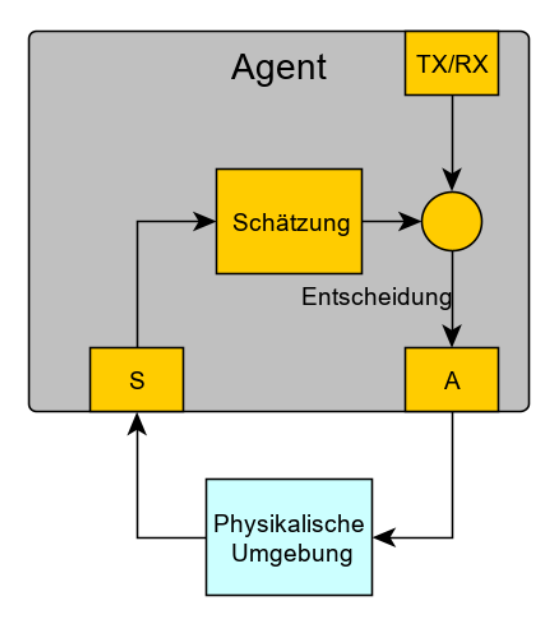
Kombinierte Konfigurationen

- Das System-of-Systems besitzt eine dynamisch veränderliche Zahl von Subsystemen
- Die einzelnen Systeme sind autonom funktional und nicht von einer zentralen Instanz abhängig
- Die Systeme interagieren auf unterschiedlichen Wegen (Informationskanal, physikalische Systeme)
- Die Kopplung durch Kommunikationskanäle führt zu Unsicherheiten (Latenz, Paketverluste)

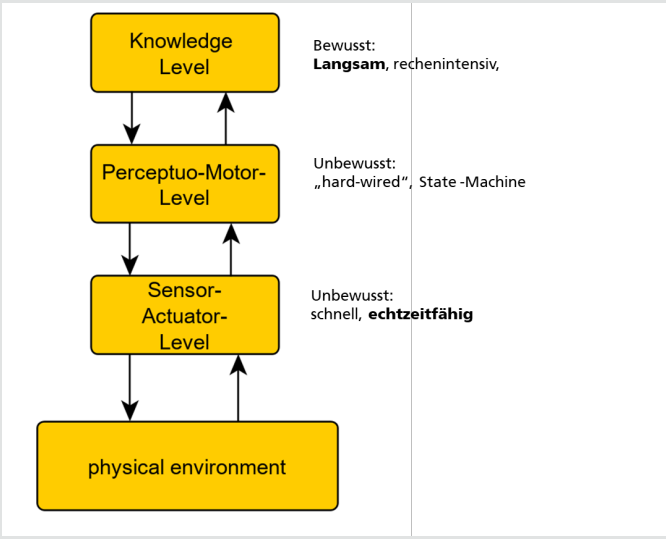
Cyberphysikalische Systeme als Netzwerke von Agenten

- Ein Agent kann als *Regelkreis*
  - Kognition – Entscheidung – Reaktion – Feedback der Umgebung

  aufgefasst werden, der mit anderen vernetzt ist

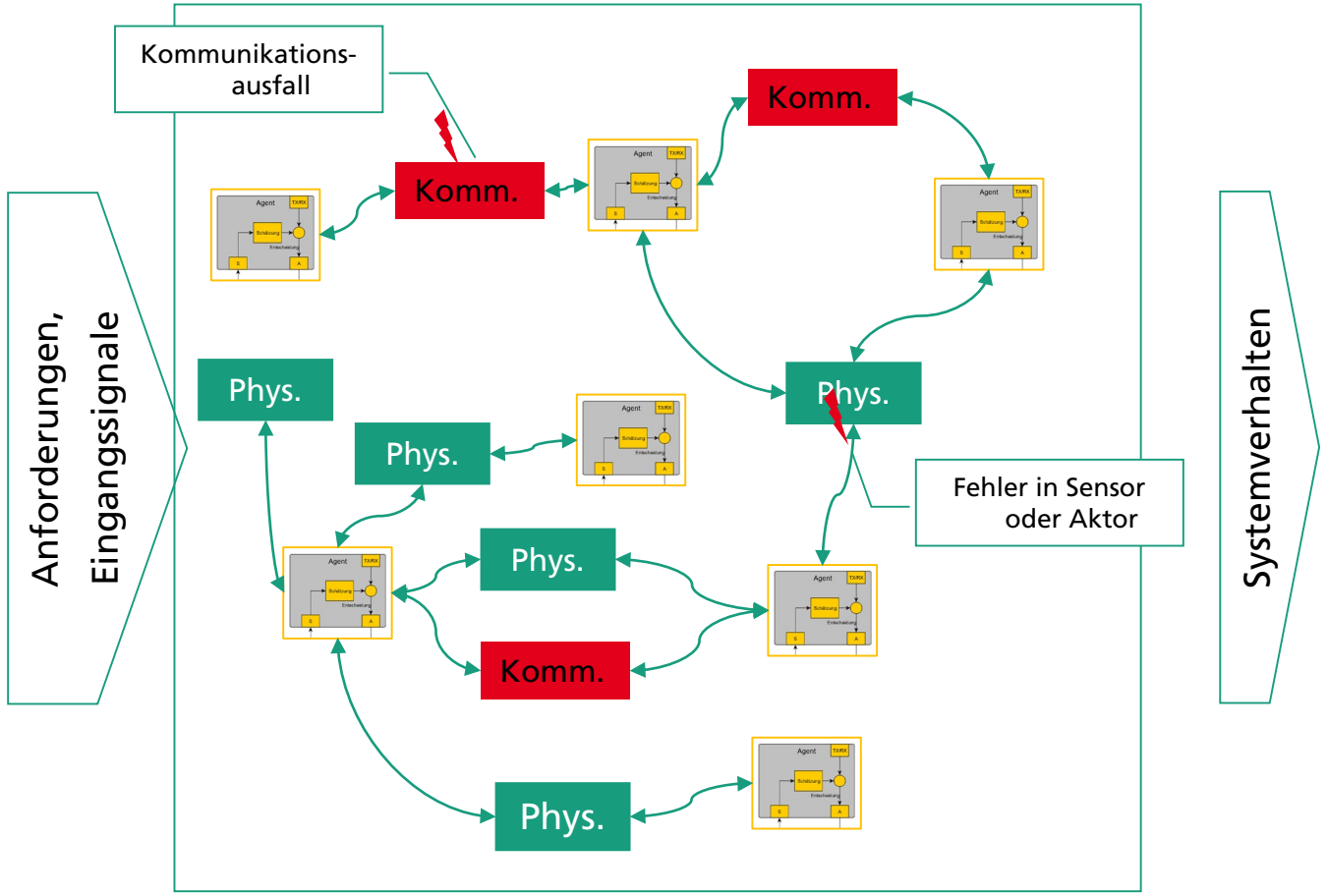


- Modellbildung in Anlehnung an die Organisation biologischer Systeme
- Verschiedene Schichten
  - Aggregation von Umweltinformationen nach oben
  - „Parametrierung“ niedrigerer Schichten durch höhere Instanzen



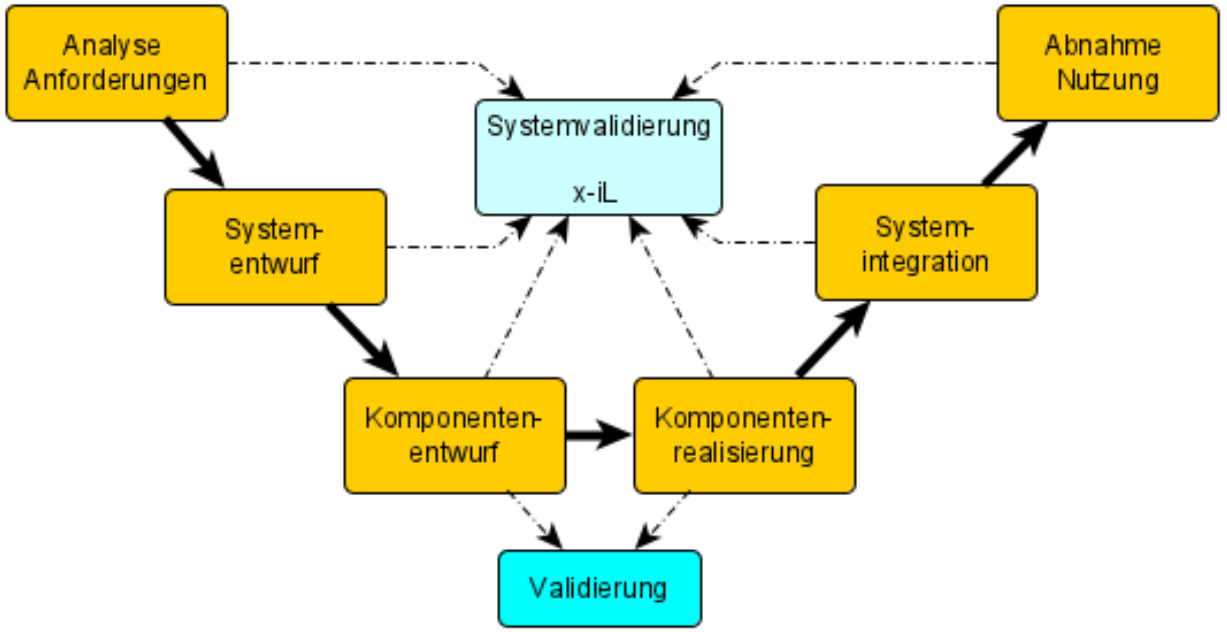
Methodische Systementwicklung

- Das Verhalten des Gesamtsystems wird von der Interaktion der Subsysteme (Agenten) bestimmt
  - Emergenz in komplexen Systemen
- Anforderungen an Zuverlässigkeit, Ausfallraten etc. in den meisten Anwendungen
- Perspektive: Adaptive Subsysteme erhöhen die Resilienz
- Komplexe Interaktion *adaptiver* Systeme erschwert die Fehleranalyse



Methodische Systementwicklung

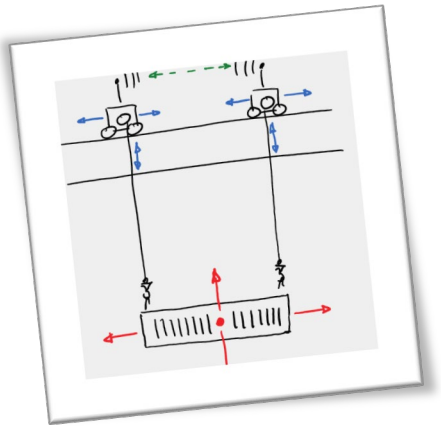
- Relevant v.a. für komplexe Systeme mit hoher Anforderung an Zuverlässigkeit und Sicherheit
- Standardisierte Vorgehensweise nach VDI 2206 zur Entwicklung mechatronischer Systeme
- Trend zur Integration der Validierung möglichst früh in den Entwicklungsprozess
  - Nutzung virtueller Methoden
  - Integration von Simulation und Versuch (xiL)

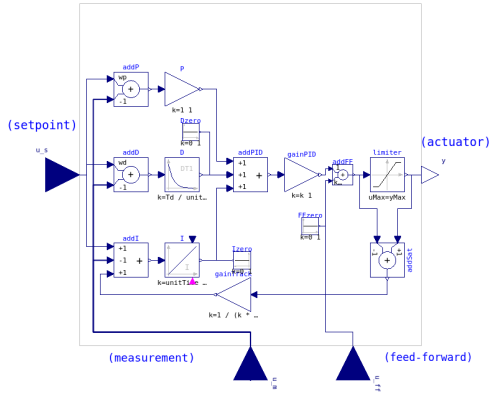


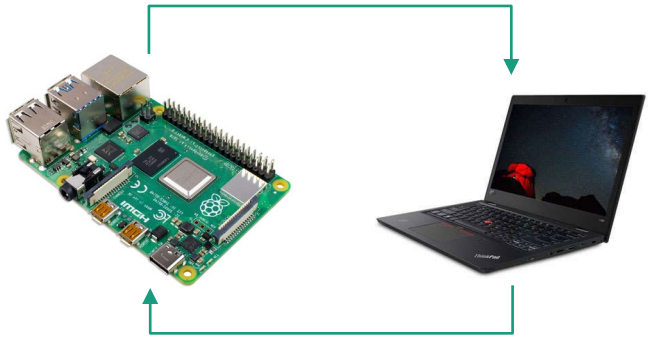
→ Schnellere und effizientere Produktentwicklung

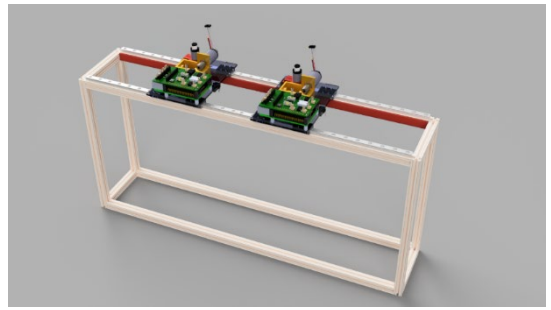
Methodische Systementwicklung

- *Durchgängige* modellbasierte Entwicklung von der Analyse der Anforderungen bis zur Realisierung
  - Nutzung von Tools zur automatischen Codegenerierung
  - Nutzung industriegängiger Werkzeuge bzw. Schnittstellen
  - Schnelle Entwicklung, frühzeitige Identifikation von Schwachstellen



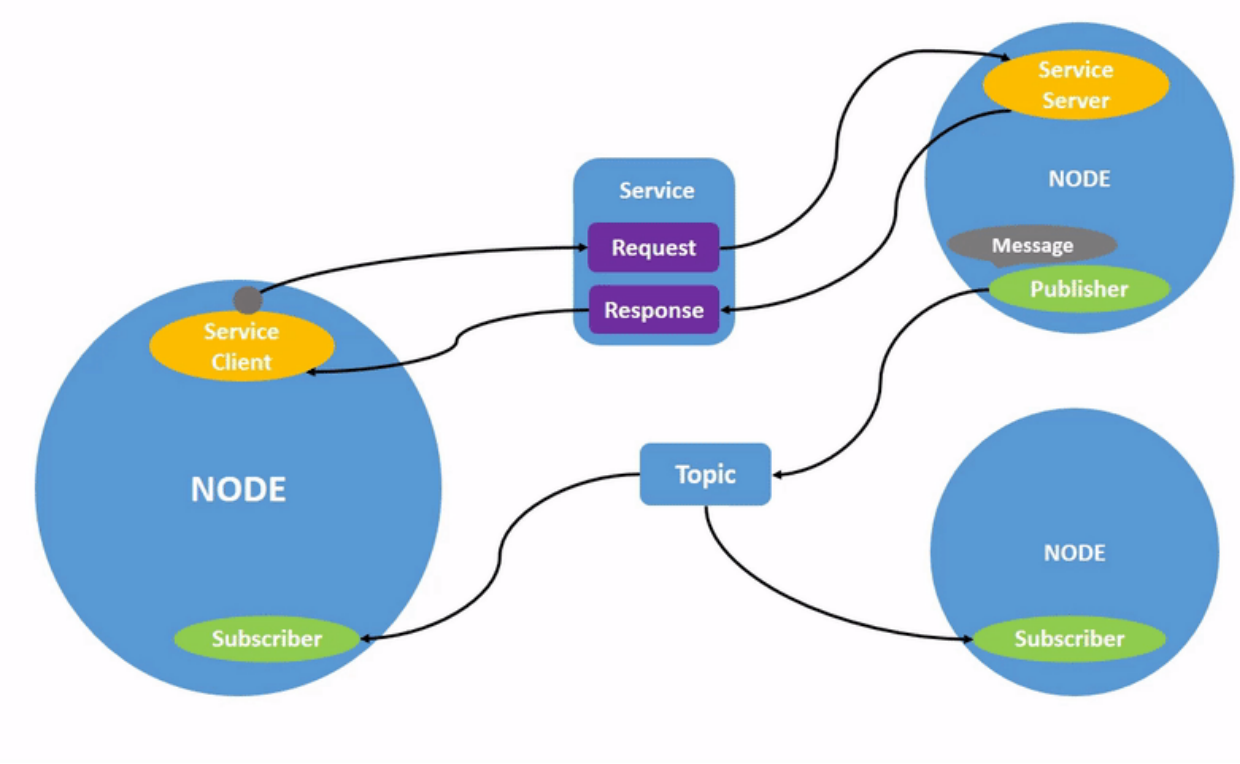






Fraunhofer IIS

Konzept auf Basis von ROS2



- Kommunikationsbasierte Architektur
  - Systemkomponenten als Knoten (Nodes)
  - Standardisierte Kommunikationsschnittstelle über topics und services
- Vollständig verteilte Architektur
  - Systeme können im Netzwerk verteilt simuliert (oder realisiert) werden
  - Nodes können beliebige Informationsverarbeitung realisieren
    (Simulation, Hardwareschnittstellen)
  - Zahlreiche Schnittstellen zu Industriehardware bereits vorhanden
- In ROS2: Echtzeitfähigkeit durch QOS policies und Speicherallokierung

https://index.ros.org/doc/ros2/Tutorials/Understanding-ROS2-Nodes/

Fraunhofer IIS

Umsetzung einer Systemsimulation mit ROS und Modelica

- Einbindung von FMU in ROS Systeme
  - Simulation von physikalischen Regelstrecken
  - Anwendung für die Systemsimulation, aber auch als mitrechnender digitaler Zwilling in der realisierten Anwendung
- Synchronisierung der Zeitbasen zwischen FMU und ROS
- Entwicklung von Bosch, als Open Source Bibliothek verfügbar

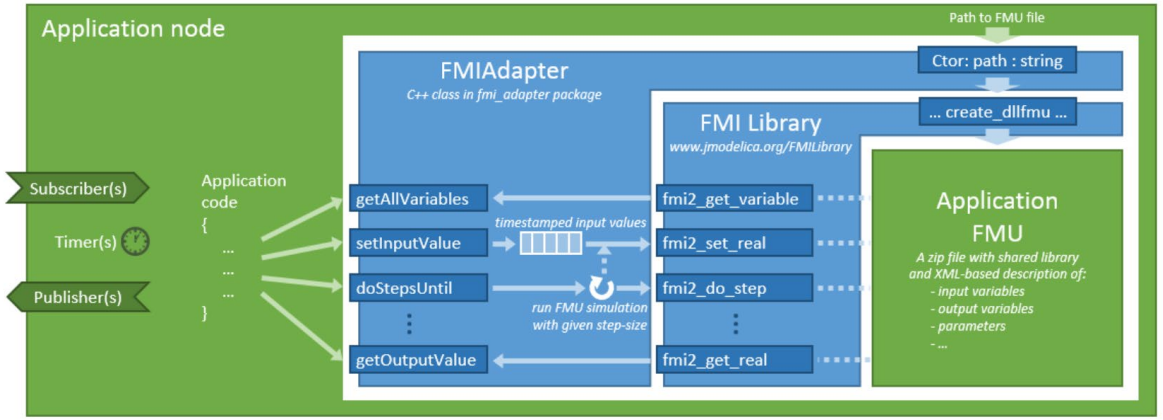
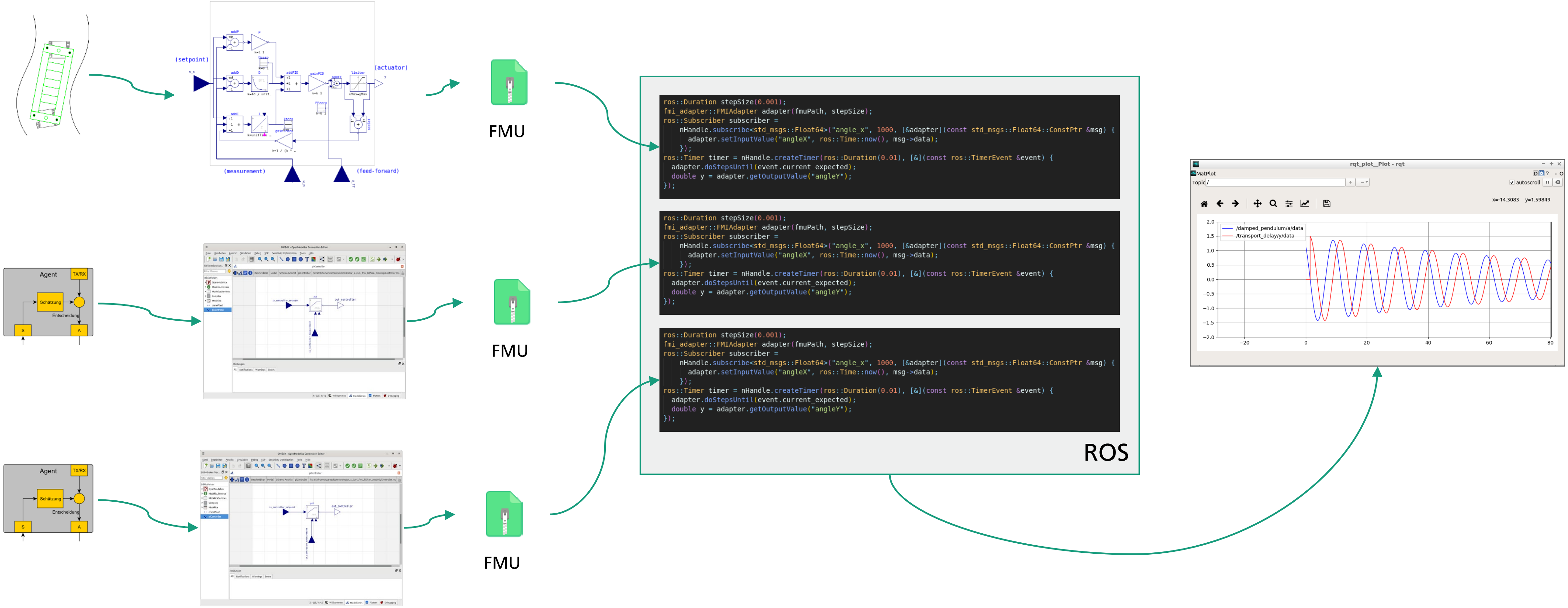


Figure 11. Architecture diagram from fmi_adapter package, illustrating library use

FMI	ROS
input variable	subscription
output variable	publisher
state variable	*no explicit counterpart*
parameter initialization	parameter server
simulation time	ROS clock - offset
communication step-size	timer

Schröder, Nikolas; Lenord, Oliver; Lange, Ralph (2019): Enhanced Motion Control of a Self-Driving Vehicle Using Modelica, FMI and ROS. In: Proceedings of the 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019. The 13th International Modelica Conference, Regensburg, Germany, March 4–6, 2019, 04.03.2019 - 06.03.2019: Linköing University Electronic Press (Linköping Electronic Conference Proceedings), S. 441–450, zuletzt geprüft am 06.05.2020.

Umsetzung einer Systemsimulation mit ROS und Modelica

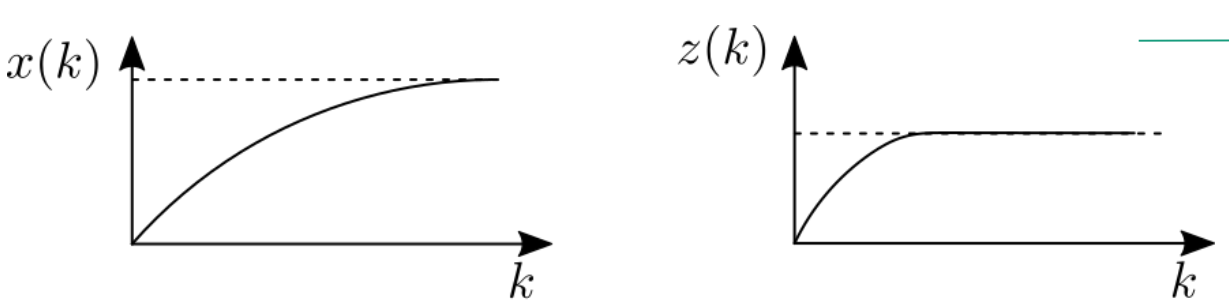


Skalierbar – Einbindung zusätzlicher (Sub)Systeme leicht möglich

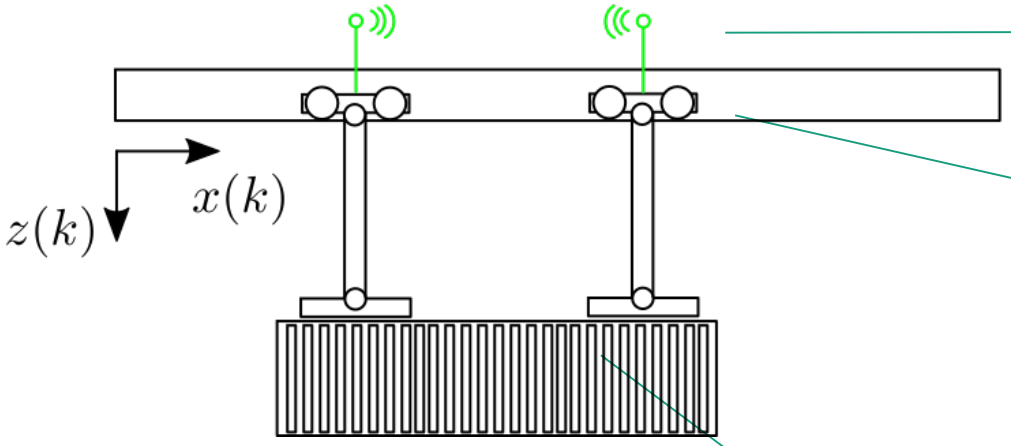
Offen – ROS ermöglicht eine domänenspezifische Implementierung von Teilsystemen

Fraunhofer IIS

Demosystem und Testumgebung



Vorgabe von Trajektoren für die Last



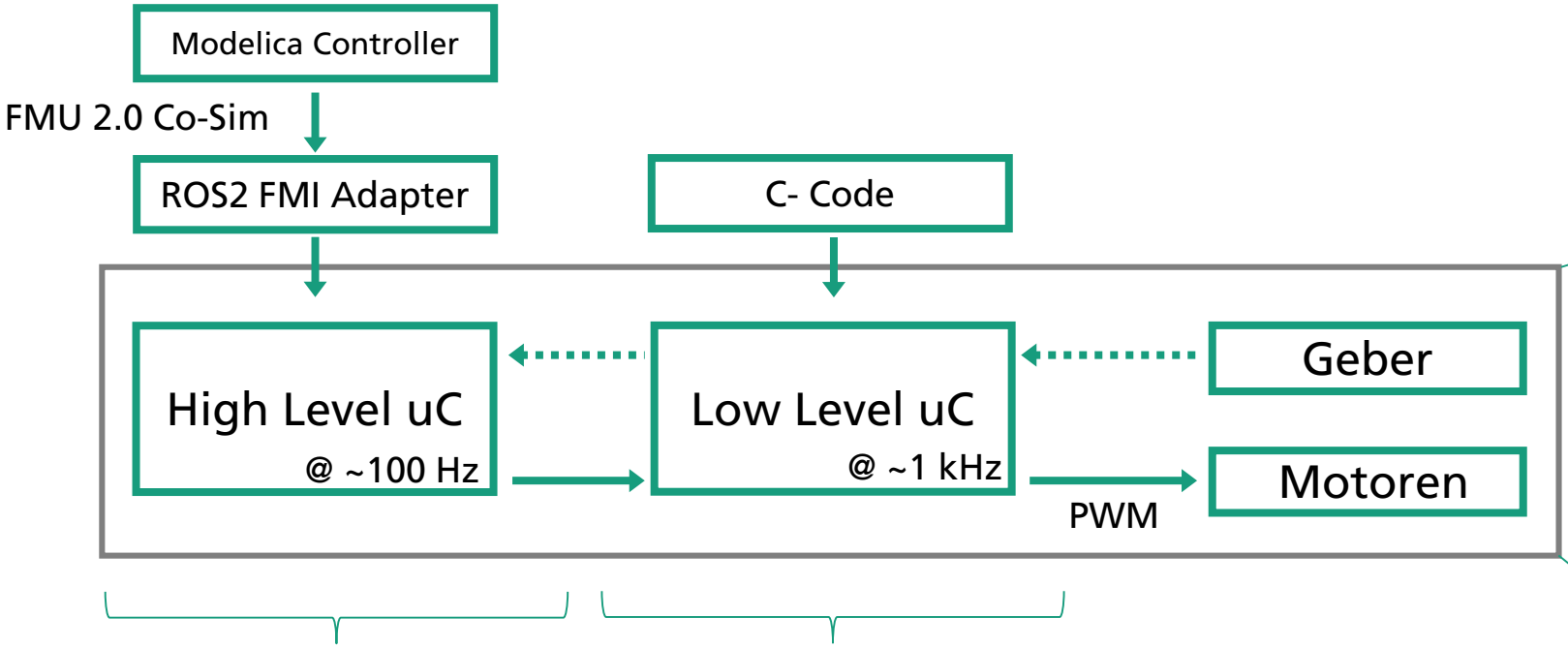
Kommunikations-system

Lokale Antriebsregelung

Last wird von beiden Kränen bewegt

Fraunhofer IIS

Demosystem - Systemkonzept



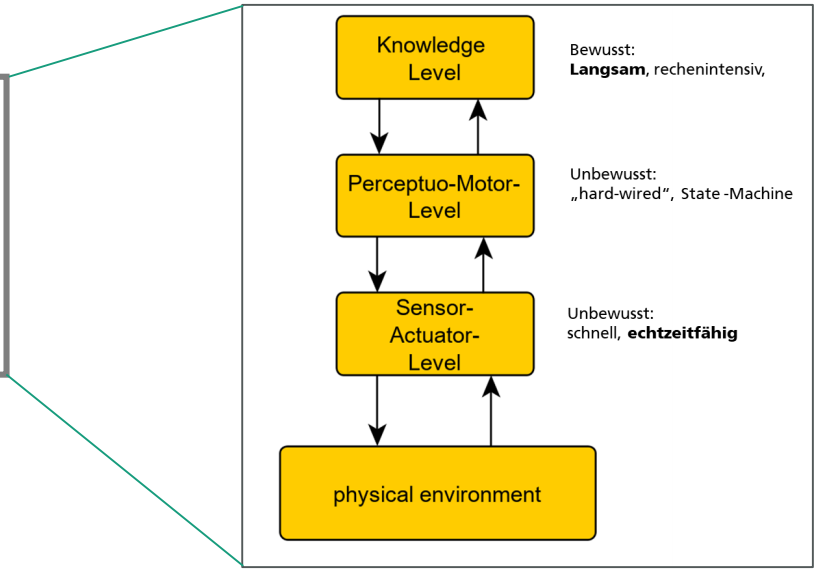
State Machine

Trajektorienregelung

Notfallfunktionen

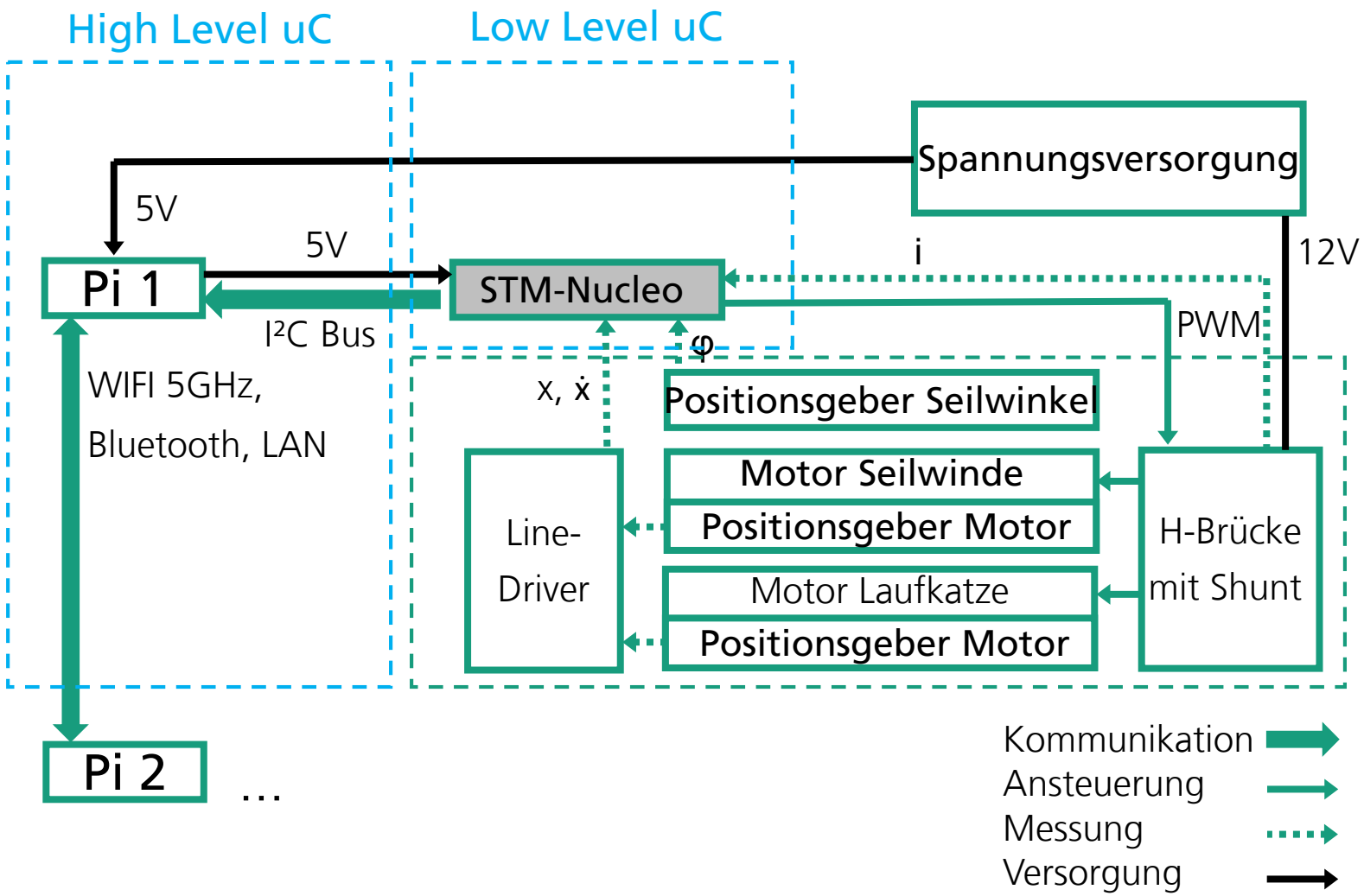
Kalibrierung

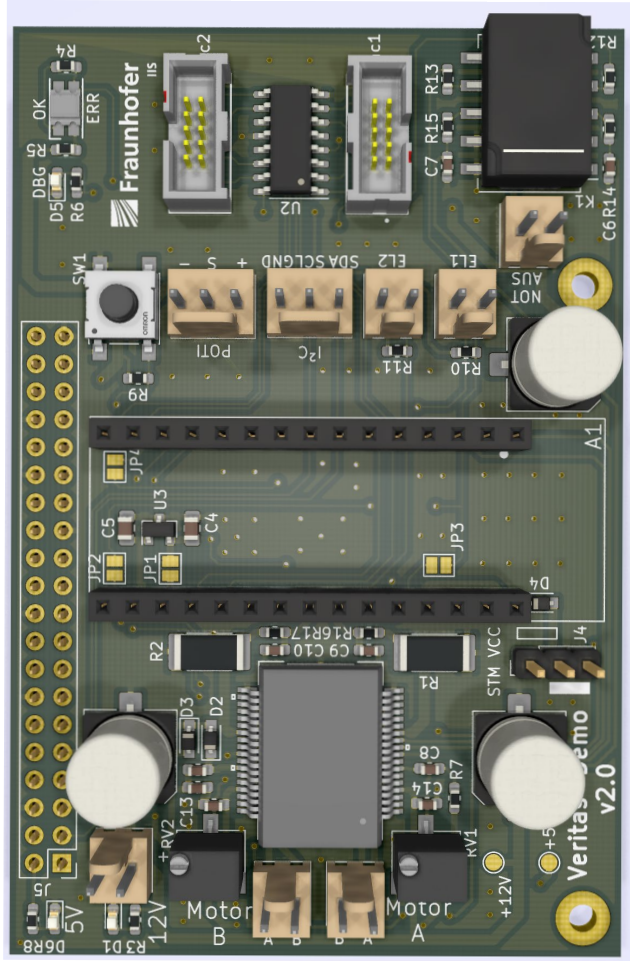
Stromregelung



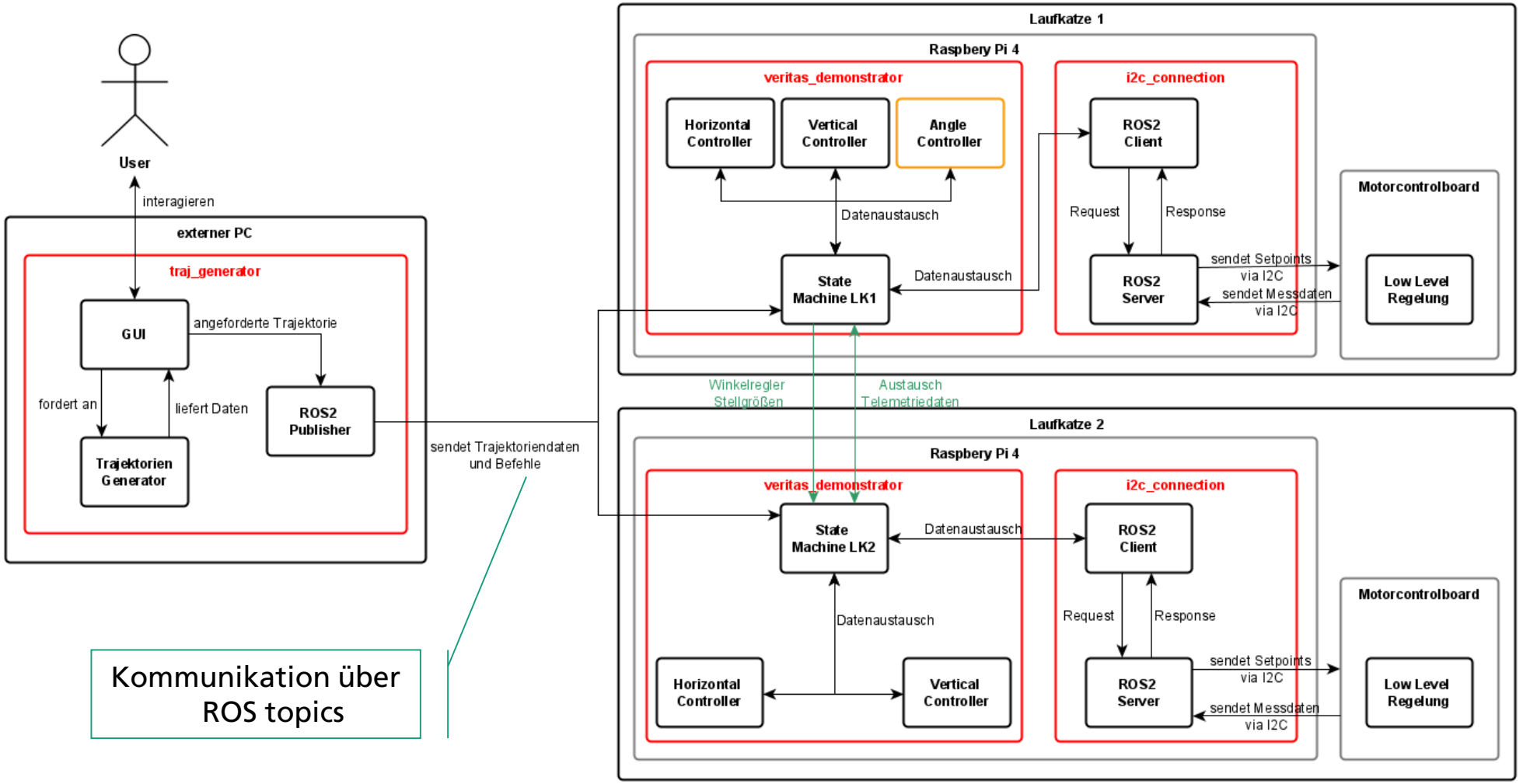
Parallele zum Agenten

Demosystem - Hardwarekonzept



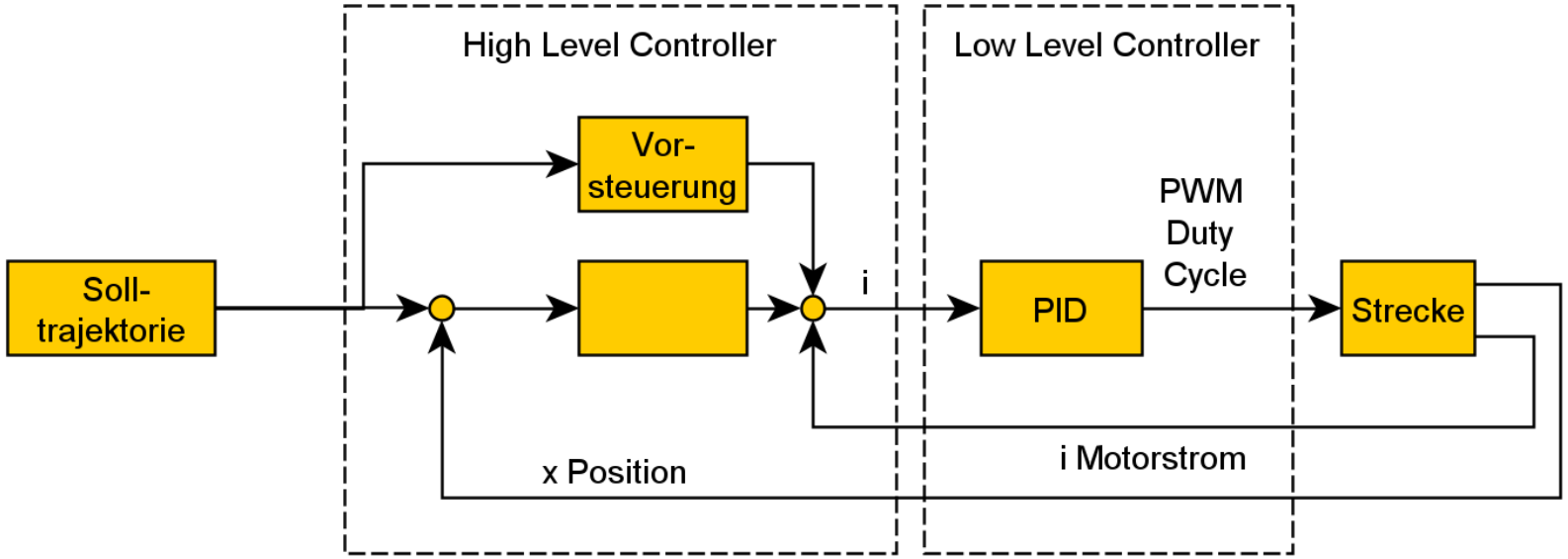


Demosystem: Systemarchitektur mit ROS2 und Hardware

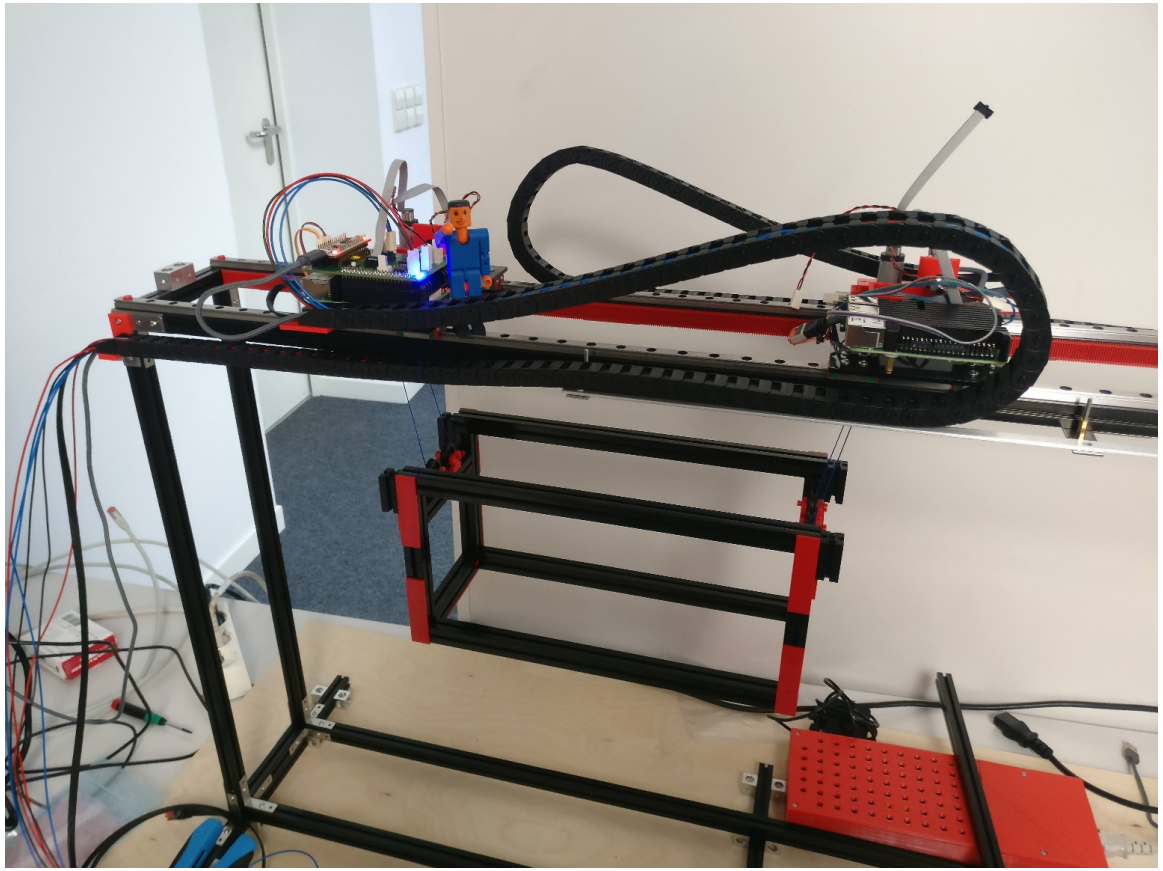


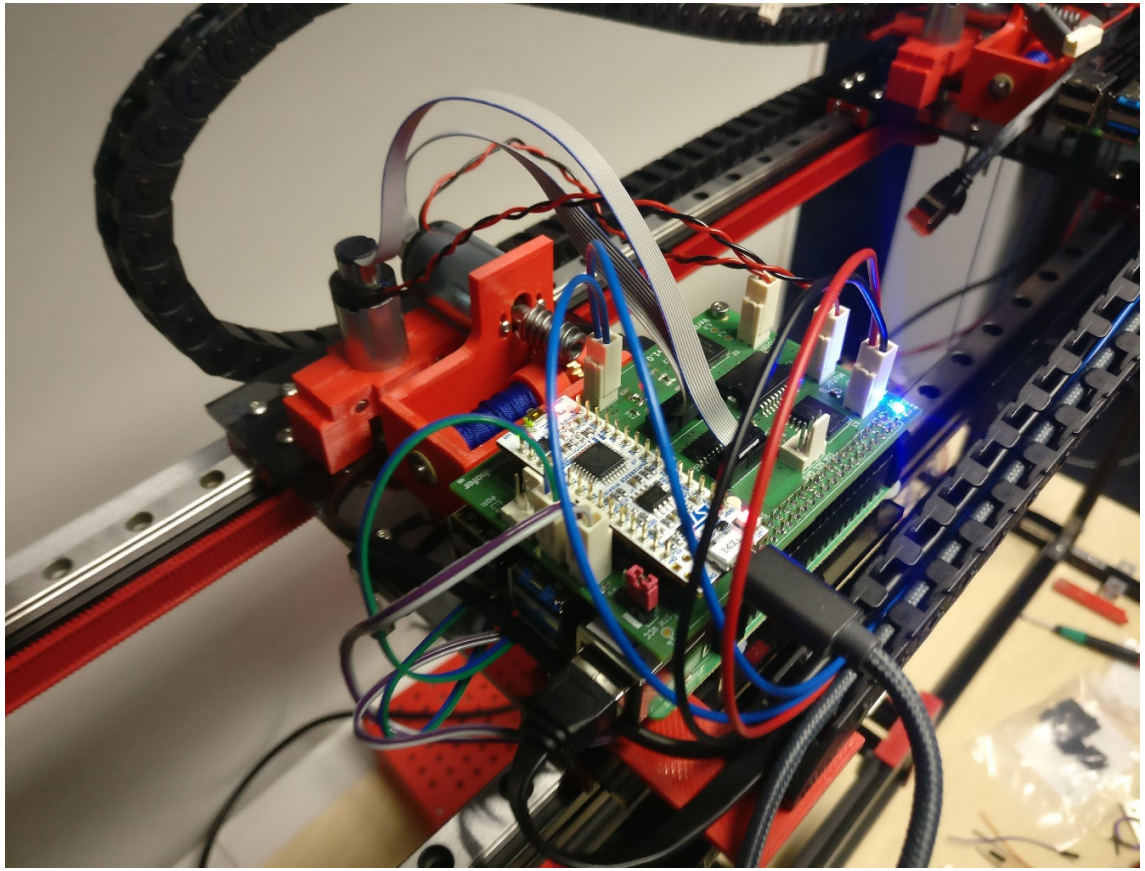
Hierarchisches Regelungskonzept

- Vermaschung von Regelkreisen
- Realisierung unterschiedlicher Abtastraten auf den Hardwareplattformen



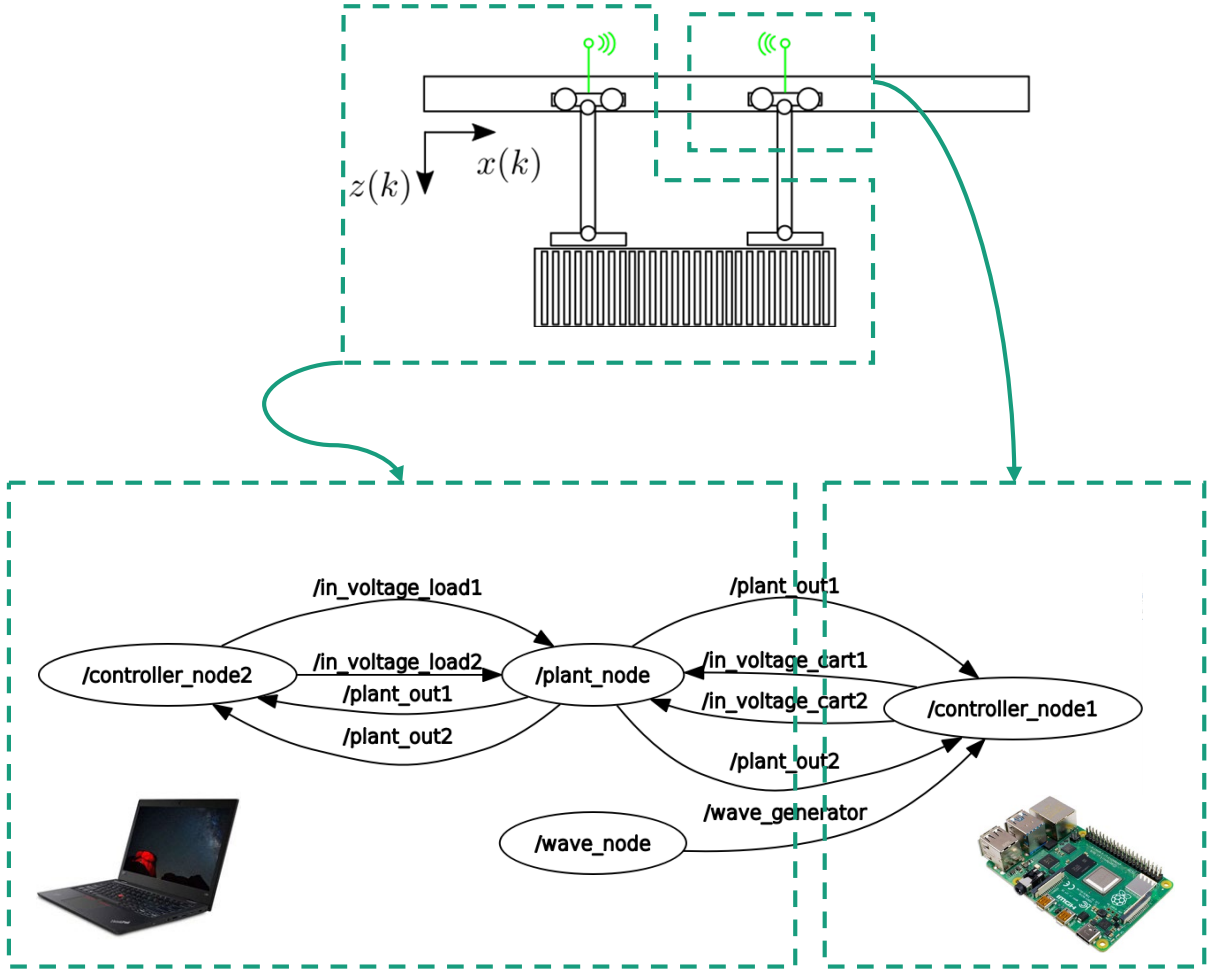
Demosystem: Aufbau der Hardware





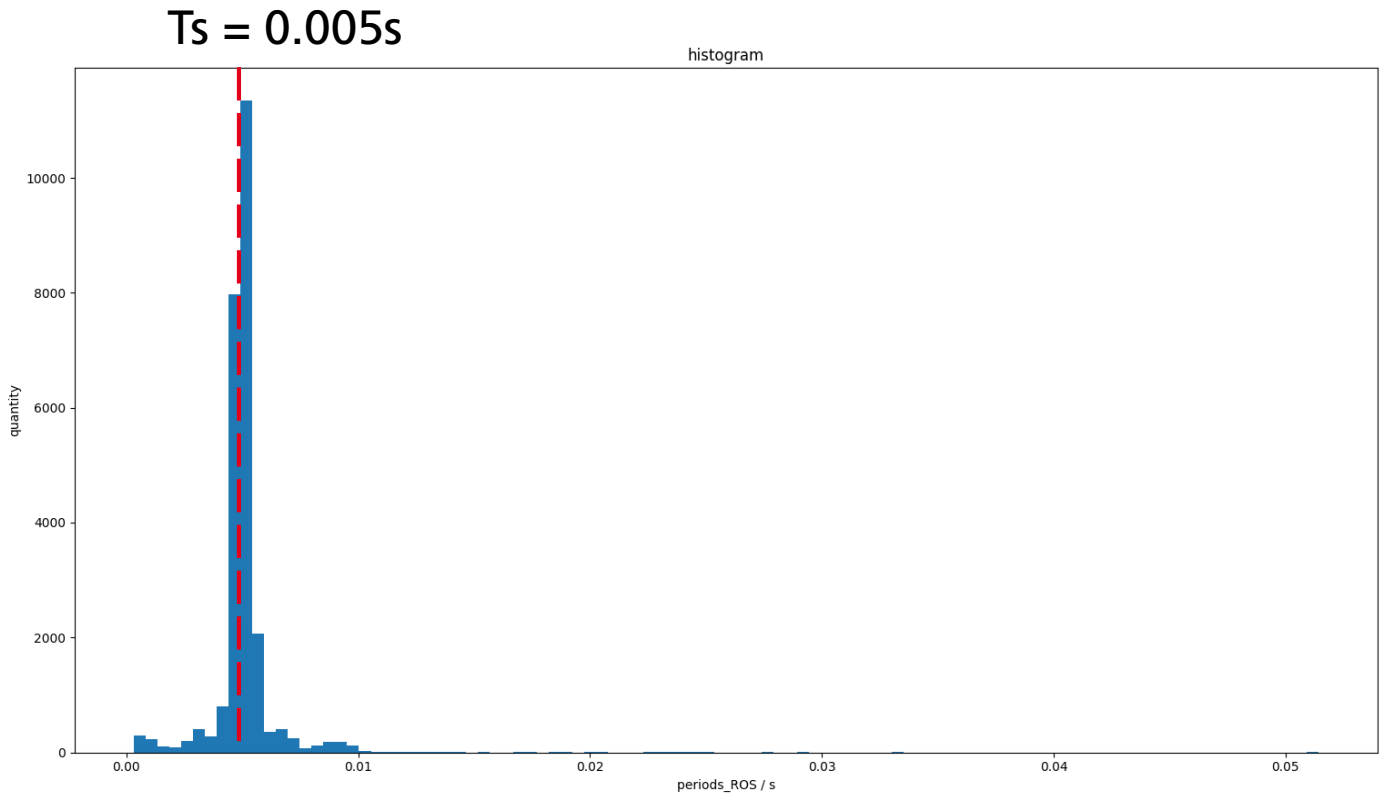
Hardware in the Loop Test

- Ausnutzung der verteilten Architektur von ROS
- Schrittweise Implementierung auf der Zielhardware für systematisches Testen möglich
- Prüfung von Fehlerszenarien
  - Ausfälle von Komponenten
  - Ausfall / Verschlechterung der Kommunikation
  - …
- Anforderungen an den HiL Test – möglichst geringe Imperfektionen des Testsystems

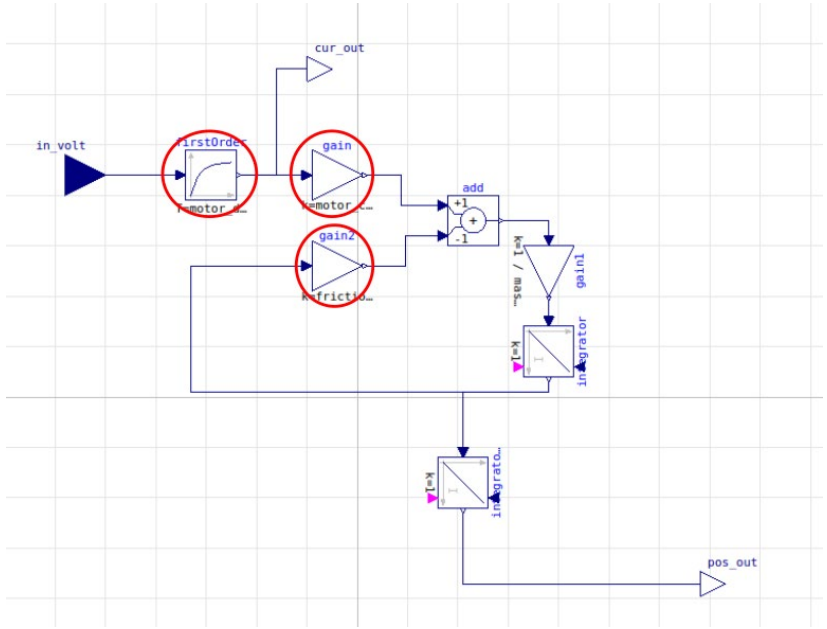


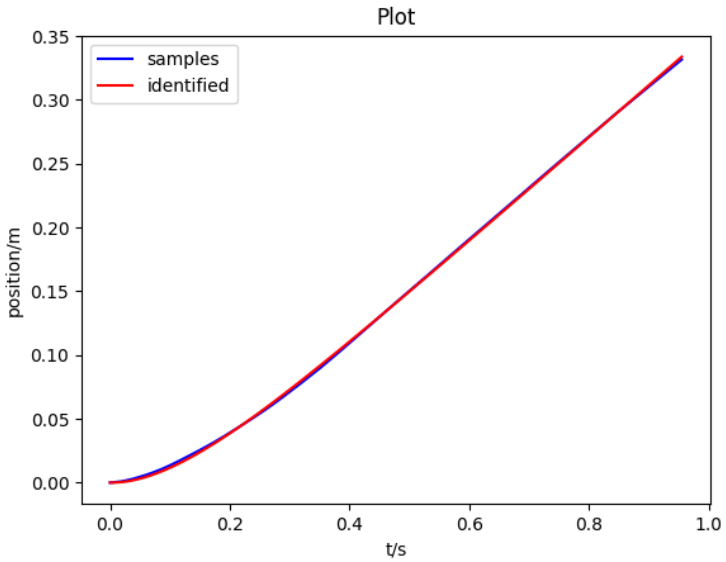
Hardware in the Loop Test

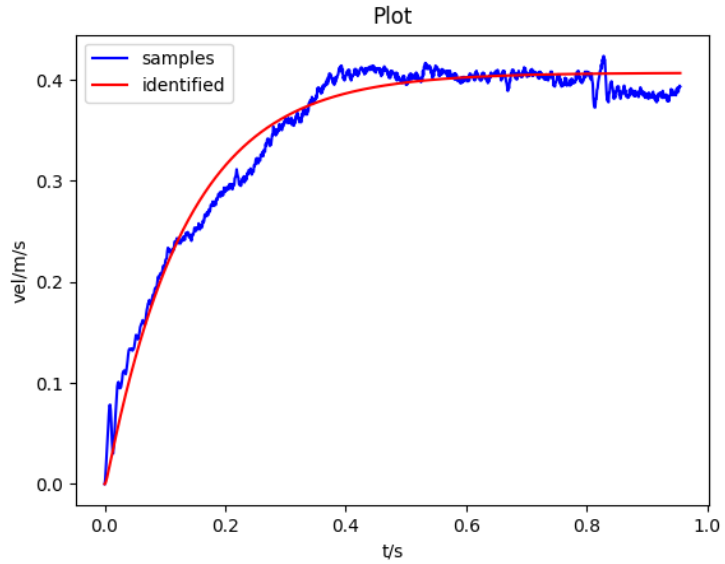
- Messung des Jitters
  - Simulation der Strecke auf PC
  - Regler auf Raspberry Pi
- Präzise Taktung insbesondere bei Closed Loop Tests notwendig
- Jitter für erste Tests ausreichend klein
  - Begrenzung der Abtastrate des Reglers auf 0.01 s
- Verbesserungsmöglichkeiten
  - RT-Kernel
  - Leistungsfähigere Plattform



Erste Messergebnisse und Parameterabgleich







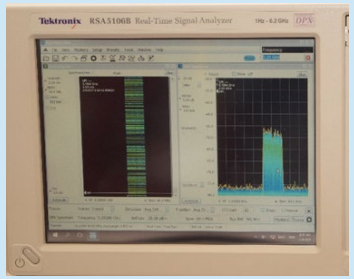
- Parameterabgleich der Motoren
  - Motorkonstante
  - Motorverzögerung
  - Reibungskonstante

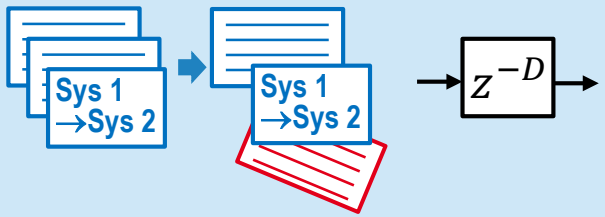
Position und Geschwindigkeit

Messung vs. Simulation

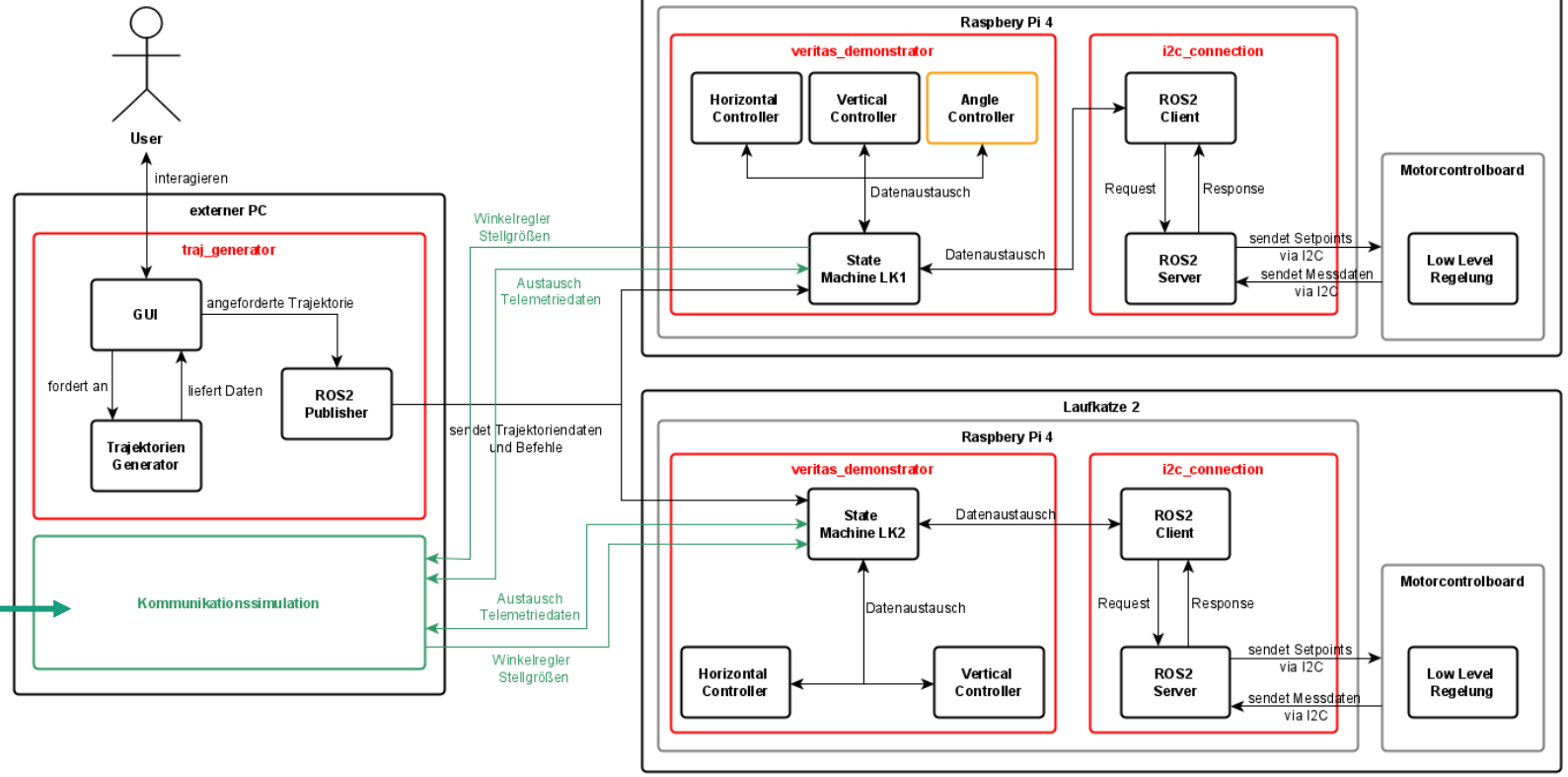
Ausblick: Integration einer Simulation des Kommunikationssystems

Experimentelle Untersuchung von Kommunikationsproblemen





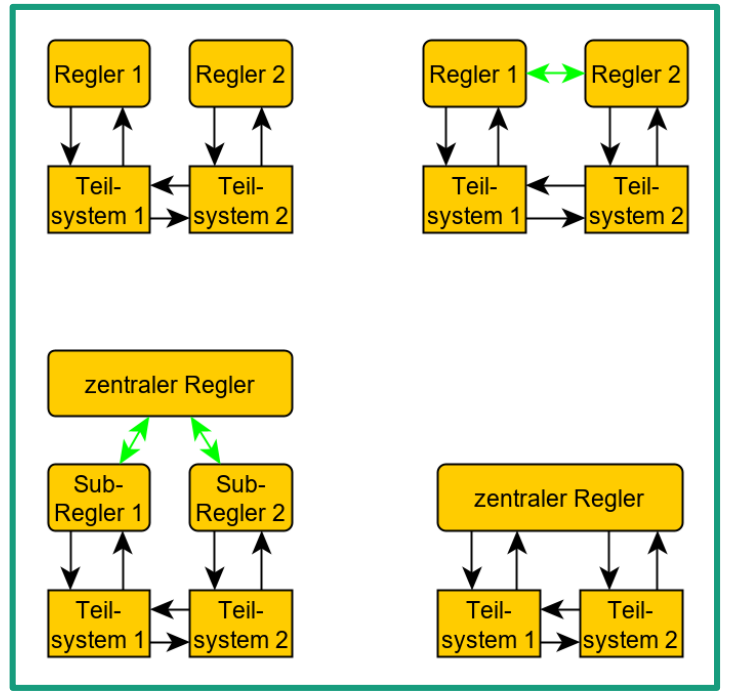
Modelle für Effekte auf Telegrammebene



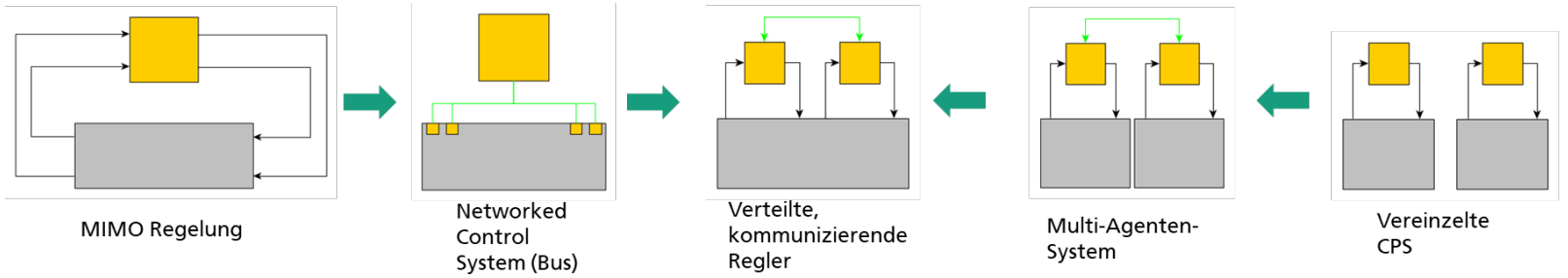
Fraunhofer IIS

Zusammenfassung und weiterführende Arbeiten

- Framework für die modellbasierte Entwicklung und Evaluierung verteilter Systeme
- Umsetzung eines einfachen Demonstrationsszenarios als Testumgebung



- Umsetzung eines Modells des Kommunikationssystems
- Vergleichende Betrachtung unterschiedlicher Regelungskonzepte
  - Performance vs. Robustheit gegen Fehler und Störungen
  - Verteilung von komplexen Algorithmen vs zentrale Ansätze
- Management der Kommunikation in unsicherer Umgebung



Weiterführende Arbeiten (II)

- Implementierung von ROS2 (MicroROS) auf der low-level uC Plattform
  - Vollständige modellbasierte Entwicklung von Algorithmen
- Integration von Schnittstellen zu weiteren multiphysikalischen Simulationsumgebungen
  - Erschliessung weiterer Anwendungen (UAV, Robotik)
- Dokumentierte Veröffentlichung des Frameworks als Open Source Projekt

CONTACT

Dr. Dirk Mayer

Head of
Distributed Data Analysis and Control

✉ dirk.mayer@eas.iis.fraunhofer.de
☎ +49 351 4640- 720

Dr. Andreas Frotzscher

Group manager
Industrial wireless communication

✉ Andreas.Frotzscher@eas.iis.fraunhofer.de
☎ +49 351 4640-836

Fraunhofer Institute for Integrate Circuits IIS
Division Engineering of Adaptive Systems EAS
Zeunerstraße 38
01069 Dresden
Germany

www.eas.iis.fraunhofer.de

Fraunhofer IIS