



Technische Universität Wien

Department für Geodäsie und Geoinformation
Forschungsgruppe Ingenieurgeodäsie
Gusshausstraße 27-29 / E120-5
A-1040 Wien
Tel.: +43 (1) 58801 - 12804, Fax: +43 (1) 58801 - 12894

Projekt

Seekur Jr

Titel

Setup & Simulation zur Steuerung

Autor:	Tomas Thalmann
Kontakt:	tthalmann@gmail.com
Datum:	06.05.2014
Dokument:	MA-TH/14S/1
Datei:	MA-TH-2014S-1.docx
Version:	0.1
Status:	Draft

Kurzfassung:

Als Vorbereitung zur Masterarbeit an der Forschungsgruppe Ingenieurgeodäsie wurde das Robotersystem von MobileRobots Modell Seekur Jr untersucht. Ziel ist es die notwendigen Schritte zur Inbetriebnahme des Roboters zu dokumentieren und die Möglichkeiten zur Steuerung zu analysieren.

Dieser Bericht soll sowohl einen detaillierten Einblick in das Steuerungs-Modell und die mitgelieferten Steuerungs-Schnittstellen bieten als auch eine mögliche Gestaltung eines Steuerprozesses aufzeigen.

Außerdem wurde eine Simulationsumgebung entworfen, die es ermöglicht den Roboter inklusive Boardcomputer am PC zu simulieren.

1. Setup

1.1. Spezifikationen

Der *Seekur Jr* wurde von adept MobileRobots¹ so konzipiert, dass er sowohl in Gebäuden als auch im Freien betrieben werden kann. Er ist nach IP-54 wasser-resistent, kann ohne weiteres bei leichtem Regen und Schneefall betrieben werden, und ist somit auch vor Staub geschützt. Sowohl mit leichtem steinigem Gelände als auch auf schlammigen Untergrund ist der *Seekur Jr* fahrtüchtig. Die wichtigsten technische Spezifikationen sind in Tabelle 1 und Abbildung 1 ersichtlich.

Dimensionen:

Gewicht:	70kg
Breite:	83,5cm
Höhe:	49,4cm
Länge:	119,8cm

Überwindbare Hindernisse:

Max. Stufenhöhe:	ca. 20cm
Max. Steigung:	70%
Max. Wassertiefe:	12cm

Fortbewegung:

Max. Geschwindigkeit:	1,1m/s
Max. Drehgeschwindigkeit:	100°/s

Akku:

Max. Betriebszeit Stillstand:	ca. 12h
Max. Betriebszeit Fahrt:	ca. 3h
Ladedauer:	ca. 3h

Tabelle 1 Seekur Jr Spezifikationen

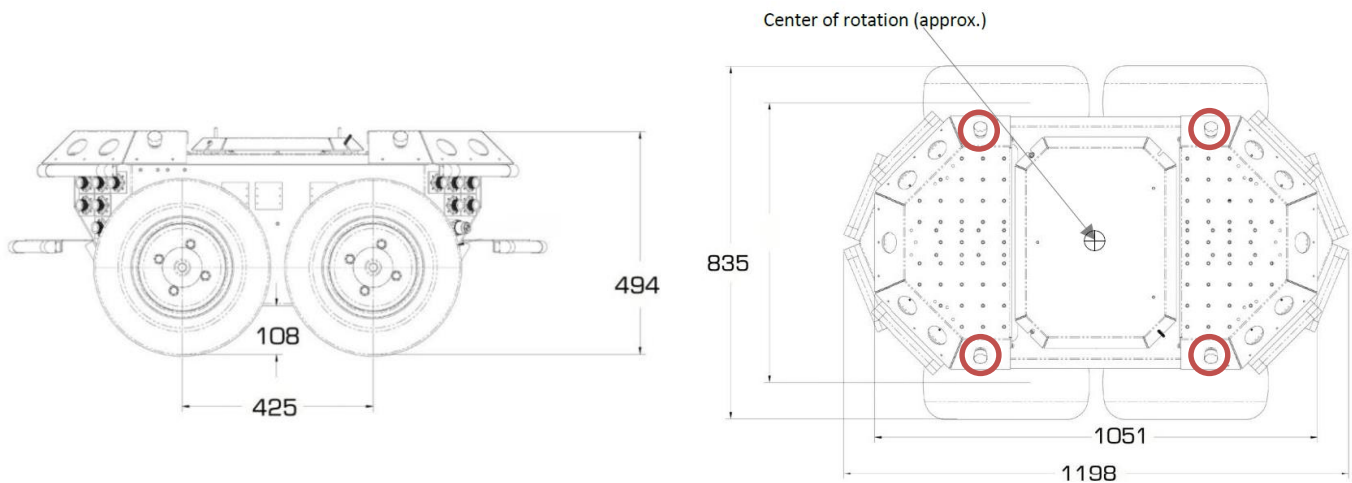


Abbildung 1 Seekur Jr Dimensionen

In Abbildung 2 sind die externen Anschlüsse des Roboters ersichtlich. Der **Stromanschluss (charger)** zum Laden der Batterien mit dem mitgelieferten Netzteil befindet sich hinten links. Gleich über dem Stromanschluss rechts ist der **Ethernet-LAN-Anschluss** zu finden, der für die spätere Netzwerkkonfiguration notwendig ist. Links davon ist der **Accessories-Anschluss** mit dessen Hilfe es mit dem speziellen Verbindungskabel möglich ist Bildschirm, Tastatur und

¹ <http://www.mobilerobots.com>

Maus an den Boardcomputer anzuschließen. Hinten rechts sind unter den Hauptschaltern der Anschluss für den **Joystick** eingebaut.

An der Vorderseite eingezeichnet ist der **µController maintenance Anschluss**, über den ein Firmwareupdate für den Microcontroller durchgeführt werden kann (siehe Abschnitt 1.3).

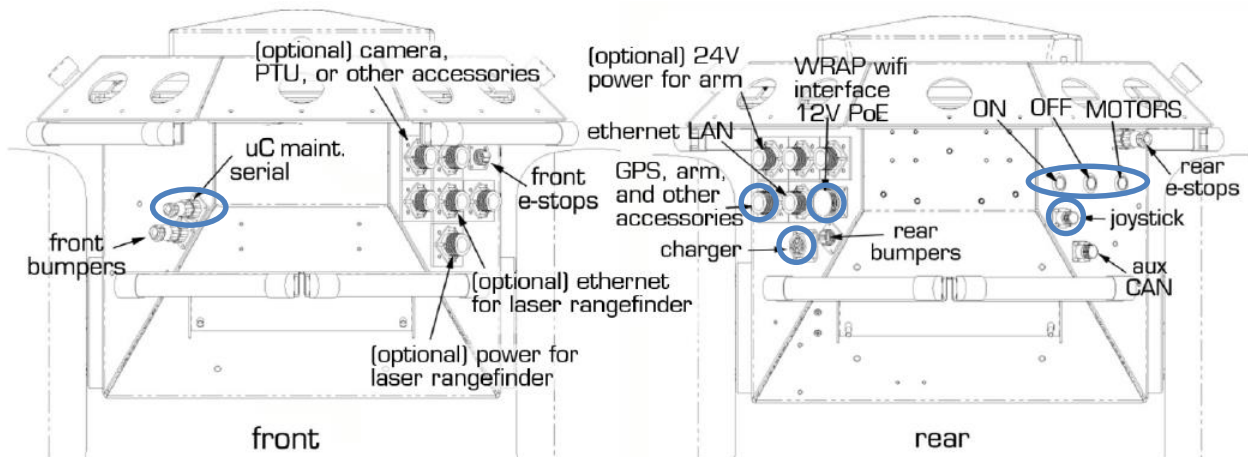


Abbildung 2 Externe Anschlüsse

1.2. Inbetriebnahme

Zur Inbetriebnahme des Roboters gibt es hinten rechts 3 Hauptschalter. ON/OFF (grün/rot) sind die Hauptschalter und sind für die Stromversorgung aller internen Bauteile mit Ausnahme der Motoren zuständig. Mit dem eigenständigen MOTORS Schalter kann man zusätzlich die Stromzufuhr der Motoren kontrollieren. Um den Roboter zu bewegen muss man zusätzlich diesen Schalter betätigen.

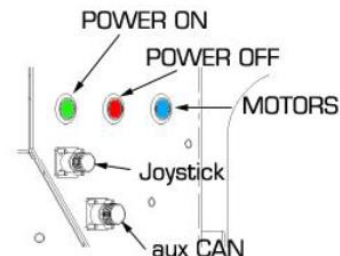


Abbildung 3 Inbetriebnahme

Ist das Netzteil angeschlossen, d.h. befindet sich der Roboter im Charging-Modus blinkt der MOTORS Schalter und eine Bewegung ist nicht möglich.

Der *Seekur Jr* ist außerdem mit 4 roten Not-Aus-Tastern ausgestattet (in Abbildung 1 ersichtlich) mit denen jederzeit die Stromzufuhr gekappt werden kann.

1.3. Interner Aufbau

Die Steuereinheit besteht aus einem Microcontroller und einem Boardcomputer. Der Microcontroller läuft unter der eigenen Firmware *SeekurOS* und ist für die Low-Level-Aufgaben zuständig, d.h. er ist für die Stromversorgung der einzelnen Komponenten zuständig, übernimmt die direkte Ansteuerung der Motoren (und koordiniert diese auch), liest die Encoder- und Gyroskopdaten aus und vieles mehr.

Der Microcontroller ist über eine RS-232 serielle Schnittstelle mit dem Boardcomputer verbunden und hat auch einen externen Anschluss (siehe Abbildung 2). Über diesen kann alternativ (ohne Boardcomputer) direkt mit dem Microcontroller kommuniziert werden.

1.4. Netzwerkkonfiguration

Zusätzlich zum Boardcomputer/Microcontroller-System ist am Roboter selbst ein Onboard-Netzwerk realisiert. Kernkomponente für dieses Netzwerk stellt ein Monowall-Router² mit Firewall dar. (siehe Abbildung 4)

Die Komponenten dieses Netzwerks sind in **Fehler! Verweisquelle konnte nicht gefunden werden.** ersichtlich:

- Boardcomputer (OS: Linux Debian 5.0)
- Sick LMS-111 Laser Rangefinder
- WRAP (Wireless Router/access point) (M0n0wall)
- Netgear N150 Wireless Router

Der Zugriff auf den Boardcomputer um die Netzwerkkonfiguration durchzuführen erfolgt über das Terminalkabel und den Accessories-Anschluss. (siehe Abschnitt 1.5)



Abbildung 4 M0n0wall Router

Konfiguration Boardcomputer:

Der Boardcomputer ist die zentrale Steuerungseinheit, die wie oben erwähnt mit dem Microcontroller kommuniziert und die Befehle an diesen weiterleitet.

Am Boardcomputer wird eine statische IP-Konfiguration durchgeführt.

Adresse:	192.168.0.32
Subnet:	255.255.255.0
Gateway:	192.168.0.150
DNS Server:	192.168.0.150

Die Einstellungen findet man über die graphische Oberfläche unter:

System > Administration > Network

² <http://www.m0n0.ch>

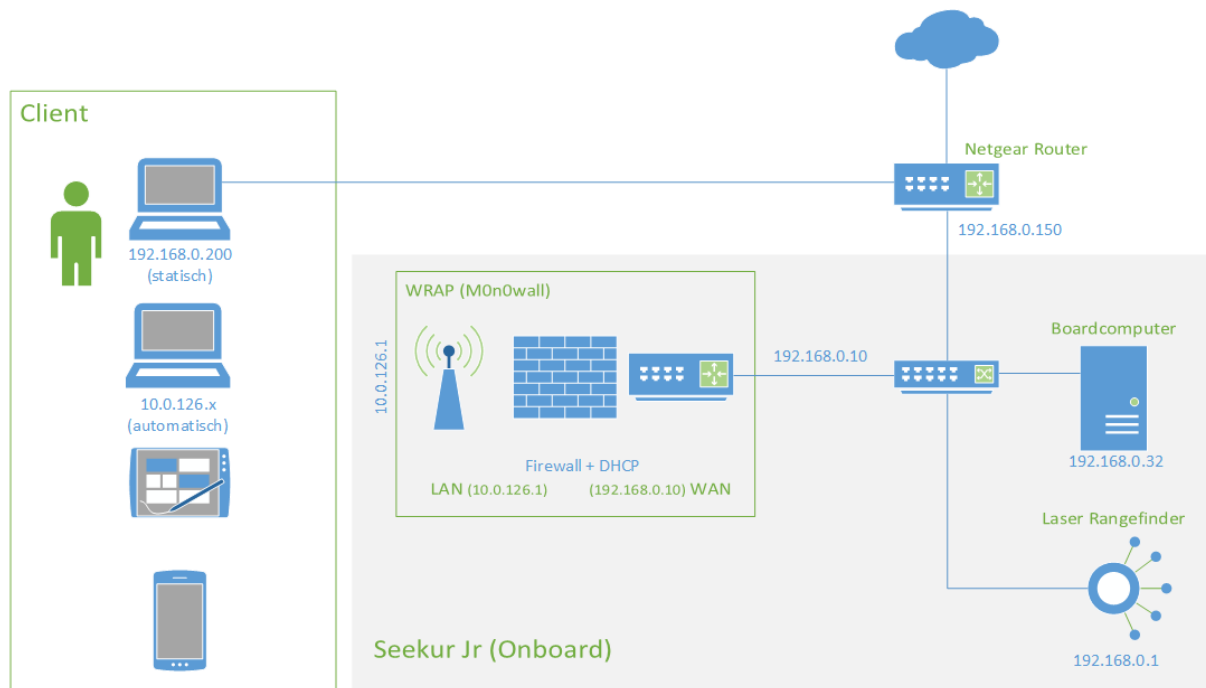


Abbildung 5 Netzwerkkonfiguration

Konfiguration WRAP:

Der WRAP ermöglicht es jedem Clientgerät außerhalb des Roboters ebenfalls mit dem Boardcomputer und den anderen Instrumenten im Onboardnetzwerk zu kommunizieren. Dazu wird ein WLAN Accesspoint eingerichtet.

Die Konfiguration des M0n0wall Routers erfolgt via Webinterface, das über <http://192.168.0.10> vom Boardcomputer aus erreichbar ist. Alternativ kann das Interface auch am Client, der sich im WLAN **SeekurJr** (falls bereits konfiguriert) befindet, über <http://10.0.126.1> erreicht werden.

Die Zugangsdaten des WLANs und Benutzername/Passwort für das Administrationsinterface findet man in Anhang A.

Konfiguration des LAN-Interfaces:

Adresse:	10.0.126.1
Subnet:	255.255.255.0 (/24)
Wireless Mode:	hostap
Wireless SSID:	SeekurJr
WPA Mode:	PSK
WPA Version:	WPA + WPA2
Cipher:	TKIP + AES/CCMP
PSK:	RoboterWlan

Konfiguration des WAN-Interfaces:

Type:	static
Adresse:	192.168.0.10
Subnet:	255.255.255.0 (/24)
Keine PPPoE oder Ähnliche Einstellungen	

Konfiguration der Firewall:

Standardmäßig sowohl auf LAN als auch auf WAN Seite alles erlauben, falls die Rule allow any nicht vorhanden ist muss diese erstellt werden.

Konfiguration DHCP-Server:

DHCP Server auf LAN-Seite aktivieren	
Range:	10.0.126.101 bis 10.0.126.200

Eine funktionierende Konfiguration des Routers befindet sich am Netzlaufwerk **unter** und kann über den Menüpunkt **Diagnostics > Backup/Restore** eingespielt werden!

Damit ist die Basiskonfiguration abgeschlossen und ein beliebiges Endgerät (IP-Adresse automatisch beziehen) ist nun in der Lage über das WLAN **SeekurJr** auf den Boardcomputer zuzugreifen.

Kontrolle über **Windows + R > cmd > ping 192.168.0.32**

Konfiguration Netgear Router:

Um es dem Boardcomputer zu Wartungs- und Updatezwecken Internetzugriff zu ermöglichen ist ein weiterer Router notwendig.

Der externe Ethernetanschluss am Roboter wird über ein Netzkabel an einen beliebigen LAN-Port am Netgear Router angeschlossen. Zusätzlich wird der gelbe WAN-Port am Netgear Router mit einem Netzwerkanschluss der TU Wien verbunden.

Das Webinterface ist über <http://192.168.0.150> erreichbar. Sollte ein Hardware-Reset notwendig sein, findet sich die Adresse des Admin-interfaces sowie Standard- Username und Passwort auf der Unterseite des Routers.

Konfiguration des WAN-Interfaces (Internet):

Adresse:	automatisch beziehen
----------	----------------------

Oder alternativ:

Adresse:	128.130.78.58
Subnet:	255.255.255.192
Gateway:	128.130.78.1
DNS:	128.130.78.3

Konfiguration des LAN-Interfaces:

Adresse:	192.168.0.150
Subnet:	255.255.255.0 (/24)
DHCP Server:	disabled
WLAN:	disabled

Damit sollte am Boardcomputer z.B. <http://www.mobilerobots.com> über den Webbrowser erreichbar sein. Die Internetverbindung ist vor allem für Abschnitt 1.5 bzw. Abschnitt 2.3 notwendig.

Eine funktionierende Konfiguration des Routers befindet sich ebenfalls am Netzlaufwerk **unter** und kann über den Menüpunkt **Erweitert > Administration > Einstellungen sichern** eingespielt werden!

1.5. Zugriff auf den Boardcomputer

Zugriff über Telnet:

Über Windows zum Beispiel mit der Freeware PuTTY. Die IP-Adresse des Onboard-Rechners (**192.168.0.32**), Port **22** eingeben und Verbindungstyp SSH auswählen.

Als Beispiel kann man dann mit `cd /usr/local/Aria/examples` und `./demo` die Demosteuerung über die Pfeiltasten starten.

Zugriff über VNC:³

Zugriff über Hardwareanschluss:

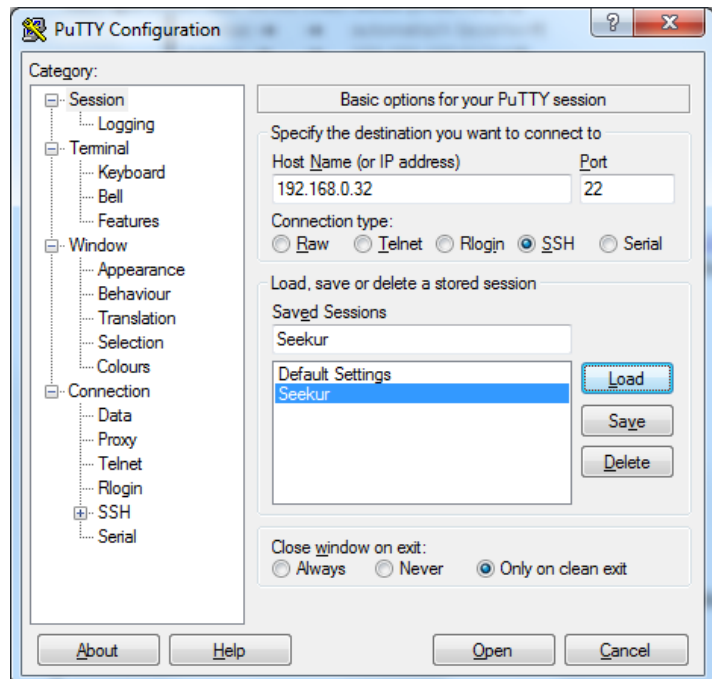


Abbildung 6 Zugriff über telnet (PuTTY)

2. Simulationsumgebung

Beschreibung VirtualBox und Konfiguration der Simulationsumgebung (Netzwerkeinstellungen)
MobileSim

3. Steuerung

Die Kommunikation mit dem Microcontroller des Roboters zur Steuerung kann über mehrere Konfigurationen erfolgen (siehe Abbildung 7 und Abbildung 8):

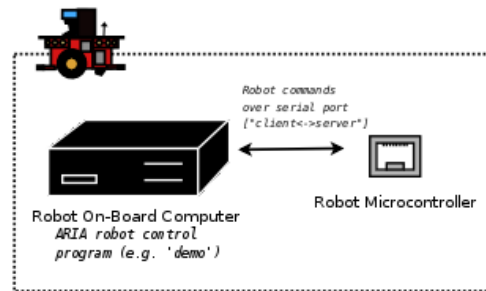
- I. Der Boardcomputer selbst übernimmt die Steuerung, d.h. dort läuft eine Anwendung die auf der ARIA API basiert. Der Zugriff auf den Boardcomputer erfolgt per Telnet oder VNC (siehe Abschnitt 1.5)
- II. Am Boardcomputer läuft eine Server-Anwendung (z.B. ARNetworking, aber nicht zwingend!). Die Steuerung erfolgt über einen beliebigen Client-Rechner im roboter-eigenen WLAN **SeekurJr**. Die Anwendung am Client-Rechner mit dem Boardcomputer erfolgt über TCP/IP und muss die von der Server-Anwendung spezifischen Methoden ausführen.
- III. Sehr ähnlich zu II. wird am Boardrechner das Programm **ipthru** gestartet. Dieses Programm fungiert als Brücke zwischen WLAN und Serial-Port. Der Client-Rechner kann dann mittels der ARIA API über diese Brücke mit dem Microcontroller

³ <http://www.realvnc.com/download/viewer/>

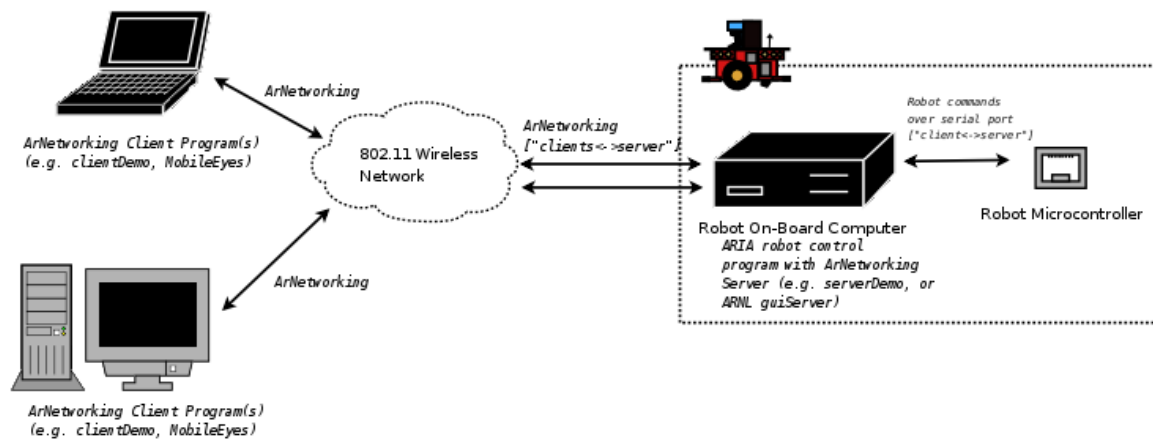
kommunizieren. Die Methoden aus den Abschnitten 3.1, 3.2 und 3.3 können somit direkt am Client verwendet werden.

- IV. Diese Option ist nutzlos, da sie nicht empfohlen wird, bzw. zusätzliche Hardware notwendig wäre.
- V. Die Kommunikation erfolgt mit einem simulierten Roboter der Anwendung MobileSim. Diese Option wurde in Abschnitt 2 verwendet.
- VI. Es ist auch möglich einen eigenen Steuerrechner direkt mit dem Microcontroller über den seriellen Anschluss zu verbinden. So könnte man seinen eigenen Boardcomputer installieren und konfigurieren.

I. Robot control program on on-board computer with no remote clients; use 'ssh' or 'telnet' to log in remotely for configuration/administration:



II. End-user clients communicate with robot control server on on-board computer (the recommended method for robots with on-board computers):



III. Robot control software on off-board computer sends commands to robot microcontroller by way of the robot on-board computer running 'ipthru' to bridge from TCP to microcontroller serial connection:

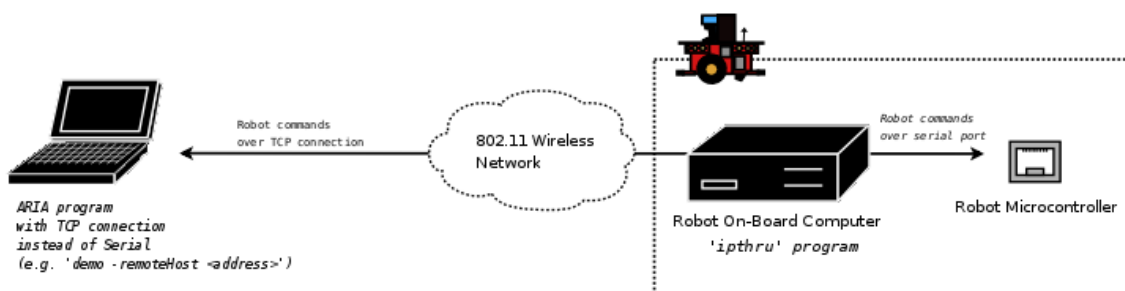
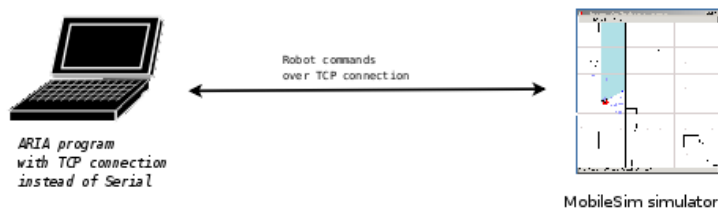


Abbildung 7 Kommunikation mit dem Roboter (Möglichkeiten 1-3)

IV. Robot control software on off-board computer sends commands to robot microcontroller by way of an serial-ethernet (TCP) bridge device (recommended method for small robots without on-board computers, such as Amigobot):



V. Robot control software on off-board computer sends commands to simulated robot via TCP:



VI. Mount your own computer and send commands to microcontroller:

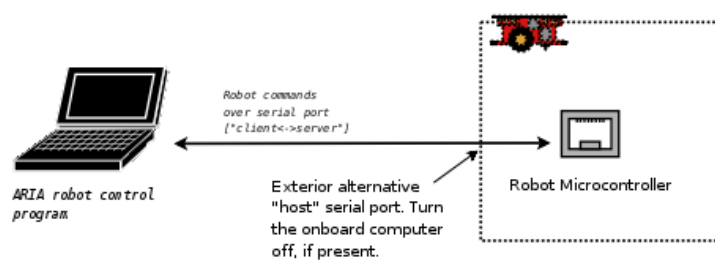


Abbildung 8 Kommunikation mit dem Roboter (Möglichkeiten 4-6)

3.1. Bewegungsmodell

Der Seekur Jr verfügt über ein Acceleration/Deceleration-Bewegungsmodell. Das heißt, wenn der Roboter einen Befehl erhält, beschleunigt er entsprechend des aktuellen Acceleration-Werts bzw. bremst

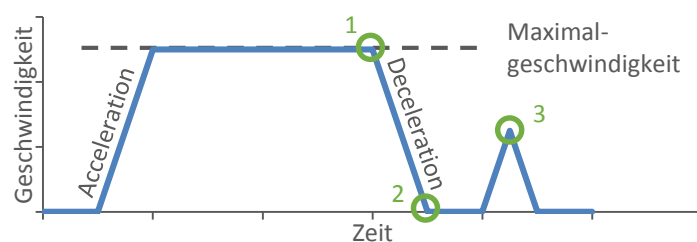


Abbildung 9 Bewegungsmodell

entsprechend des aktuellen Deceleration-Werts (siehe Abbildung 9). Man hat die Möglichkeit diese Werte sowohl für die Translation als auch für die Rotation zu setzen.

Der Microcontroller verfolgt/realisiert außerdem 2 Ansätze zur Bewegungssteuerung:

- **Start/Stop Control:** Man übergibt eine gewünschte Geschwindigkeit. Der Roboter beschleunigt entsprechend Abbildung 9 bis er die Sollgeschwindigkeit erreicht hat und hält diese bei, solange bis er einen neuen Geschwindigkeitsbefehl oder einen Stoppbefehl erhält (Markierung 1 in Abbildung 9). Dann bremst er entsprechend des Deceleration-Werts bis er schlussendlich die Ruheposition erreicht (Markierung 2 in Abbildung 9)
- **Position Integration Control:** Mit den integrierten Encoderdaten und Gyroskopdaten (und den IMU-Daten, wenn vorhanden) kann der Microcontroller selbständig seine Position berechnen. Dadurch ist es möglich eine zu fahrende Distanz, bzw. eine Rotation zu übergeben. Der Roboter berechnet dann das Geschwindigkeits-Trapez aus Abbildung 9, beschleunigt also bis er die aktuelle Maximalgeschwindigkeit erreicht hat und verzögert rechtzeitig vor der Sollposition (Markierung 2). Ist die Bewegung nur kurz kann es so auch vorkommen, dass die Maximalgeschwindigkeit nicht erreicht wird (Markierung 3 in Abbildung 9).

Übersicht der entsprechenden Commands: + Grafik Orientierung

3.2. Sensoren

Encoder, Gyroscope, Regelung (PID), Bumpers, Laserrangefinder, Sonar?, IMU?

3.3. API

MobileRobots entwickelt für die Roboter eine Palette an Softwarepaketen. Das **Advanced Robot Interface for Applications (ARIA)** stellt dabei die Kernkomponente dar und wurde in C++ entwickelt. Dieses SDK stellt die wichtigsten Basismethoden zur Steuerung des Roboters und zum Auslesen der Sensordaten zur Verfügung.

Zusätzlich gibt es darauf aufbauend C++ Libraries zur Ortsbestimmung und intelligenten Navigation. Die Libraries sind entsprechend der verwendeten Sensoren:

- ARNL – verwendet den Onboard-Laser-Rangefinder
- SONARNL – Sonar-Sensoren
- MOGS – Positionierung mittels GPS
- BASEARNL – implementiert die Navigationsfunktionalität

Eine Steueranwendung auf Basis ARIA sendet *Command Packets* an den Microcontroller. Dies erfolgt über eine der drei Möglichkeiten:

- **Direct Commands**

Die unterste Ebene der API. Damit ist es möglich ein *Command Packet* selbst an den Microcontroller zu senden. Ein Command beginnt immer mit einer Nummer, die den Befehl

spezifiziert (z.B. 4 steht für Motoren einschalten). Danach folgen, falls erforderlich, die Parameter des Befehls.

- **Motion Command Functions**

Eine Ebene drüber sind die *Motion Command Functions* angesiedelt. Diese Methoden sind Wrapper für die einfachen Bewegungsbefehle, sodass man sich als Entwickler nicht darum kümmern muss, wie das jeweilige *command packet* aufgebaut ist.

- **Actions**

Für anspruchsvollerer Bewegungs-Befehle wurde das Konzept der Actions eingeführt. Neben einer Sammlung von vordefinierten Actions hat man die Möglichkeit selbst wiederverwendbare Bewegungsmodule zu entwickeln, in denen verschiedene Sensoren (siehe Abschnitt 3.2) mit verschiedenen Bewegungsbefehle kombiniert werden können.

3.4. Entwicklungsumgebung C++

Installation neuester ARIA libs

Java SDK Installieren über Synaptic

Repositories: alle entfernen und binary (deb) <ftp.de.debian.org/debian-archive/debian> main hinzufügen

Openjdk-6-jdk installieren

Eclipse Setup

Alternative:

Java JRE 7 herunterladen

Entpacken: `tar xfv *.tar.gz`

Nach `/usr/java` kopieren: `cp -avr jdk1.7 /usr/java`

Netbeans installieren: `Sh netbeansinstaller.sh`

Linker & Compiler Einstellungen für HelloWorld Seekur Jr
(Eclipseprojekt unter `/local/th/src`)

3.5. Entwicklungsumgebung Matlab

Es besteht auch die Möglichkeit den Roboter über Matlab zu steuern. Allerdings handelt es sich dabei um ein reduziertes, vereinfachtes Interface das nicht die gesamte Funktionalität unterstützt.

Anhang A Passwörter & Zugangsdaten

Echtsystem:

Boardcomputer	USR: root	PWD: password
Monowall Admin Interface	USR: admin	PWD: mono
Netgear Admin Interface	USR:	PWD:
WLAN	SSID: SeekurJr	PWD:
Webportal mobilerobots.com	USR: Vienna4	PWD: ab7-2+B!

Simulationsumgebung:

Boardcomputer	USR: debian	PWD: reverse
---------------	-------------	--------------