

Schlüsselkomponenten für die Exploration mit teil-autonomen mobilen Robotern

Rainer Worst, Hartmut Surmann, Kai Pervölz und Marco Hartich

Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
Schloss Birlinghoven, 53754 Sankt Augustin
rainer.worst@iais.fraunhofer.de
WWW home page: <http://www.iais.fraunhofer.de/>

Zusammenfassung. Mobile Roboter finden immer weiteren Einsatz für verschiedenartige Erkundungsaufgaben. Diese Systeme werden meist von einem Bediener ferngesteuert. Auf der anderen Seite werden ständig bessere Verfahren entwickelt, um mobile Roboter autonom operieren zu lassen. Die am Fraunhofer IAIS entwickelten Schlüsselkomponenten 3D-Sensorik und Hybrid-Steuerung ermöglichen durch die Verbindung von Autonomie und Tele-Operation die Entwicklung von Erkundungsrobotern, die optimal an ihre Aufgaben angepasst werden können.

1 Einleitung

Mobile Roboter gewinnen weltweit zunehmend an Bedeutung, insbesondere bei einer Vielzahl anstehender Mess-, Erkundungs- und Bewachungsaufgaben. Ein typisches Anwendungsszenario stammt aus dem Bereich Urban Search And Rescue (USAR). Mobile Roboter sollen dabei helfen, eine unbekannte und unzugängliche Umgebung zu erkunden, z. B. Hohlräume in Gebäuden, die durch ein Erdbeben eingestürzt sind [1]. Dieses Szenario ist auch Grundlage des bekannten RoboCup Rescue Wettbewerbs. Das Ziel besteht hier darin, Opfer in den Gebäuden zu finden und ihre Position sowie ihre Befindlichkeit zu bestimmen.

Fraunhofer IAIS hat in den Jahren 2004 bis 2006, allein oder mit Partnern, an diesem Wettbewerb teilgenommen [2]. Die dabei gesammelten Erfahrungen sind in die Entwicklung von Komponenten eingeflossen, die für Explorationsroboter benötigt werden – unabhängig von deren konkretem Einsatzzweck. Derartige Roboter werden heute meistens ferngesteuert und nur selten autonom betrieben. Systeme, die beide Betriebsarten erlauben, bieten jedoch folgende Vorteile gegenüber reiner Tele-Operation:

1. Steuerung wird durch autonome Assistenzsysteme erleichtert.
2. Sicherheitsmechanismen aus dem autonomen Betrieb können auch bei Fernsteuerung aktiv sein.
3. Bediener werden von Routineaufgaben entlastet.
4. Betrieb mehrerer Roboter durch einen Bediener als Koordinator ist möglich.
5. Roboter kann bei Funkabbruch von Fernsteuerung auf Autonomie umschalten und dadurch selbstständig in einen sicheren Zustand übergehen.

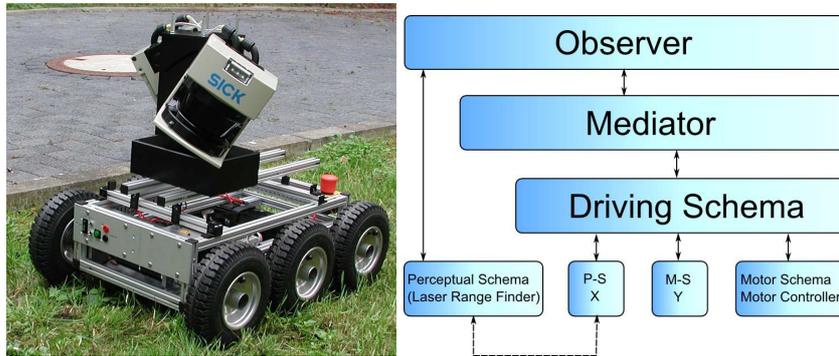


Abb. 1. Links: Explorationsroboter mit 2 kontinuierlich drehenden Laserscannern. Rechts: Schemabasierte Darstellung der Steuerungs-Software

Dieses Papier skizziert kurz die verschiedenen Schlüsselkomponenten mit den entsprechenden Verweisen auf die Literatur; es beschreibt keine Roboterarchitektur.

2 3D-Sensorik für Kartenbau und Navigation

3D-Kartierung mittels Robotern und 3D-Laserscannern findet zur Zeit das Interesse verschiedener Forschergruppen, weil diese Vorteile gegenüber den häufig eingesetzten 2D-Scannern bieten, z. B. beim Erkennen von überstehenden Objektkanten, wie sie an Tischen vorkommen.

Die am Fraunhofer IAIS entwickelten zwei Varianten (kontinuierlich drehend oder nickend) des 3D-Laserscanners 3DLS [3] erzeugen 3D-Punktwolken. Wird der Scanner auf einer mobilen Plattform (Volksbot oder KURT2) montiert, so generiert das System kontinuierlich bzw. von Fall zu Fall Punktwolken, die den mit dem Roboter abgefahrenen Bereich dreidimensional abbilden (Abb. 1). Die Punktwolken werden mit einer schnellen Variante des ICP-Algorithmus auf der Basis der Roboterposenschätzung registriert und so zu einer 3D-Karte der befahrenen Umgebung zusammengesetzt. In der zusammengesetzten Punktwolke können Linien und Flächen detektiert, sowie einzelne Bereiche, wie Boden, Wand oder Decke, klassifiziert werden. Es wird weiterhin versucht, diese klassifizierten Bereiche mit geometrischen Formen zur Deckung zu bringen [4,5].

Die Selbstortung des Roboters ist sowohl bei Tele-Operation als auch bei autonomem Betrieb von großer Bedeutung. Die oben erwähnte Registrierung einzelner Punktwolken ermöglicht eine verbesserte Selbstortung des Roboters durch SLAM-Algorithmen (Simultaneous Localization And Mapping) [6]. Sie erfolgt kontinuierlich und wird in gewissen Intervallen rekaliibriert, z. B. immer dann, wenn der Roboter anhält, um eine neue 3D-Aufnahme zu erzeugen. Für den autonomen Betrieb lassen sich bekannte und bewährte 2D-Steuerungs-

Lokalisierungs- und Kartierungsalgorithmen verwenden, wenn man aus der 3D-Szene eine 2D-Projektion generiert, die mehrere horizontale Ebenen enthält. Der Vorteil dieser sogenannten virtuellen Scans gegenüber der direkten Benutzung eines 2D-Laserscanners ist hier die Berücksichtigung überhängender Hindernisse und die daraus folgende Vermeidung von Kollisionen, etwa mit Tischen [2].

Der Explorationsroboter liefert als Ergebnis eine 3D-Karte der erkundeten Umgebung. Die gewonnenen Daten werden so dargestellt, dass der Bediener sich ein Bild von der Situation machen kann. Der Bediener kann das Bild sowohl während der Exploration (online) als auch im Nachhinein (offline) auswerten. Wichtig ist hierbei, dass er die Perspektive auswählen kann, aus der er die Szene betrachtet. Während normalerweise die Darstellung aus Sicht der Plattform den gewünschten Eindruck vermitteln wird, ist es für die Fernsteuerung des Roboters hilfreich, auch einmal die Perspektive eines externen Beobachters einzunehmen und so die Plattform in ihrer Umgebung zu sehen. Eine (zusätzliche) Grundrissdarstellung ist hier unter Umständen hilfreicher als eine reine 3D-Karte.

Bei der offline-Auswertung gibt es zusätzlich die Möglichkeit, bessere aber rechenzeit-intensivere Registrierungsverfahren zu verwenden und die Daten mittels einer speziellen Viewer-Software zu betrachten, mit der man die Szene beliebig drehen oder in sie hinein- und aus ihr herauszoomen kann. Auch das virtuelle Wandern oder Fliegen durch die Szene trägt zum besseren Begreifen der dargestellten Situation bei.

3 Grundlagen der Hybrid-Steuerung

Bediener, die Roboter steuern, sind oftmals schon nach kürzester Zeit erschöpft, wenn sie für die Steuerung des Roboters und die Überprüfung der gewonnenen Daten verantwortlich sind. Dies belegt eine Studie, die von Murphy nach dem Attentat vom 11. September durchgeführt wurde [7]. Die angegebenen Gründe sind zum einen primitive Benutzerschnittstellen und zum anderen unzureichende Funktionalitäten der Roboter. So wird es heute als sinnvoll angesehen, zwei Personen zur Steuerung heranzuziehen, und zwar einen Bediener, der steuert, und einen Supervisor, der die Daten auswertet und ggf. Befehle gibt. Deren Entlastung, die Reduktion auf nur einen Bediener oder aber technische Probleme wie Funkabbrüche erfordern weitere unterstützende Funktionen, wie beispielsweise das automatische Ausweichen vor Hindernissen. Diese verschiedenen Möglichkeiten der Robotersteuerung werden von Wegner [8] beschrieben. Er beleuchtet die Steuerungsproblematik speziell im Hinblick auf den Einsatz von Robotern im Bereich von USAR-Anwendungen und definiert dazu einen Schema-basierten Ansatz, der im Wesentlichen darauf beruht, dass durch die einzelnen Schemas ein gewichteter Vektor folgender Form generiert wird:

$$\bar{a} = \begin{bmatrix} \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} M \\ \Theta \end{bmatrix} \quad (1)$$

wobei M der Amplitude, Θ der Ausrichtung und ϕ einer Gewichtung entspricht. Eine Priorisierung findet in der Form statt, dass jedes Motorschema seinen eigenen Vektor generiert. Der Parameter ϕ kann in einer Hybridsteuerung dazu

genutzt werden, ein bestimmtes Verhalten zu priorisieren. Um die Autonomie und die Tele-Operation zu verschmelzen, stellt Wegner zwei Vorgehensweisen vor:

1. Eingriffs- und Bestätigungsverhalten (Intervention Recognition)
2. Vermittler (Mediator)

Steckt der Roboter fest, weiß er nicht mehr weiter oder hat einen Zielpunkt erreicht, so meldet er dies dem Bediener (Eingriffs- und Bestätigungsverhalten). Die Mischung aus Autonomie und Tele-Operation übernimmt der Mediator (Gewichtung durch ϕ). Dieser wägt ab, ob eine vom Operator veranlasste Aktion durchgeführt werden soll oder nicht, z. B. im Falle von drohender Gefahr für den Roboter (Fahrt auf eine Wand zu).

Große Verzögerungszeiten, Verbindungsabbrüche oder die Vernachlässigung der Kontrolle (z. B. bei multi agent navigation) müssen kompensiert werden. Je länger die Zeitverzögerung ist, desto mehr Autonomie muss der Roboter aufweisen. Im Experiment von Crandall und Goodrich [9] werden drei Verhalten verwendet: Zielgesteuertes Verhalten, Hindernissen ausweichen und ein widersprechendes Verhalten. Zudem definieren sie in ihrem Projekt drei Schemas:

1. Komplette teleoperiert
2. Vorgeben von Wegpunkten
3. Komplette autonom

Das Vorgeben von Wegpunkten stellt hierbei eine wichtige Verbindung zwischen der direkten Teleoperation und dem autonomen Fahren da. Im einfachsten Fall werden die Punkte durch den Bediener vorgegeben. Im Außenbereich können Wegpunkte automatisch durch ein Navigationssystem erzeugt und mittels GPS-Sensoren abgefahren werden. Im Innenbereich werden sogenannte Trajektorien mit Hilfe von Voronoi-Diagrammen basierend auf gegebenen Karten und Laserabstandsdaten erzeugt (Abb. 2). Für Verbindungsgeraden von Anfangs- und Endpunkten unterschiedlicher Linien wird auf der Hälfte eine Senkrechte konstruiert und mit den anderen Senkrechten verbunden. Die Schnittpunkte dieser Mittelsenkrechten schneiden sich an den Thiessen-Scheitelpunkten, deren Verbindung dann das Thiessen- bzw. Voronoi-Polygon ergibt. Der gesuchte Trajektorie läuft entlang der Scheitelpunkte. Im Laserscan wahrgenommene Hindernisse werden somit automatisch umfahren, wobei das Verfahren mit der GPS-Navigation im Außenbereich kombinierbar ist.

Eine weitere wichtige Komponente ist dann das Abfahren der Wegpunkte. De Wit et al. [10] aber auch Indiveri et al. [11] beschreiben geschlossene, zeitinvariante und global-stabile Regelungen, um durch Punkte vorgegebene Trajektorien abzufahren. Abb. 2 links zeigt ein abzufahrendes Dreieck und die tatsächlich gefahrene Strecke. Mit Hilfe eines Roboter- oder Fahrzeugmodells können die Fahrtrajektorien vorberechnet und dem Bediener zusätzlich zu den Solltrajektorien angezeigt werden.

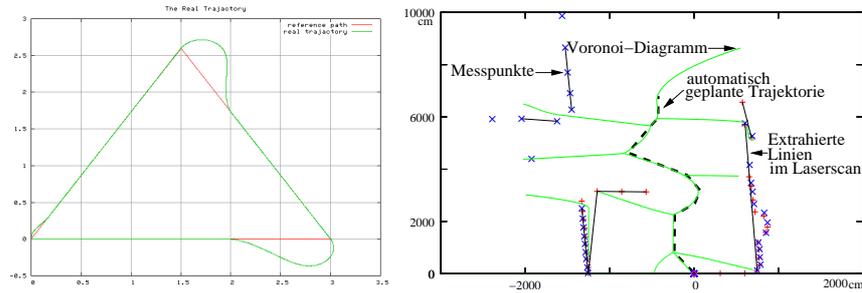


Abb. 2. Links: Fahrt nach Wegepunkten (Dreieck) mit dem Verfahren nach [11]. An den Eckpunkten fährt der Roboter auf einer Kurve zurück auf den Referenzweg. Rechts: Beispiel von selbständig geplanten Wegepunkten in einem Laserscan mit Hilfe eines Voronoi-Diagramms.

4 Verbindung von Autonomie und Tele-Operation

Mit der am Fraunhofer IAIS entwickelten grafischen Benutzerschnittstelle ist es möglich, einen einstellbaren Grad an Autonomie und Teleoperation zu realisieren. Die Berechnung des resultierenden Vektors wird ähnlich wie bei Wegner [8] mittels einer Vektoraddition ermittelt.

Für die grafische Benutzerschnittstelle (Abb. 3) wurde Java-Webstart verwendet. Dadurch ist es möglich, die Benutzerschnittstelle als Applikation von einem Webserver durch Anklicken eines Hyperlinks zu laden. Der Unterschied zu einem herkömmlichen Java-Applet ist, dass man zum einen nicht mehr auf den Browser angewiesen ist und zum anderen aus der Sandbox einer Applet-Applikation heraussteigen kann. Dies ist nötig, um z. B. auf dem Rechner des Bedieners einen Joystick nutzen zu können.

Um die Steuerung mittels der grafischen Benutzerschnittstelle so einfach wie möglich zu halten, wurde deren umfangreiche Funktionalität in Menüs verlagert. Bediener werden dadurch ausschließlich bei der Konfiguration des Roboters mit dessen Einstellungen konfrontiert. Während des Einsatzes können sie sich somit voll und ganz auf die Daten und die Steuerung selbst konzentrieren. Die grafische Benutzerschnittstelle wirkt damit nicht überladen. Bei Bedarf ist die Konfiguration des Roboters (Auswahl neuer Verhalten, Bearbeiten von Trajektorienpunkten, Gewichtung der einzelnen Parameter usw.) auch zur Laufzeit möglich.

Die Kommunikation zwischen grafischer Benutzerschnittstelle und Roboter wurde mittels SOAP umgesetzt. Ein Vorteil dieses Protokolls liegt in der einfachen Erweiterbarkeit. Funktionalitäten des Roboters können ohne Änderung des Protokolls zügig hinzugefügt oder entnommen werden. Weiter ist es bei der Verwendung von SOAP zusammen mit Java-Webstart, möglich eine Steuerung über das Internet zu realisieren, wobei natürlich die unterschiedlichen Laufzeiten im Internet berücksichtigt werden müssen.

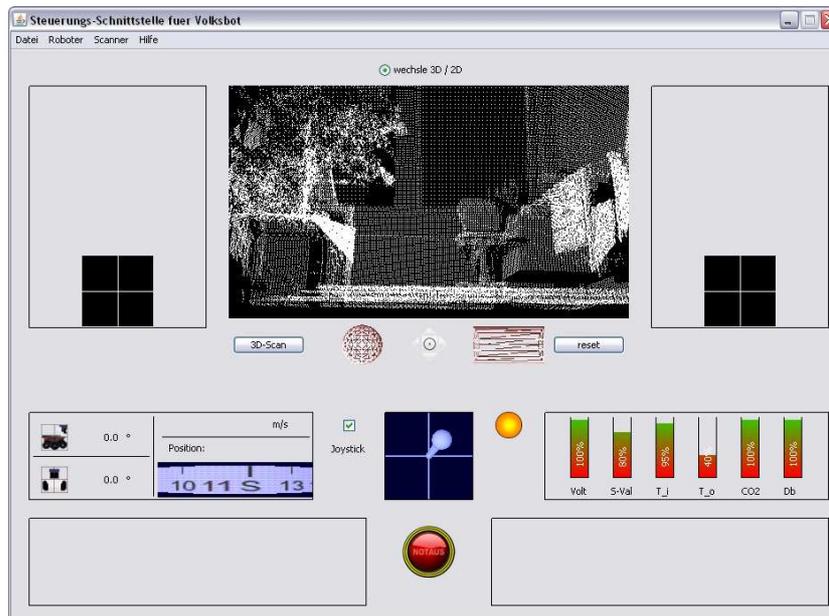


Abb. 3. Muster der grafischen Benutzerschnittstelle

Um eine Kommunikationsunterbrechung detektieren zu können, wurde ein Pollingverfahren implementiert. Der Client (grafische Benutzerschnittstelle) sendet, sofern keine anderen Daten – wie etwa Joystick-Kommandos – anstehen, ständig Polling-Informationen an den Server (Roboter). Das Verhalten bei Verbindungsabbruch wird im Voraus auf Seiten des Clients definiert. Das einfachste Verhalten ist z. B. ein Stopp-Kommando. Wurde die Verbindung wieder etabliert, so wird dem Client die Information des erkannten Verbindungsabbruchs gemeldet und der normale Betrieb wird nach Bestätigung durch den Client wieder eingerichtet. Das Bereitstellen dieses Dienstes erfolgt über den Observer (Abb. 1).

Der Mediator in Abb. 1 übernimmt die Auswahl der einzelnen Verhalten. Dazu liest dieser zu Beginn die Konfiguration des Roboters aus einer auf dem Roboter befindlichen Datenbank aus. In dieser Datenbank ist definiert, wie der Roboter gesteuert wird, mit welcher Gewichtung die einzelnen Verhaltensweisen in die Steuerung einfließen und was nach einem Verbindungsabbruch erfolgen soll. Mögliche Verhaltensweisen sind hier z. B. verschiedene Arten der Hindernisvermeidung oder ein zielgerichtetes Verhalten. Während der Laufzeit prüft der Mediator, ob eine Rekonfiguration seitens des Clients vorgenommen wurde. Ist dies der Fall, stoppt der Mediator alle laufenden Schemas, liest die neuen Konfigurationsdaten aus der Datenbank und übergibt diese dem Driving-Schema. Detektiert der Observer einen Verbindungsabbruch, so wird dieses Ereignis vom

Observer an den Mediator gemeldet. Der Mediator kann dann das laufende Verhalten beenden und das vorgegebene Verhalten für einen Verbindungsabbruch starten.

Aufbauend auf diesem neuen System ist es möglich, schon bestehende Verhaltensweisen mittels des Mediators zu verwalten. Diese müssen evtl. noch an das System angepasst werden.

Danksagung

Wir danken unseren Partnern von den Universitäten Osnabrück und Hannover im Team Deutschland1 für die fruchtbare Zusammenarbeit bei der Vorbereitung und der Teilnahme am RoboCup Rescue 2005 und 2006.

Literaturverzeichnis

1. Nüchter, A., Lingemann, K., Hertzberg, J., et al.: Mapping of Rescue Environments with KURT3D. Proc. of the Int. Workshop on Safety, Security and Rescue Robotics (SSRR'05), Kobe, Japan (2005) 158–163
2. Nüchter, A., Lingemann, K., Hertzberg, J., et al.: The RoboCup Rescue Team Deutschland 1. KI - Künstliche Intelligenz 2/06 (2006) Böttcher IT, 24–29
3. 3DLS: www.3d-scanner.net; Volksbot: www.volksbot.de; KURT2: www.kurt2.de
4. Nüchter, A., Surmann, H., Hertzberg, J.: Automatic Classification of Objects in 3D Laser Range Scans. Proc. of the 8th Conf. on Intelligent Autonomous Systems (IAS'04), IOS Press, Amsterdam, The Netherlands (2004) 963–970
5. Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H.: Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM. Proc. of the Int. Workshop on Safety, Security and Rescue Robotics (SSRR'06), Gaithersburg, Maryland, USA (2006)
6. Surmann, H., Nüchter, A., Lingemann, K., Hertzberg, J.: 6D SLAM – Preliminary Report on closing the loop in Six Dimensions. Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'04), Lisbon, Portugal (2004)
7. Murphy, R. R.: Human-Robot Interaction in Rescue Robotics. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 34 (2004) 138–153
8. Wegner, R.: Balancing Robotic Teleoperation and Autonomy in a Complex and Dynamic Environment. Master thesis, University of Manitoba (2003).
9. Crandall, J. W., Goodrich, M. A.: Experiments in Adjustable Autonomy. IEEE Int. Conference on Systems, Man and Cybernetics, Vol. 3 (2001) 1624–1629
10. de Wit, C. C., Khennouf, H., Samson, C., Sordalen, O. J.: Nonlinear Control Design for Mobile Robots. In Yuan F. Zheng, editor, Recent Trends in Mobile Robots, World Scientific Series in Robotics and Automated Systems (1993) 121–156
11. Indiveri, G., Corradini, M. L.: Switching linear path following for bounded curvature car-like vehicles. Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'04), Lisbon, Portugal, (2004)