# SAMS[1]
# Sicherheitskomponente für Autonome Mobile Serviceroboter


Dr. **Christoph Lüth**, Dr. **Udo Frese, Holger Täubig**, **Dennis Walter**
Deutsches Forschungszentrum für Künstliche Intelligenz,
Fachbereich Sichere Kognitive Systeme, Bremen

**Daniel Hausmann**, Universität Bremen

**Kurzfassung**

Ziel des Projektes *SAMS* ist die Entwicklung einer zulassungsfähigen Fahrwegsicherung für Serviceroboter und fahrerlose Transportsysteme (FTS), die mit einem zertifizierten Sicherheitslaserscanner die überwachte Sicherheitszone dynamisch dem Zustand des Fahrzeugs anpasst. Während dies für Forschungsprototypen Stand der Technik ist [4, 5], konzentriert sich das Projekt *SAMS* auf eine zulassungsfähige Entwicklung [2], die durch ein TÜV-Gutachten nachgewiesen werden soll. Kernvorhaben ist dabei die formale mathematische Modellierung und der Korrektheitsbeweis der Implementierung.

## 1. Einleitung

Sicherheit vor Kollisionen zu gewährleisten ist eine gesetzliche Voraussetzung für die Zulassung jeden Serviceroboters. Bei industriell eingesetzten fahrerlosen Transportsystemen (FTS) überwacht ein Laserscanner als zugelassenes Sicherheitsbauteil ein Schutzfeld um das Fahrzeug, und leitet Maßnahmen zum sicheren Halt des Fahrzeugs ein, sobald sich ein Hindernis im Schutzfeld befindet. Die mangelnde Flexibilität und eng begrenzte Anzahl solch fester Zonen schränkt Bahnführung und Geschwindigkeit unnötig ein, und verhindert in vielen Fällen eine effiziente und kostengünstige Bahnführung. In dem Projekt SAMS wird eine zulassungsfähige Komponente für die sichere Servicerobotik entwickelt, die einen mobilen Roboter durch Auswertung der Entfernungsmessdaten eines Sicherheits-laserscanners rechtzeitig vor einer Kollision mit einem Hindernis stoppt. Die Lösung soll als eigenständiges Gerät entwickelt werden, um so Produzenten von Servicerobotern in die Lage zu versetzen, eine Lösung für die Sicherheitsproblematik einzukaufen und die Eigenentwicklung auf Anwendungsfragen zu konzentrieren. Entscheidend ist die angestrebte Sicherheitszulassung, und die dafür nötige formale Modellierung und der formale Nachweis der Korrektheit der Implementierung.

[1]Gefördert vom BMBF im Rahmen der Leitinnovation „Servicerobotik“ , Förderkennzeichen 01-IM-F02.

## 2. Systemarchitektur

Die Sicherheitskomponente ist als eigenständige Komponente konzipiert, die ausgehend von dem aktuellen Zustand, gegeben durch den Geschwindigkeitsvektor sowie die Messdaten des Laserscanners, berechnet, ob ein Nothalt eingeleitet werden muss, um eine Kollision zu vermeiden. Diese Kernfunktionalität wird als *Fahrwegsicherung* bezeichnet.

Die Komponente wird vor Inbetriebnahme mit dem *Bremsmodell* des autonomen Roboters parametriert, welches für einen gegebenen Geschwindigkeitsvektor die bei einer Notbremsung bei dieser Geschwindigkeit überstrichene Bremsfläche angibt (Konfigurationsphase). Mittels des Konfigurationsprogramms, das auf einem handelsüblichen PC läuft**,** gibt der Benutzer die Geometrie des Roboters an, auf welchem die Komponente betrieben werden soll, sowie das Bremsmodell (s. unten). Für jeden Geschwindigkeitsvektor wird die bei der Bremsung mit dieser Geschwindigkeit überstrichene Fläche berechnet. Nach Abschluss der Parametrierung werden die Daten an die Sicherungskomponente übertragen.

Zur Laufzeit werden die Eingabedaten der Komponente (je ein Odometriesensor für das linke und rechte Rad per Quadraturzähler, welche zusammen den Geschwindgkeitsvektor ergeben, und die Messdaten des Laserscanners über eine Ethernetschnittstelle) mit der vorberechneten Bremsfläche für diese Geschwindigkeit verglichen, und wenn eine Kollision droht, wird auf einem sicheren Ausgang ein *Nothaltsignal* ausgelöst (diese Situation wird in Abb. 1 dargestellt).

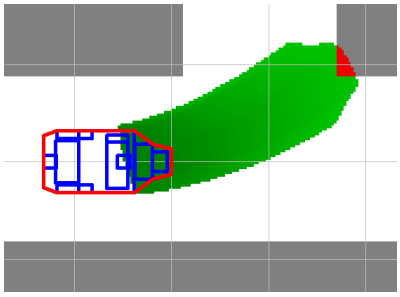

*Abb. 1: Schutzfeldverletzung*

Die Komponente kann ferner ein Warnsignal generieren, welches bevorstehende Notbremsungen ankündigt, so dass diese im Vorfeld verhindert werden können. Als Hardwareplattform dient ein doppelt ausgelegtes Board auf Basis des netX-100.

## 3. Ermittlung des Bremsmodells

Das Bremsmodell beschreibt für jeden Geschwindigkeitsvektor die Fläche, die beim Abbremsen aus dieser Geschwindigkeit bis zum Stillstand des Fahrzeuges überstrichen wird. Eine erste Grundannahme des verwendeten Bremsmodells ist, dass der Lenkwinkel des Fahrzeuges während des Bremsvorganges konstant bleibt, d.h. dass die Ortskurve der Bremsbewegung ein Kreisbogen ist. Die Länge dieses Kreisbogens wird durch zwei Abschätzungen ermittelt, die einerseits konservativ sind und somit die Sicherheit des Gesamtsystems gewährleisten, es andererseits aber auch ermöglichen, das Bremsmodell bereits mit nur einem Messwert vollständig zu parametrieren.

Die erste Abschätzung berechnet dabei den Bremsweg für Kurvenfahrten aus dem Bremsweg für Geradeausfahrt. Hierzu wird eine energetische Betrachtung durchgeführt: es wird angenommen, dass der Bremsweg proportional zur dabei abgebauten kinetischen Energie ist, und dann die kinetische Energie bei Kurvenfahrt mit der in Geradeausfahrt in Verhältnis gesetzt, wobei sich schwer messbare Größen wie Rotationsträgheit durch vereinfachende Annahmen wie eine konstante Massedichte herauskürzen lassen.

Im zweiten Schritt wird der Bremsweg **s**(**v**) bei Geradeausfahrt aus einer oder mehreren Messungen stückweise linear approximiert. Aufgrund der physikalischen Eigenschaften des Bremsvorganges ist **s**(**v**) konvex und die stückweise lineare Abschätzung damit eine obere Schranke des Bremsweges. Die Bremswege können so entweder mit wenig Aufwand durch nur eine Bremswegmessung für die maximale Geschwindigkeit oder beliebig genau durch eine größere Anzahl von Messwerten abgeschätzt werden (praktisch reichen meist zwei Messwerte aus), wobei jede Möglichkeit nachweislich konservativ ist.

**4. Formale Korrektheit**

Ein wesentlicher Bestandteil der Erfüllung normativer Anforderungen bezüglich der Sicherheit ist der Nachweis der Korrektheit der für den Betrieb der Sicherungskomponente nötigen Software [2]. Die Verifikation umfasst dabei zwei wesentliche Eigenschaften:

- Die Schutzfelder im Konfigurationsprogramm werden korrekt berechnet,
- und im laufenden Betrieb wird ein Schutzfeld korrekt ausgewählt und überwacht.

Da die verwendete Software komplex und die generierten Daten sehr zahlreich sind (über 5000 Schutzfelder) kann durch bloßen Code Review und stichprobenartiges Testen keine hinreichende Zuverlässigkeit der Software garantiert werden. Die im Projekt entwickelte Verifikationsumgebung, die auf dem Theorembeweiser Isabelle [1] aufbaut, liefert hingegen eine wesentlich stärkere Zuverlässigkeitsaussage und damit Software-Sicherheitsintegrität. Die Verifikationsumgebung besteht aus

- einem formalisierten Domänenmodell, welches Geometrie und Physik von zweidimensionalen Objekten in der Ebene (Bewegung, Bremswege etc.) umfasst;
- einer Spezifikationssprache, mit der sich Korrektheitsaussagen über Programmfunktionen treffen lassen;
- sowie einem Werkzeug, das diese Aussagen teils automatisch beweist, und teils als Verifikationsbedingungen zum interaktiven Beweis an den Anwender weitergibt.

```
/*@ requires  valid_array(v, start, len) &&
              0 <= start && 0 < len
  @ ensures   (\forall int i; start < i && i < start + len -->
                              v[i-1] <= v[i]) &&
              is_permutation(\old(v), v, start, len)
  @*/
void sort(int *v, int start, int len)
{ ... }
```

*Abb. 2: Spezifikation einer Sortierfunktion*

Abb. 2 zeigt eine C-Funktion mit ihrer Spezifikation. Spezifikation und Programmtext sind in unserem Ansatz eng miteinander verknüpft. Ein Korrektheitsbeweis gilt somit immer für konkrete Funktionen und deren Spezifikationen, was eine Korrumpierung der Sicherheit durch nachträgliche Änderungen verhindert. Ferner prüft die Verifikationsumgebung die Einhaltung der internen, am MISRA-C Standard [3] orientierten Programmierrichtlinien. Die Verifikationsumgebung unterliegt neben der Software der Sicherungskomponente ebenfalls der Begutachtung durch den TÜV.

**5. Zusammenfassung**

Sicherheit ist eines der dringendsten Anwendungsprobleme in der Servicerobotik: vor dem Einsatz beim Endanwender liegt die Hürde einer sicherheitstechnischen Zulassung (z.B. durch den TÜV). Das Projekt SAMS will für die Fahrwegsicherung diese Hürde nehmen, indem eine formal korrekt verifizierte, separat einsetzbare Komponente zur Kollisionsvermeidung entwickelt und zertifiziert wird, die Endanwendern und Entwicklern von Servicerobotern *safety in a box* bietet.

Die Laufzeit des Projektes ist von 2006 bis 2008; momentan ist ein Prototyp der Fahrwegsicherung implementiert, und die Verifikationsumgebung im Aufbau.

**5. Literatur**

[1] T. Nipkow, L. C. Paulson, M. Wenzel: *Isabelle/HOL-- A Proof Assistant for Higher-Order Logic.* LNCS 2283, Springer 2002.

[2] DIN EN 61508, *Funktionale Sicherheit sicherheitsbezogener elektrischer/ elektronischer/ programmierbarer elektronischer Systeme*.

[3] The Motor Industry Software Reliability Association, *MISRA-C:2004 Guidelines for the use of the C language in critical systems*.

[4] A. Lankenau, T. Röfer (2001): A Safe and Versatile Mobility Assistant. Reinventing the Wheelchair. IEEE Robotics and Automation Magazine, 8 (1), S. 29 - 37.

[5] D. Fox, W. Burgard, S. Thrun (1997): The Dynamic Window Approach to Collision Avoidance. IEEE Robotics and Automation Magazine, 4 (1), S. 23-33.