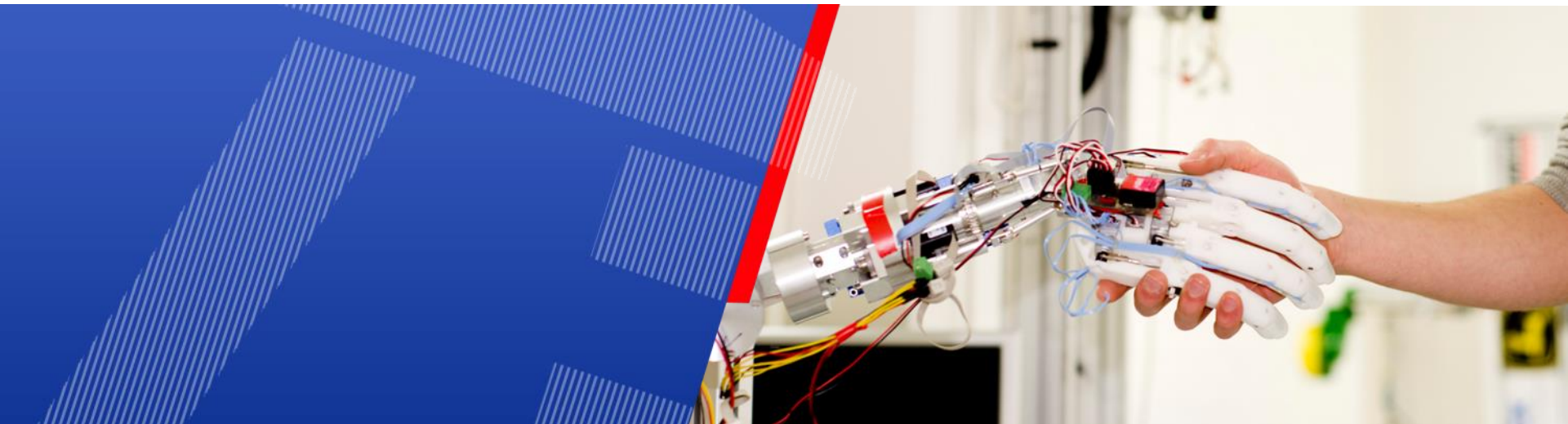


# Zusammenfassung und Anwendungsbeispiel



**Prof. Karsten Berns**

Robotics Research Lab

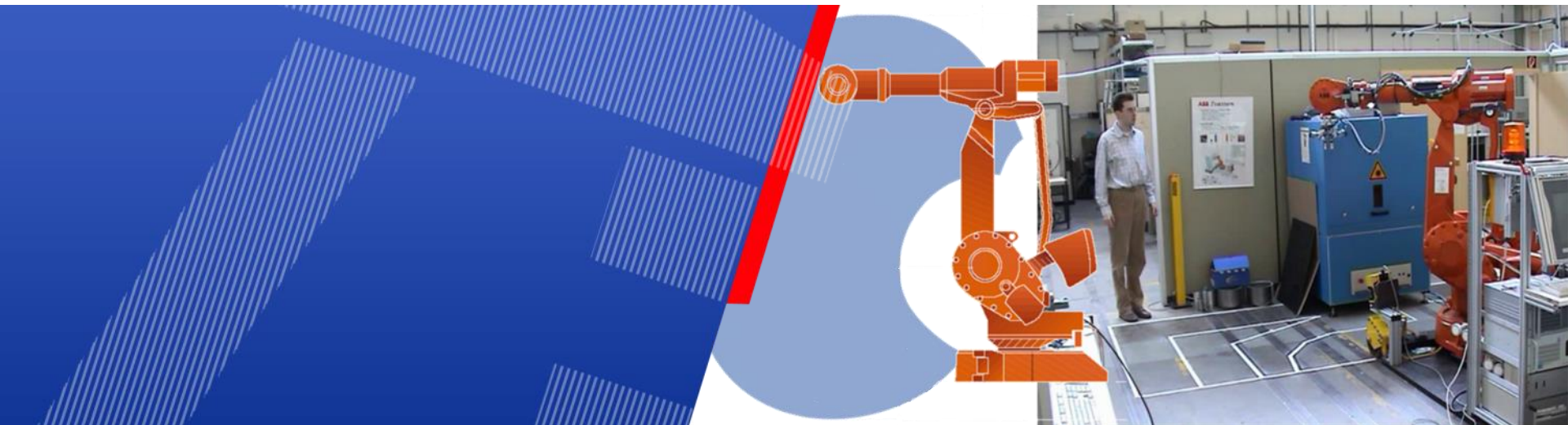
Department of Computer Science

University of Kaiserslautern, Germany

# Inhalt

- Anwendung „humanzentrierte Automatisierung“
- Robotersystem, Hardware und Kommunikationssystem
- Steuerungskonzept
- Testergebnisse
  - Quantitative Aussagen zur Reaktionszeit des Systems
  - Optimieren des Systems
  - Allgemeine Berechnung des Sicherheitsbereichs
- Zusammenfassung der Vorlesung

# Aufbau einer dynamischen flexiblen Sicherheitsumgebung für Roboter



Basierend auf der  
**Diplomarbeit**  
von **Roland Krieger**

# Die „humanzentrierte Automatisierung“

- Bisher: Arbeitsbereichstrennung von Mensch und Maschine
  - Mensch nur als Beobachter (Kontrolle, Wartung, ...)
- Zukunft: Kooperation, Synergien nutzen
  - Schnelligkeit, Kraft, Genauigkeit und Medienresistenz des Roboters
  - Sensorische Fähigkeiten, Wissen und Geschicklichkeit des Menschen
- Mensch-Roboter-Kooperation ist junges Forschungsgebiet (erste Publikation 1992 von Y. Nakauchi: „Multi-Agent Interface Architecture for Human-Robot Cooperation“)
- Fraunhofer Institut (IPA) prägt Begriff „Humanzentrierte Automatisierung“

# Problemstellung und Zielsetzung

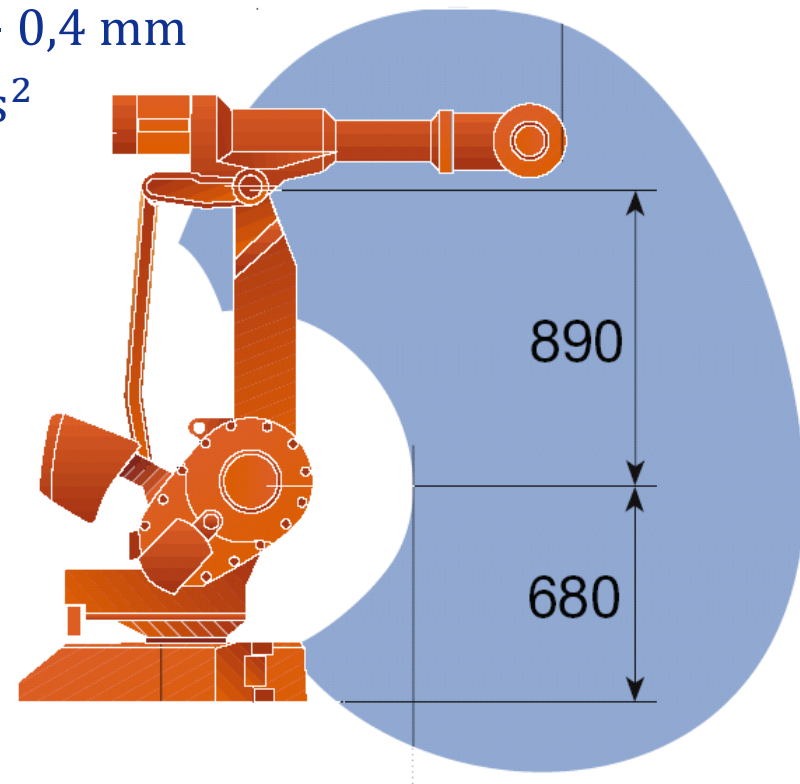
- Problem: Industrieroboter „blind“ gegenüber der Umwelt
  - Meist nur interne Sensoren
  - Häufig nur binäre Erkennung (Arbeitsraum frei/besetzt)
- Ziel: Aufbau einer dynam. flexiblen Sicherheitsumgebung
  - Verschiedene Sicherheitsbereiche mit diversen Aktionsauslösungen: Not-Aus, Verlangsamung der Geschwindigkeit oder akustischer Warnmeldung
  - Detektion von Menschen durch Zusammenwirken verschiedener Sensoren unterschiedlichen physikalischen Prinzips: Ultraschall, Laserscanner, Lichtvorhang, Kontaktmatten, Radar usw.
- Adaptive Bahnplanung unter Berücksichtigung des aktuellen Hindernisraums
- Fernziel: Interaktion des Roboters mit dem Menschen

# Überwachung des Schutzbereichs

- Möglichkeiten
  - Überwachung des Roboterarmes  
(Einpunkt-, Mehrpunkt-, oder Ganzarmüberwachung)
  - Überwachung des Arbeitsbereiches
- Statische oder Dynamische Überwachung
  - Enge Kooperation mit Menschen gewünscht
  - Flexibles Reagieren auf unterschiedliche Situationen
  - Somit statische (vor Programmausführung festgelegte Überwachungsbereiche) nicht geeignet

## Robotersystem: ABB IRB4400

- Max. Reichweite: 1,96m
- Handhabungskapazität: 45 kg
- Positioniergenauigkeit: 0,07 – 0,1 mm
- Wiederholgenauigkeit: 0,25 – 0,4 mm
- Max. TCP Beschleunigung: 14 m/s<sup>2</sup>





## Steuerungshardware: Sicherheits-SPS PSS3000

- Mehrkanaliger, diversitärer Aufbau
- Integrierte Test- und Sicherungsalgorithmen
- Genügt Anforderungen der EN 954 – 1, 11/94 bis Kategorie 4
- Datenspeicher: 234 KByte, Programmspeicher: 1024 KByte

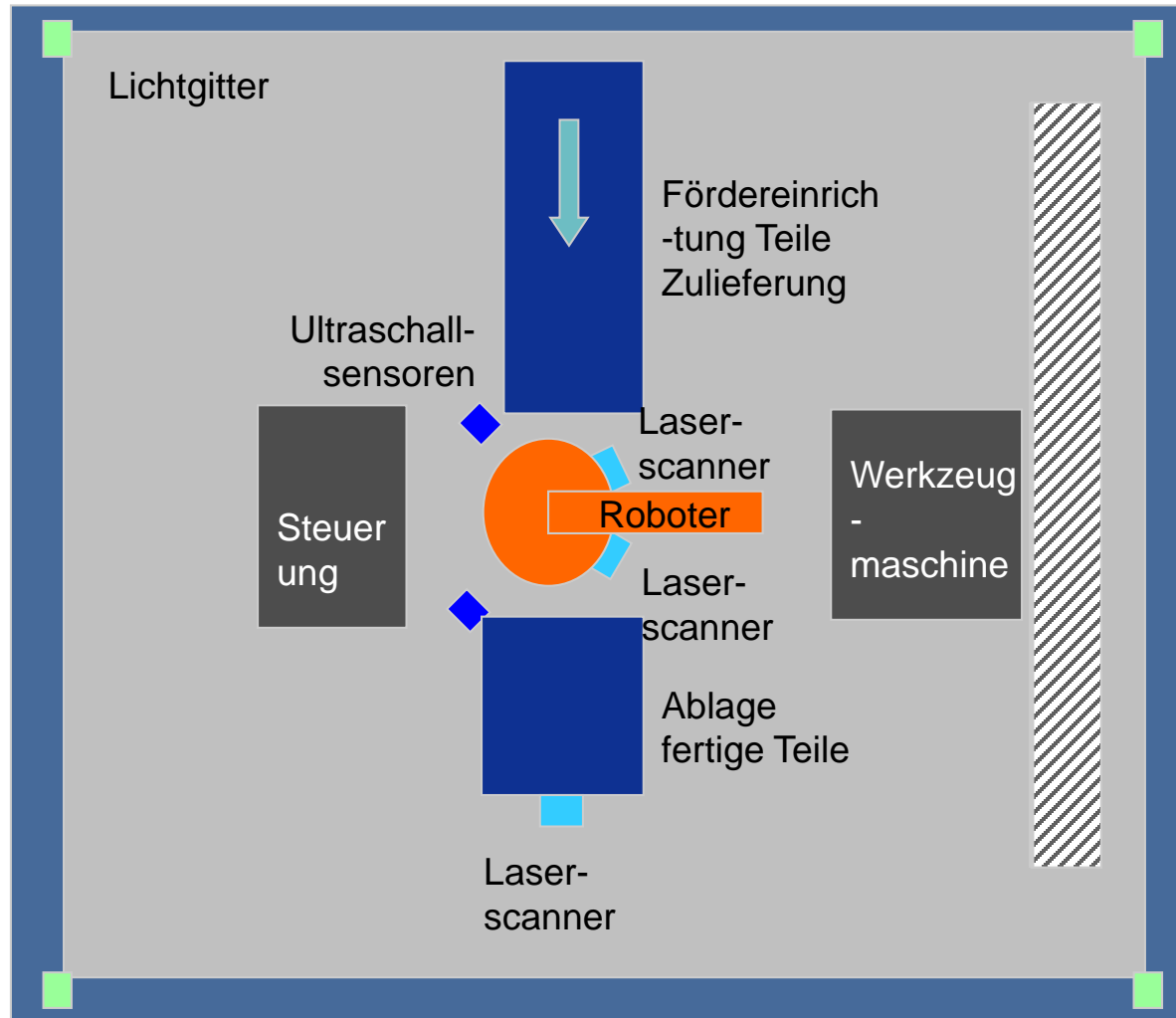




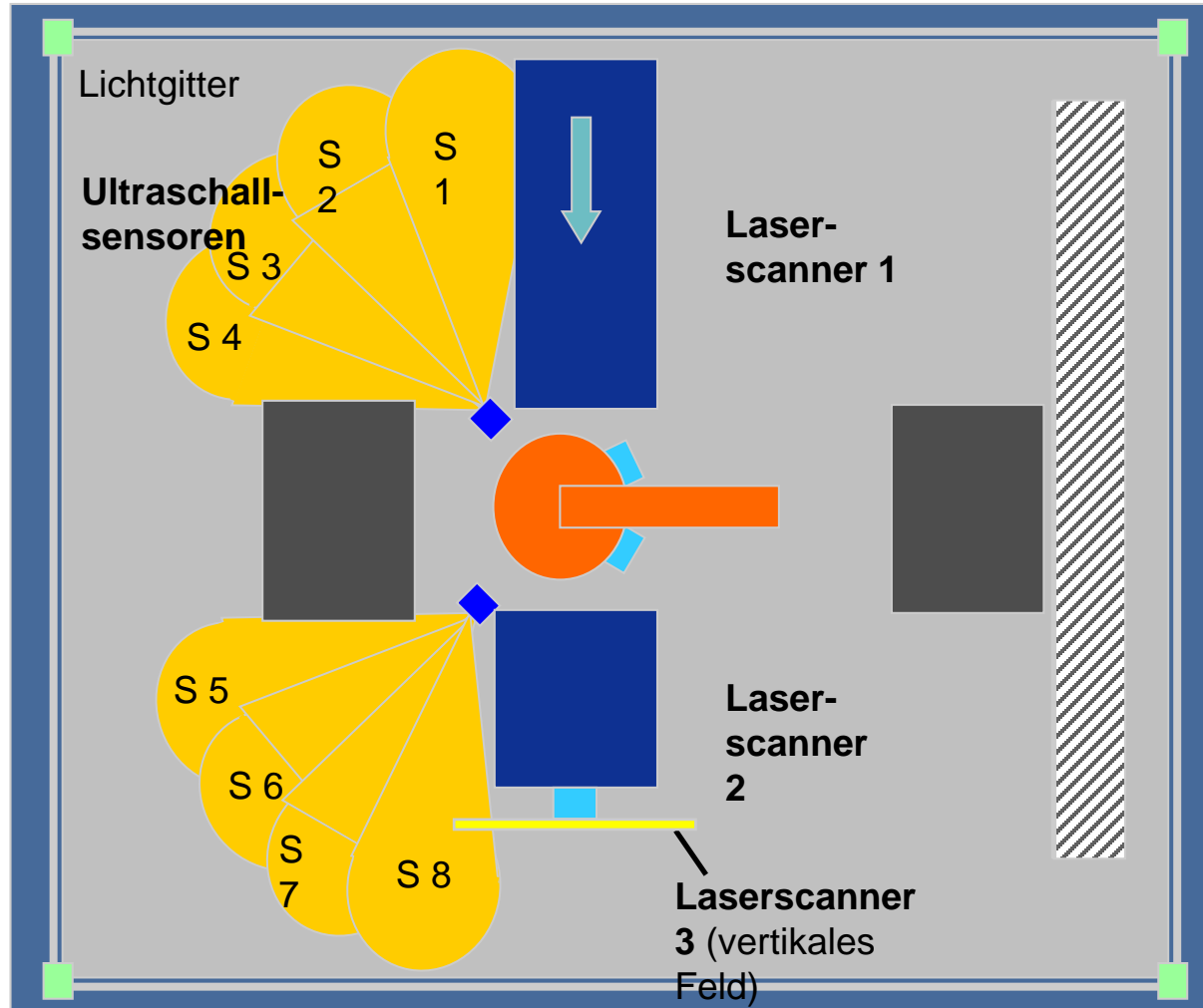
# Sensorik

- Laserscanner
  - Bei Maschinensicherheit etabliert, große Industrieakzeptanz
  - Abdecken großer Sicherheitsbereiche (bis 12 m)
  - Flexibel einsetzbar (Schutz-/Warnfelder)
  - Kostenintensiv
- Ultraschallsensoren
  - Kostengünstige Massenware
  - Extrem störanfällig, nicht zertifiziert u. sicherheitsgerichtet
- Lichtgitter
  - In vorliegender Anwendung genutzt, um Geschwindigkeit des Roboters zu beeinflussen (kein Not-Aus)
  - Geeignet für sicherheitsgerichtete Anwendungen
  - Bewährt im industriellen Einsatz

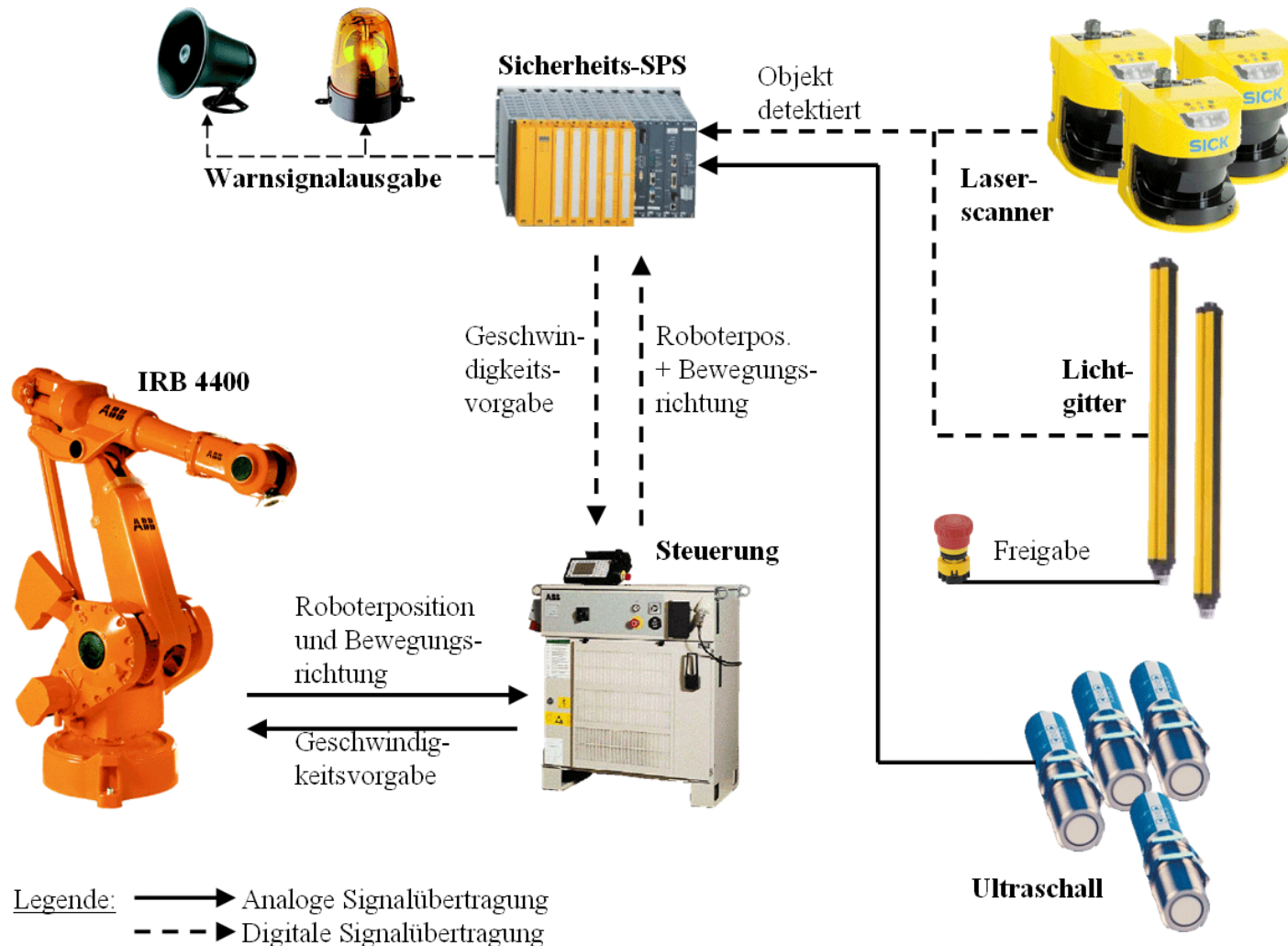
# Arbeitszelle: Systemaufbau



# Arbeitszelle: Messbereiche



# Kommunikation zwischen Systemkomponenten



# Statische Schutzbereiche mit Not-Aus

- Analog zur klassischen Roboterzellen
- Gesamter Arbeitsraum statisch überwacht
- Betreten der Zone 0 unterbricht das Lichtgitter und schaltet die Anlage in den Not-Aus-Zustand
- Druck auf Freigabeschalters gibt Anlage frei

# Stat. Schutzbereich mit Not-Aus u. Warnbereich

- Not-Aus Bereich zusätzlich von Warnzone umgeben
- Unterbrechung des Lichtgitters reduziert Geschwindigkeit
- Verletzung von Zone 0 lässt Signallampe aufleuchten
- Verletzung von Zone 1 stoppt Roboter
- Anlage fährt selbstständig an sobald Verletzung aufgehoben
- Druck auf Freigabeschalter fährt Anlage wieder auf ursprüngliche Geschwindigkeit hoch und erlischt Signallampe

# Dynamische Schutzbereiche

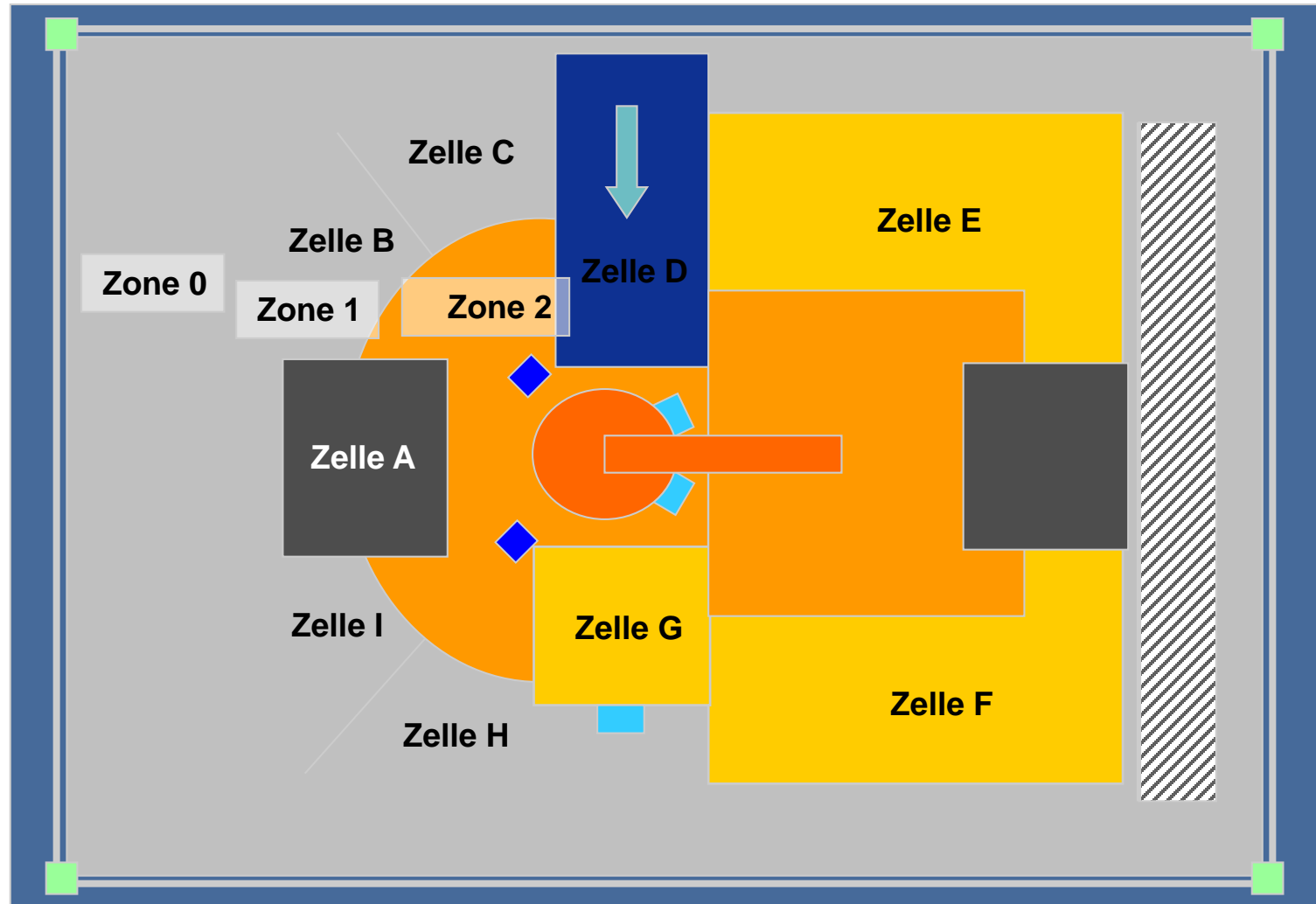
- Geschwindigkeitsanpassung und Not-Aus
- Drei gestaffelte Sicherheitszonen, deren Verletzung je nach Roboterposition ...
  - keine Reaktion,
  - eine Geschwindigkeitsreduzierung oder
  - den Stillstand der Anlage bedingt
- **Schutzbereich beinhaltet ...**
  - Gesamte Zone 2 und Zelle  $X$ , falls sich Roboter in Zelle  $X$  des Sicherheitsbereichs 1 befindet
  - Zusätzlich Zelle  $X + 1$ , falls sich Roboter in Zelle  $X$  des Sicherheitsbereichs 1 befindet und eine Bewegung in Richtung Zelle  $X + 1$  ausführt
- **Warnbereich wird von den beiden Zellen der Zone 1 gebildet, die direkt an die Zelle/n des Schutzbereichs grenzen**



# Dynamische Schutzbereiche

- Zone 0 verletzt: Roboter setzt Geschwindigkeit herab, Signallampe leuchtet auf
- Warnbereich in Zone 1 verletzt: Akustisches Signal ertönt
- Schutzbereiche in Zone 1 verletzt: Stoppen des Roboters
- Zone 2 verletzt: Stoppen des Roboters
- Schutzbereichsverletzung aufgehoben: Anlage läuft an
- Druck auf Quittierungsschalter: Anlage schaltet auf ursprüngliche Geschwindigkeit hoch, Signallampe erlischt

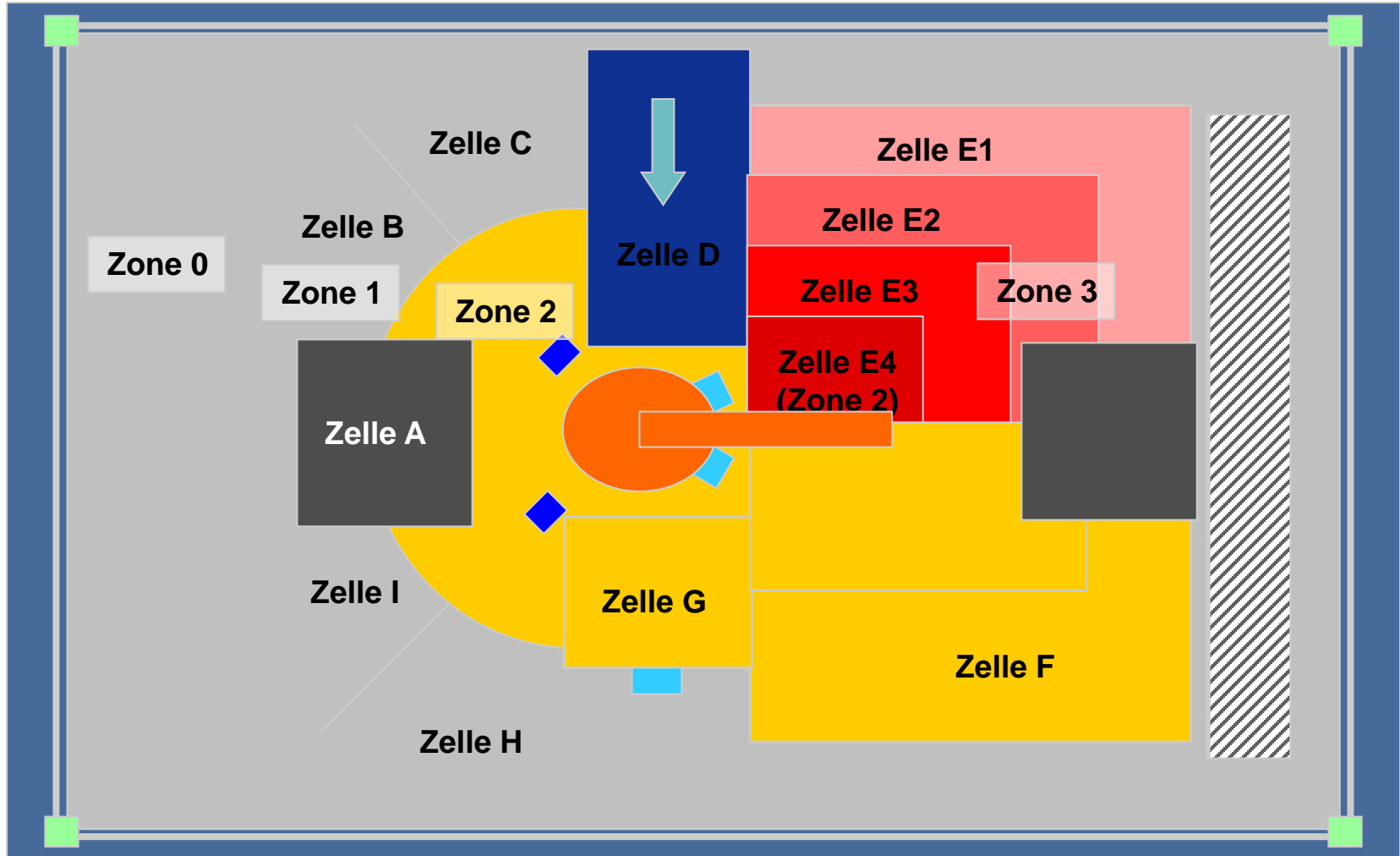
# Arbeitszelle: Sicherheitszonen



# Realisierung einer Ausweichbewegung

- Zelle  $E$  ist in vier Subzellen aufgeteilt (Zone 3)
  - Erfordert zusätzliche Überwachungsmöglichkeiten
  - Multiplexen der Subzellen auf Laserscanner, zirkuläre Überwachung
- Aktionen bei Verletzung der Sicherheitsbereiche
  - Zonen 0,1,2: Analog zu vorigem Beispiel
  - Zone 3: Ausweichbewegung (abhängig von verletzter Zelle) statt Stop-Befehl

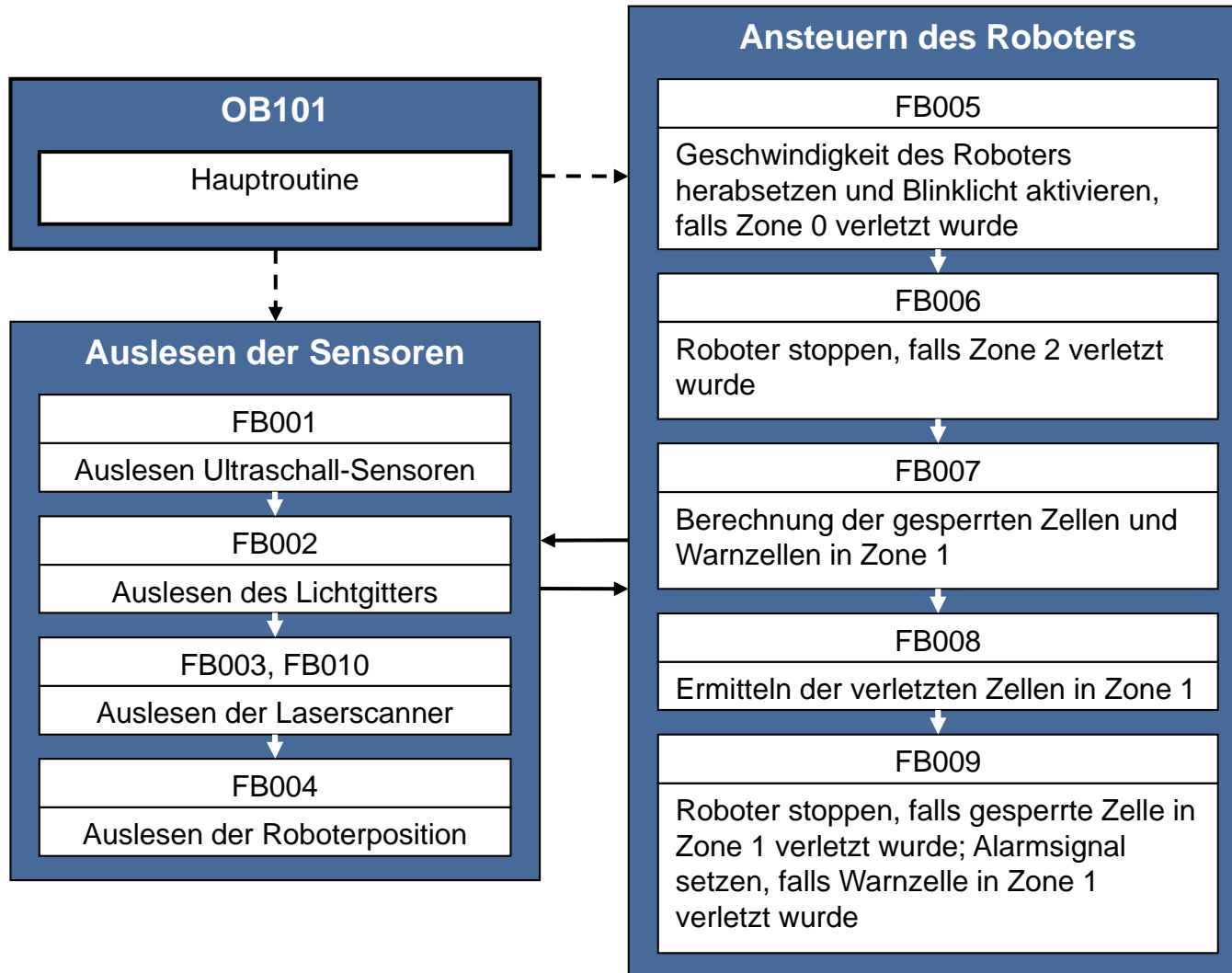
# Arbeitszelle: Sicherheitszonen



# Arbeitszelle: Sicherheitszonen



# Implementierung auf der SPS



# Testen des Systems

- Software: Bottom-up Implementierung, Black- und White-Box-Tests während des gesamten Entwicklungszyklusses
- Laserscanner, SPS und Lichtgitter: Keine intensiven HW-Tests, da zertifizierte Komponenten für Sicherheitsanwendungen
- Ultraschallsensoren
  - Schwachstelle in der Hardware
  - Mäßige Objekterkennung
  - Objekte mit stark reflektierenden Oberflächen konnten sicher erkannt werden (falls ausgesendeter Schall nicht vom Sensor weggespiegelt wird)
  - Objekte mit Schall schluckender Oberflächen werden nur mäßig gut detektiert



# Quantitative Aussagen zur Reaktionszeit

Versuch Nr.	Reaktionszeit Laserscanner (ms)	Reaktionszeit SPS (ms)	Reaktionszeit Roboterstrg (ms)	Nachlaufzeit Roboter (ms)	Summe (ms)
1	120	139	410	221	890
2	120	58	405	238	821
3	120	215	411	223	969
4	120	121	416	113	770
5	120	45	415	227	807
6	120	124	415	224	883
7	120	63	419	229	831
8	120	177	417	224	938
9	120	81	414	236	851
10	120	97	422	218	857
Minimum	120	45	405	113	770
Maximum	120	215	422	238	969
Median	120	109	415	224	854
Mittelwert	120	112	414,4	215,3	861,7
Mtl. Abw.	0	43,2	3,52	20,46	46,64

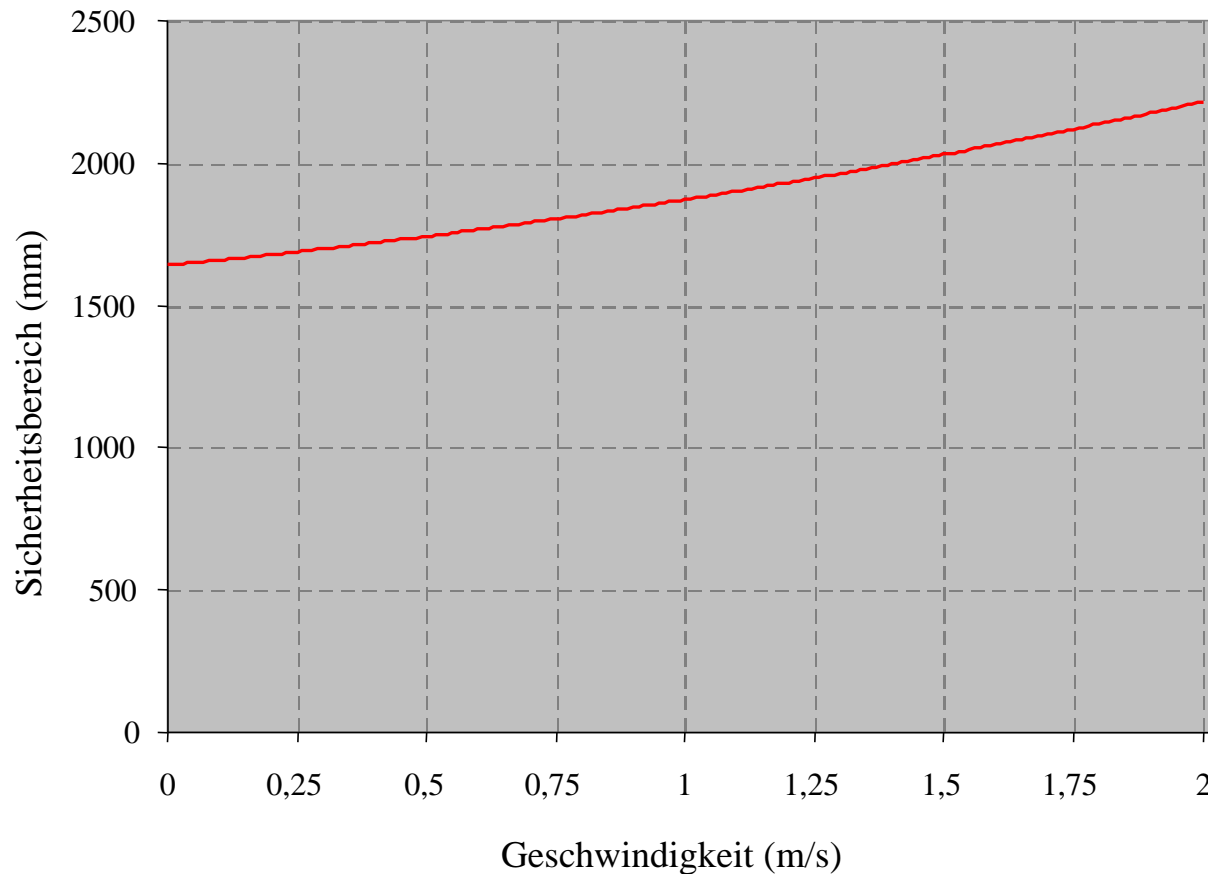
## Quantitative Aussagen

- Gesamtnachlaufzeit des Systems im Mittel 862 ms und im Worst-Case 1050 ms
- Reaktions-/Nachlaufzeiten
  - Laserscanners: 120 ms (Worst-Case 120 ms)
  - SPS: 112 ms (Worst-Case 270 ms)
  - Robotersteuerung: 414 ms (gem. Max. 422 ms)
  - Roboterantrieb: 215 ms (gem. Max. 238 ms)
- Erforderliche Sicherheitsabstand  $s = 4,6$  m im Worst-Case
$$s_{\text{sicherheit}}(v_{\text{roboter}}) = t_{\text{gesamt}} \cdot v_{\text{mensch}} + t_{\text{reaktion}} \cdot v_{\text{roboter}} + s_{\text{nachlauf}} + s_{\text{mindestabstand}}$$
- → Optimierungen erforderlich

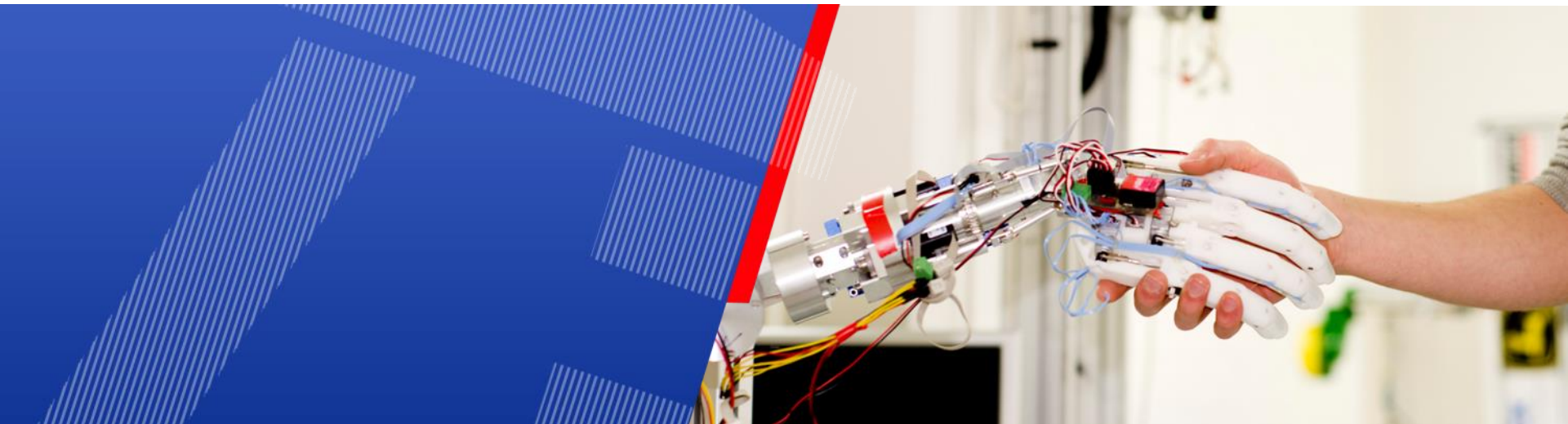
# Optimierung des Systems

- Weitere Laserscanner vermeiden Feldumschaltung  
→ Verminderung um 240 ms
- Direkte Steuerung statt Trap-Routine  
→ Verminderung um ca. 410 ms
- Optimierung
  - Ansprechzeit reduziert auf 400 ms
  - Abstand reduziert auf 2,2 m

# Allgemeine Berechnung d. Sicherheitsbereichs



# Zusammenfassung der Vorlesung



**Prof. Karsten Berns**

Robotics Research Lab

Department of Computer Science

University of Kaiserslautern, Germany

# Themen

- Teilsysteme
- Raumkinematik
- Robotermodellierung
- Bahnsteuerung
- Endeffektoren und Greiferplanung
- Planungssysteme
- Steuerungsarchitekturen
- Programmierung von Robotersystemen

# Teilsysteme und Raumkinematik

- Teilsysteme
  - Mechanische Komponenten
  - Gelenktypen
  - Grundkonfiguration für Roboter
  - Arbeitsraum
  - Antriebe, Sensoren, Regelung
- Raumkinematik
  - Beschreibung von Objekten und Objektlagen im 3D euklidischen Raum ( $E^3$ )
  - Orientierungsbeschreibung mit  $3 \times 3$  Matrizen
  - 6-dim. Beschreibungsvektor
  - Homogene Koordinaten und Transformationsmatrix
  - Verkettete Lagebeschreibungen



# Modellierung und Bahnsteuerung

- Robotermodellierung
  - Freiheitsgrade
  - Geometrisches Modell
  - Kinematisches Modell
  - Direktes Kinematisches Problem
  - Inverses Kinematisches Problem
  - Dynamisches Modell
- Bahnsteuerung
  - Grundlagen der Bahnsteuerung
  - Interpolationsarten
  - Bahnsteuerung
  - Spline-Interpolation

# Greif- und Planungssysteme

- Endeffektoren und Greifplanung
  - Grundlagen
  - Grifffhierarchie
  - Klassifikation von Greifplanungssystemen
  - Planung von Umgreifoperationen
  - Szenenstabilität
- Planungssysteme
  - Grundlagen für Roboterplanung
  - Planungsarten
  - Planen als Suche
  - Der Cranfield-Montage-Benchmark

# Steuerungsarchitekturen und Programmierung

- Steuerungsarchitektur
  - Grundlegende Fähigkeiten
  - Schematische Darstellung der 4 Basisarchitekturen
  - Hierarchisch funktionsorientierte Architekturen
  - Verteilte funktionsorientierte Architekturen
  - Hierarchisch verhaltensorientierte Architekturen
  - Verteilt verhaltensorientierte Architekturen
- Programmierung
  - Programmierung von Industrierobotern
  - Online-/Offline-Verfahren
  - Arten der Programmierung
  - Umweltmodellierung

## Kommentare zur Vorlesung

- Was war gut/schlecht?
- Inhaltliche Unklarheiten?
- Struktur?
- ...
- Vorlesungsumfrage der Fachschaft (<https://vlu.cs.uni-kl.de>)

# Vielen Dank

