

**Steuerungs- und Überwachungstechnik für  
modulare, kabelgeführte Handhabungssysteme**

Vom Fachbereich Maschinenbau

der Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte Dissertation

von

Dipl.-Ing. Matthias Blaschke

geboren am 24.5.1967 in Minden

2001

- 1 .Referent: Prof. Dr.-Ing. Dr.-Ing. E.h. mult. H. Haferkamp
2. Referent: Prof. Dr.-Ing. Dr.-Ing. E.h. mult. H. K. Tönshoff
3. Referent: Prof. Dr.-Ing. M. Gietzelt

Tag der Prüfung: 18.10.2001

---

## **Vorwort**

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkstoffkunde der Universität Hannover.

Meinem Doktorvater, Herrn Prof. Dr.-Ing. Dr.-Ing. E.h. mult. H. Haferkamp, dem ehemaligen Direktor des Institutes für Werkstoffkunde, gilt mein besonderer Dank für die Unterstützung und Förderung der Arbeit.

Ebenso danke ich Herrn Prof. Dr.-Ing. Dr.-Ing. E.h. mult. H. K. Tönshoff, dem Leiter des Instituts für Fertigungstechnik und spanende Werkzeugmaschinen, für die Übernahme des ersten Koreferats.

Herrn Prof. Dr.-Ing. M. Gietzelt, dem Leiter des Instituts für Energietechnik, sei an dieser Stelle für die Übernahme des Vorsitzes innerhalb des Promotionsverfahrens gedankt.

Herrn Prof. Dr.-Ing. Fr.-W. Bach, Direktor des Institutes für Werkstoffkunde, danke ich für die Anregung der Arbeit sowie für die fachliche Unterstützung.

Für die langjährige Zusammenarbeit auf dem Gebiet der Handhabungs- und Steuerungstechnik danke ich meinen ehemaligen Kollegen Marc Bachmann, Jörn Seevers, Hans-Jörg Mahner, Stefan Knepper, Reiner Schwieger und Marc-Oliver Borck. Ebenso danke Mathias Niemeyer, Thomas Kühn, Reiner Schwieger und Stefanie Jöhren für das Korrekturlesen der Arbeit.

Mein besonderer Dank gilt auch allen Studenten, die mich in Rahmen Ihrer Tätigkeiten als hilfswissenschaftliche Mitarbeiter sowie als Studien- und Diplomarbeiter in meiner Arbeit unterstützt haben.

## **Abstract**

Die Durchführung von Wartungs-, Inspektions-, Reparatur- und Stilllegungsarbeiten in kerntechnischen Anlagen setzt in weiten Bereichen den Einsatz von fernhantierbarer Technik voraus, um die Belastung des Personals durch radioaktive Strahlung zu minimieren. Neben Führungsmaschinen, Industrierobotern oder auch fernhantierbaren Werkzeugen können in den vorhandenen Unterwasserinstallationen wie beispielsweise dem Brennelemente-Lagerbecken auch Unterwasserfahrzeuge, sog. ROVs (Remotely Operated Vehicles), als mobile Systemträger eingesetzt werden.

Am Markt verfügbare, kleinbauende ROVs sind von der konstruktiven und steuerungstechnischen Auslegung in der Regel als geschlossene Systeme anzusehen und nur durch den Hersteller bzw. mit herstellerspezifischen Komponenten erweiterbar. Meist fehlen Kommunikations- bzw. Signal-Schnittstellen, um individuelle Erweiterungen mit zusätzlichen Modulen oder Sensoren vornehmen zu können.

Im Rahmen dieser Arbeit wurde ein Unterwasser-Handhabungssystem mit der Bezeichnung TRIBUN konzipiert und realisiert, das eine Kombination aus universellem Trägerfahrzeug und Funktionsmodulen darstellt. Diese Fahrzeug- und Werkzeugmodule werden über vier Multifunktionskupplungen am Rumpf adaptiert. Als entscheidende Neuerung gegenüber verfügbaren Systemen wird dem Endanwender somit eine Art Baukastensystem für die Unterwasser-Handhabungstechnik zur Verfügung gestellt, das eine Konfiguration des Systems auf Basis der aktuellen Gegebenheiten am Einsatzort erlaubt. Durch Verwendung eines offenen Multimaster-Steuerungskonzept und die Vernetzung der Module über den CAN-Bus, einem standardisierten, in der Automatisierungstechnik weit verbreitetem Feldbus, kann das System problemlos über die ursprüngliche Spezifikation hinaus erweitert werden.

Erweiterungen des PC-basierten Steuerungsprogramms und zusätzliche Feldbus- und Schnittstellenkoppler erlauben eine Fernbedienung des Fahrzeugs über ein Rechnernetzwerk sowie bei komplexen Handhabungsaufgaben eine Aufteilung von Steuerungs- und Bedienfunktionen auf mehrere Rechner und Bediener. Als unterstützendes System wurde eine 3D-Visualisierungs-Software entwickelt, die auf den VRML97-Standard basiert und dem modularen Systemcharakter gerecht wird. Die Kontrolle und Überwachung des Fahrzeugs kann im Netzwerkbetrieb über einen Webbrowser erfolgen.

Durch den Einsatz von Mensch-Maschine-Schnittstellen wie Head-Mounted-Display, Headtracker oder 6-Achsen-Joystick wird ein hoher Bedienkomfort gewährleistet und gleichzeitig die Einarbeitungs- und Trainingszeiten gering gehalten.

Die vorgestellten Hard- und Softwarelösungen sind dabei nur zum Teil für ein ROV ausgelegt und weisen direkt oder mit leichten Modifikationen eine hohe Einsatzbandbreite im Bereich der Handhabungstechnik auf. So lassen sich Entwicklungs- und Testzeiten und damit auch die Kosten für Entwurf und Erprobung von Prototypen reduzieren. Gleichzeitig können mit geringem finanziellem Aufwand Handhabungssysteme mit verbesserten Kontroll- und Überwachungsfunktionen ausgestattet werden.

## **Schlagworte:**

Steuerungstechnik, Handhabungstechnik, Fernhantierung, Unterwasserfahrzeug, ROV, 3D-Visualisierung, Netzwerktechnik, Überwachungstechnik

---

**abstract**

Carrying-out inspection, maintenance, repair and decommissioning work in nuclear facilities demands for tele-operated technology within a wide range to minimize the radioactive load on the personnel. In addition to guiding machines, industrial robots or teleoperated tools underwater vehicles, the so called ROVs (Remotely Operated Vehicles) can be used in underwater facilities like the fuel rod storage basin as mobile system carriers.

In general small-building ROVs being available on the market today must be treated as closed systems in terms of mechanical and control design. They are only extendable by the manufacturer or customized components. Often communication or signal interfaces are missing preventing individual extensions with further modules or sensors.

In the frame of this work an underwater handling system called TRIBUN has been developed and realized. It represents a combination of a universal carrier vehicle and functional modules. Four multi-functional ledges with plugs located on the pressure housing allow the mounting of these functional modules. The essential innovation in comparison to systems on the market is providing the user with a sophisticated construction kit allowing him to choose the final configuration on the basis of the current conditions at the location of use. By employing an open multi-master control design and connecting the modules via CAN-bus, a standardized field bus systems widely used in automation technology, the system can be extended easily beyond the original specification.

Extensions of the PC-based control programme and additional field bus and interface couplers permit a remote operation of the vehicle via a computer network. Furthermore control and operational functions can be distributed among various computers and operators. A three dimensional visualization software has been developed as a supporting system. It is based on the VRML97 standard and meets the requirements of the modular system design. The control and monitoring of the vehicle is possible in networking mode by using a web browser.

Employing man-machine-interfaces like a head-mounted-display, a head-tracker or six-axis-joystick ensures a high comfort of operation. At the same time the training periods are significantly reduced.

The hard- and software solutions presented here are only partly designed for a ROV. Directly or with slight modifications they have a wide range of application in the area of handling technology. Thus development and test times and therefore also the costs for prototype development are reduced. At the same time handling systems can be equipped with improved control and monitoring functions at small expenditure.

**Keywords:** control engineering, handling technology, remote handling, underwater vehicle, ROV, 3D-visualization, network technology, monitoring technology

---

1	Einleitung .....	1
2	Stand der Technik und Wissenschaft.....	2
2.1	Industrieroboter und mobile Systemträger.....	2
2.2	Unbemannte Unterwasserfahrzeuge .....	3
2.2.1	Aufbau und Steuerung von ROVs.....	3
2.2.2	Kleinbauende ROVs.....	4
2.2.3	ROVs für den Einsatz in kerntechnischen Anlagen .....	7
2.2.4	Vorteile modular aufgebauter Fahrzeuge .....	9
2.2.5	Unterwasser-Robotersystem FAUST III .....	9
2.2.6	Modulares Unterwasserfahrzeug $\mu$ -FAUST .....	10
2.3	Motivation für die Neuentwicklung eines modularen ROVs.....	12
3	Konzeption und Aufbau des Systems .....	16
3.1	Systemanforderungen .....	16
3.2	Systemkonzept .....	17
3.3	Mechanisches Konzept und Aufbau des Fahrzeugs .....	18
3.4	Steuerungstechnik und Steuerungsarchitektur .....	22
3.4.1	Anforderungen an ein Bussystem für TRIBUN .....	23
3.4.2	Etablierte Feldbussysteme im Überblick .....	24
3.4.3	Modulelektronik und Modulaufbau .....	27
3.4.4	Auswahl der Mikrocontroller-Familie .....	28
3.4.5	Energieversorgung und Versorgungskabel .....	30
3.4.6	Plug&Play-Konzept mit Steckplatzkennung .....	32
3.4.7	Spezifische Motormodulkennwerte .....	33
3.4.8	CAN-Kommunikationsprotokoll .....	36
3.4.9	CAN-Kommunikationsverhalten der Module .....	37
3.5	Komponenten des Unterwasserfahrzeugs TRIBUN .....	39
3.5.1	Kamerasystem .....	39
3.5.2	Propellerantriebe.....	42
3.5.3	Manipulator mit Schnellkupplung .....	42
3.5.4	Zweiachsige Linearführungsmaschine.....	47
3.5.5	Lage- und Tiefenerfassung .....	48
3.5.6	Baukastensystem für Roboterarme.....	52
3.6	Mensch-Maschine-Schnittstellen .....	54
3.6.1	Schnittstellen zur Fahrzeug- und Werkzeugsteuerung .....	55
3.6.2	Schnittstellen des Kamera- und Videosystems .....	57
3.7	PC-Software für Steuerung und Bedienung .....	60
3.7.1	Softwarearchitektur .....	60
3.7.2	Objektorientierte Zerlegung des Gesamtsystems .....	61
3.7.3	Identifizierung der Klassen und Objekte .....	61
3.7.4	Identifizierung der Semantik der Klassen und Objekte .....	62
3.7.5	Systemdesign.....	63
3.7.6	Klassenimplementierungen der CAN-Kommunikation .....	64
3.7.7	Realisierung der Systemklassen und Einbindung der GUI-Elemente.....	65

---

3.7.8	Oberflächenelemente der PC-Bedienoberfläche .....	68
3.7.9	Handbediengerät.....	70
<b>3.8</b>	<b>Netzwerk-Funktionalität .....</b>	<b>71</b>
3.8.1	Schnittstellen und Buskoppler für den Netzwerkbetrieb.....	71
3.8.2	Videodatenübertragung für den Netzwerkbetrieb .....	72
3.8.3	Anbindung an die Simulation und das Positionssystem.....	72
<b>3.9</b>	<b>Sicherheitsaspekte .....</b>	<b>73</b>
<b>3.10</b>	<b>System für die 3D-Visualisierung unter VRML 97 .....</b>	<b>75</b>
3.10.1	Vorbetrachtungen zur 3D-Visualisierung.....	75
3.10.2	Grundlagen und Einsatzgebiete von VRML 97 .....	76
3.10.3	Programmier- und Kommunikationsschnittstellen.....	79
3.10.4	Datenverarbeitung des Steuerrechners .....	82
3.10.5	Funktionalität von TRIBUNVisu.....	85
3.10.6	Anwendungsbeispiele der 3D-Visualisierung.....	85
<b>4</b>	<b>Durchgeführte Tests und Ergebnisse.....</b>	<b>88</b>
4.1	Testumgebung .....	88
4.2	Inbetriebnahme der Module und Konfiguration des Systems .....	88
4.3	Fahrverhalten und Bedienung .....	90
4.3.1	Bedienung und Bewertung der MMIs.....	90
4.3.2	Rumpf mit Antriebs- und Greifarmvariationen.....	91
4.3.3	TRIBUN mit Führungsmaschine .....	91
4.4	Test der 3D-Visualisierung.....	92
4.5	Fernhantierung im Netzwerkbetrieb.....	94
4.5.1	Fernhantierung über LAN und Modem.....	94
4.5.2	Testszenarien für Fernhantierung und Fernüberwachung .....	95
4.6	Regelungstechnische Bedienerunterstützung .....	97
4.7	Systemüberblick .....	97
<b>5</b>	<b>Zusammenfassung .....</b>	<b>99</b>
<b>6</b>	<b>Literatur.....</b>	<b>101</b>

**Formelzeichen:**

$a$	$m/s^2$	: Beschleunigung
$a_i$	$m$	: Denavit-Hartenberg-Parameter
$a_{min}$	$m, rad$	: Achsuntergrenze
$a_{max}$	$m, rad$	: Achsobergrenze
$a_{mx}, a_{my}, a_{mz}$	$m/s^2$	: Messwerte des Lagesensors
$a_{mx0}, a_{my0}, a_{mz0}$		: Normierte Messwerte des Lagesensors
$a_x, a_y, a_z$	$m$	: Drehachse, Übergänge der Koordinatensysteme
$\vec{a}$		: Drehachse, VRML; Beschleunigungsvektor
$A$		: Denavit-Hartenberg-Matrix zur Transformation
$d_i$	$m$	: Denavit-Hartenberg-Parameter Abstand
$d_x, d_y, d_z$	$m$	: Translationen, Übergänge der Koordinatensysteme
$f$	$MB/s$	: Übertragungsrate
$f_{max}$	$\%$	: Maximalfehler des Lagesensors
$F$	$N$	: Kraft, allgemein
$g$	$m/s^2$	: Erdbeschleunigung, $9,81 m/s^2$
$I$	$A$	: elektrischer Strom
$k$		: Anzahl der vertikalen Propellerantriebe
$l$		: Anzahl der Antriebsmodule
$L$	$m$	: Längenangabe, allgemein
$L_1, L_2$	$m$	: Längenangabe Manipulator
$m$	$kg$	: Masse, allgemein
$M$	$Nm$	: Moment, allgemein
$n$		: relative Steckplatznummer; Anzahl horizontaler Antriebe
$N$		: Anzahl, allgemein
$q$	$rad$	: Manipulator Achsstellung, allgemein
$q_1, q_2, q_3$		: Quaternionen, Orientierung von VRML-Objekten
$R$	$\Omega$	: elektrischer Widerstand, allgemein
$R_{dezimalnm}$		: ADC-Wert der Widerstand-Steckplatzkennung
$R'$	$m$	: Abstand des Kraftangriffspunkt eines Motors

---

$T$		: Denavit-Hartenberg-Matrix für 3-Achsenmanipulator
$W$	m	: Steckplatzabstand
$x$		: Fahrzeug-Längsachse; Koordinate, allgemein
$x'$		: x-Koordinate im Steckplatz-KS
$y$		: Fahrzeug-Querachse; Koordinate, allgemein
$y'$		: y-Koordinate im Steckplatz-KS
$z$		: Fahrzeug-Vertikalachse; Koordinate, allgemein
$z'$		: z-Koordinate im Steckplatz-KS
$\alpha$	°	: Winkel, allg.; Fahrzeug-Rollwinkel; Nulllage, Steckplatz
$\alpha_i$	rad	: Denavit-Hartenberg-Parameter
$\beta$	°	: Fahrzeug-Neigungswinkel; Drehwinkel, Steckplatz
$\gamma$	°	: Fahrzeug-Gierwinkel; Höhenwinkel, Steckplatz
$\theta_{am}$	°	: Winkel, Messwert des Lagesensors (in Kugelkoordinaten)
$\theta_E$	rad	: Orientierungswinkel im Effektorsystem
$\delta$	m	: Translation, allgemein
$\vartheta_i$	°	: D.-H. Parameter
$\lambda$	%/N	: Prozentualer Fehlerkoeffizient des Lagesensors
$\varphi_{am}$	°	: Winkel, Messwert des Lagesensors (in Kugelkoordinaten)
$\phi$	°	: Lagewinkel, Steckplatz
$\Psi_E$	rad	: Orientierungswinkel im Effektorsystem

**Abkürzungen:**

A/D	Analog Digital
ADC	Analog Digital Converter
AMP	Arbitration on Message Priority
API	Application Programmers Interface
ASCII	American Standard Code for Information Interchange
ASI	Actuator/Sensor-Interface
AUV	Autonomous Underwater Vehicles
CAD	Computer Aided Design
CAL	CAN Application Layer
CAN	Controller Area Network
CCD	Charge Coupled Devices
CiA	CAN in Automation
CSMA/CA	Carrier Sense/Multiple Access with Collision Detection
DAC	Digital Analog Converter
DC	Direct Current
DIN	Deutsches Institut für Normung
DMP	Digital manufacturing and production
DTR	Data Terminal Ready
DVD	Digital Versatile Disc
EAI	External Authoring Interface
EIB	European Installation Bus
EPROM	Erasable Programmable Read Only Memory
FAUST	Freitauchender Autonomer Systemträger
FBAS	Farbsignal, Bildsignal, Austastsignal, Synchronisation; Bildsignal
FIP	Factory Instrumentation Protocol
GUI	Graphical User Interface
HMD	Head-Mounted-Display
HTML	Hypertext Markup Language
I/O	Input/Output
IBS	Ingenieur
IC	Integrated Curcuit
ID	Identifier
ISA	Industry Standard Architecture
ISO	International Standards Organisation
JDK	Java Developmet Kit
JVM	Java Virtual Machine
KISMET	Kinematic Simulation, Monitoring and Off-Line Programming Environ- ments for Telerobotics
KS	Koordinatensystem
LAN	Local Area Network
LED	Lichtemittierende Diode, Leuchtdiode

---

LON	Local Operating Network
MDI	Multiple Document Interface
MMI	Man Machine Interface
MS	Microsoft
OWL	Object Windows Library
PAL	Phase Alternating Line
PC	Personal Computer
PCI	Peripheral Component Interconnect
P-NET	Process-Network
Profibus DP	Process Fieldbus for Decentralised Peripherals
Profibus FMS	Process Fieldbus Message Specifications
PWM	Pulsweiten Modulation
ROV	Remotely Operated Vehicles
RTR	Remote Transmission Request
SGI	Silicon Graphics Industries
SPS	Speicher Programmierbare Steuerung
TCP/IP	Transmission Control Protocol / Internet Protocol
TRIBUN	Teilautonomes Robotersystem für Inspektion und Bearbeitung im Unterwasserbereich
TTL	Transistor-Transistor-Logik
URL	Uniform Resource Identifier
USB	Universal Serial Bus
VGA	Video Graphics Adapter
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
WWW	World Wide Web

# 1 Einleitung

Die Durchführung von Arbeiten im Unterwasserbereich an Bohrplattformen, Pipelines und vergleichbaren Installationen kann von Tauchern nur unter erheblicher Gefährdung ihrer Gesundheit und mit hohem finanziellen Aufwand erfolgen. Zu diesen Arbeiten gehören beispielsweise Berge- und Beobachtungsaufgaben, die Aufnahme von Messdaten oder die Durchführung von Reinigungs-, Wartungs- und Reparaturarbeiten.

Oft können bemannte Unterwasserfahrzeuge dort eingesetzt werden, wo ausreichend Raum für die aufwendige Technik verfügbar ist. Trotzdem ist man auf Grund des hohen Risikos und der hohen Kosten, gerade für die lebenserhaltenden Systeme, bestrebt, den Menschen in diesen Bereichen durch Maschinen zu ersetzen.

So wurde in den sechziger Jahren des letzten Jahrhunderts mit der Entwicklung von ferngelenkten Unterwasserfahrzeugen sog. ROVs (Remotely Operated Vehicles) begonnen, die zunächst schwerpunktmäßig die Funktion einer tauchenden Kamera hatten. Kontinuierlich wurde die Funktionalität durch Greifer, Manipulatoren und unterwassertaugliche Werkzeuge, z. B. zum Schweißen und Schneiden, oder auch durch Sensoren erweitert. Gerade im Bereich der Meerestechnik sind die Abmessungen eines Unterwasserfahrzeugs oft von untergeordneter Bedeutung, da Arbeitsraum und Installationen entsprechend groß sind.

Verlagert sich das Einsatzgebiet aber in technische Anlagen mit Unterwasserinstallationen, wie sie in der chemischen und der kerntechnischen Industrie zu finden sind, müssen die Geräte zunehmend kleinbauender werden, um ihre Aufgabe erfüllen zu können. Besondere Bedingungen liegen dabei bei einem Einsatz in einer kerntechnischen Anlage vor, da neben der normalen Gefährdung durch das Wasser zusätzlich radioaktive Strahlung auftritt und eine Kontamination mit radioaktiven Partikeln erfolgen kann.

Das Unterwasser-Handhabungssystem TRIBUN (Teilautonomes Robotersystem für Inspektion und **B**earbeitung im **U**nterwasserbereich) ist eines von mehreren Geräten einer neuen Generation im Bereich der Handhabungstechnik, die am Institut für Werkstoffkunde der Universität Hannover entwickelt und aufgebaut werden. Angesiedelt ist das Projekt zur Entwicklung von TRIBUN im Sonderforschungsbereich 264 "Automatisierte Fertigung unter Wasser", dessen Aufgabe es ist, Technologien für die Wartung, Reparatur und Stilllegung für den Offshorebereich und für kerntechnische Anlagen zu entwickeln und zu qualifizieren.

Ziel der vorliegenden Arbeit ist die Entwicklung eines hochmodularen, flexibel einsetzbaren, freitauchenden Unterwasser-Systemträgers, wobei der Schwerpunkt der Arbeit auf dem Gebiet der Steuerungstechnik liegt. Der Systemträger soll im Gegensatz zu herkömmlichen am Markt befindlichen Geräten vom Anwender nahezu frei für eine Vielzahl von Aufgaben konfiguriert werden können. Das System soll sich dabei durch extrem kurze Rüstzeiten, hohe Betriebssicherheit sowie Wartungs- und Bedienungs-freundlichkeit auszeichnen.

Innerhalb der Arbeit werden steuerungstechnische Lösungen aufgezeigt, die über die Anwendung auf das eigentliche Unterwasserfahrzeug hinaus gehen und für ein möglichst breites Anwendungsfeld auf dem Gebiet der Handhabungstechnik portierbar sind. Damit soll generell eine Verkürzung von Entwicklungszeiten erreicht werden und eine Kostenreduzierung bei Entwurf und Realisierung bei gleichzeitiger Erweiterung der Funktionalität und verbesserter Handhabung erfolgen.

## 2 Stand der Technik und Wissenschaft

### 2.1 Industrieroboter und mobile Systemträger

Der Mensch verfügt mit seinen Extremitäten über sehr universell einsetzbare von der Natur mitgegebene "Werkzeuge". Dennoch stößt er schnell an seine Grenzen, da er wenig spezialisiert ist. Er ist daher darauf angewiesen, seine Arbeits- und Lebensumgebung dieser Besonderheit anzupassen. Beispielsweise fehlt ein schützendes Fell als Wärmedämmung und außerdem ist die Haut extrem verletzlich, so dass er mit seinen Händen allein kaum die Möglichkeit hat, Objekte zu bearbeiten.

Im Laufe seiner Entwicklungsgeschichte hat der Mensch daher, angefangen bei primitiven Werkzeugen aus Holz und Steinen und später aus Metallen, immer raffiniertere und komplexere Werkzeuge zu entwickeln. Während die Werkzeuge anfangs nahezu ausschließlich manuell geführt wurden, wurde während der Industrialisierung begonnen, die Werkzeuge in großem Umfang auch teilautomatisiert und automatisiert zu führen. Beschleunigt wurde dieser Prozess durch die Entwicklung von elektronischen Steuerungen. Heute ist der Industrieroboter aus der industriellen Fertigungstechnik nicht mehr wegzudenken. Er führt schwere und monotone Arbeiten mit hoher Präzision aus und arbeitet, im Gegensatz zum normalen Arbeiter, ermüdungsfrei. Die vollautomatische Produktionsanlage, in der der Mensch im wesentlichen nur noch Überwachungsfunktionen ausführt, ist damit bereits Realität.

Handelsübliche Roboter stoßen auf Grund des häufig geringen Aktionsradius oder auch der Baugröße schnell an ihre Grenzen. Abhilfe schaffen hier Geräte, die sich im Arbeitsraum frei bewegen können, sog. mobile Systemträger. Sie fungieren als Träger für die eigentlichen Werkzeuge und Sensoren und übernehmen die notwendigen Transport- und Führungsfunktionen. Ein großes Anwendungsfeld ist dabei heute schon der gesamte Servicebereich.

Mittlerweile wurde eine Vielzahl von Servicerobotern entwickelt, die in den unterschiedlichsten Bereichen des Alltags zum Einsatz kommen. Neben der industriellen Fertigungstechnik gelangten sie in nahezu alle Bereiche des Lebens wie Landwirtschaft, Baugewerbe, Hotel und Gastronomie, Hobby und Freizeit, aber auch in Medizin und Weltraumtechnik. Die Erscheinungsformen reichen dabei von Robotern in herkömmlichen Armkinematiken über mobile Fahrzeuge mit Raupen-, Ketten- oder Radantrieb. Aber auch die verschiedensten Ausführungen von Wandkletter- und Schreitrobotern sowie Tauchfahrzeugen sind dort zu finden /1/.

Während die reinen Serviceroboter oft der Kostenreduzierung dienen, müssen z. B. in kerntechnischen Anlagen auf Grund der vorliegenden radioaktiven Strahlung die Arbeiten generell in weiten Bereichen mittels fernhantierbarer Werkzeuge, Führungsmaschinen, Roboter oder auch mobiler Systemträger erfolgen. Es wurden dabei gerade für die Stilllegung solcher Anlagen eine sehr große Anzahl dieser Geräte entwickelt und qualifiziert /2/-/12/.

Wo Maschinen den Menschen bei der Arbeit ersetzen sollen, müssen sie auch mit Schutzmechanismen ausgerüstet werden. So sind beispielsweise Taucher gegen das Auskühlen mit einer Wärmeisolierung (Neoprenanzug) ausgerüstet und verfügen über technische Geräte, die ein Atmen unter Wasser (Druckluftflasche, Atemregler etc.) erlauben. Analog dazu muss ein Gerät gegen das Eindringen von Wasser geschützt

werden. Gleichzeitig ist es erforderlich, die im Gerät entstehende Wärme an die Umgebung abgeben zu können. Generell besteht das Problem, dass bekannte Lösungen nicht einfach übernommen werden können, sondern für den Unterwasserbereich angepasst oder neu entwickelt werden müssen /13/.

Durch die räumliche Trennung von Mensch und Werkzeug verliert er gleichzeitig den direkten Einblick und damit auch die optische Kontrolle bei der Durchführung der Arbeiten. Können die Arbeitsabläufe nicht vollautomatisch ausgeführt werden, so sind unterstützende Techniken wie eine Beobachtung per Videokamera oder auch Methoden aus dem Bereich der virtuellen Realität zur Kontrolle gefordert.

## 2.2 Unbemannte Unterwasserfahrzeuge

Unbemannte Unterwasserfahrzeuge lassen sich in kabelgebundene und autonome Fahrzeuge unterteilen. Autonome Unterwasserfahrzeuge (engl. Autonomous Underwater Vehicles, AUV) verfügen über alle notwendigen Subsysteme wie eine interne Energieversorgung und Systeme zur Navigation und können daher ohne Kabelanbindung operieren /14/. Bei Abmessungen zwischen 1,5 und 9 Metern Länge besitzen sie teilweise einen Aktionsradius von einigen hundert Kilometern. AUVs sind stromlinienförmig und manövrieren im Allgemeinen mit Hilfe von Seiten- und Höhenrudern. Kabelgebundene Systeme lassen sich in geschleppte und angetriebene Geräte unterteilen. Die geschleppten Geräte (Towed Vehicles) sind in der Regel ebenfalls stromlinienförmig und benutzen Flossen oder Ruder, um eine bestimmte Tauchtiefe zu halten. Dem gegenüber verfügen ROVs über integrierte Antriebe, über die sie von Leitstand aus manövriert werden /18/.

ROVs werden seit den 70er Jahren in wachsendem Maße in der Offshoreindustrie bei der Inspektion und Wartung von Plattformen und Pipelines eingesetzt. Im Laufe ihrer technischen Entwicklung wurden aus den "tauchenden Kameras" hochkomplexe Unterwasser-Handhabungssysteme, ausgestattet mit Manipulatoren und diversen Sensoren. Mit den Aufgaben und der Einsatztiefe wuchsen die Abmessungen und das Gewicht der ROVs. Die Einsatzmöglichkeiten wurden umfangreicher und so entstand als Konsequenz der Wunsch, die Fahrzeuge an verschiedene Aufgaben anpassen zu können. Dabei ist das universell einsetzbare, modulare ROV bislang nicht über das Konzeptstadium hinausgekommen /15/.

### 2.2.1 Aufbau und Steuerung von ROVs

Am Markt sind momentan über hundert verschiedene ROVs verfügbar, die für die unterschiedlichsten Aufgabenbereiche entwickelt wurden /18/,/28/. Das System ROV besteht dabei aus den folgenden Hauptkomponenten:

- dem eigentlichen Systemträger mit Anbauteilen wie z. B. Antrieben, Scheinwerfern, Manipulatoren, Sensoren, Trimmgewichten und Auftriebskörpern,
- einem Versorgungskabel, über das das Fahrzeug mit der benötigten elektrischen Energie und Steuersignalen versorgt wird
- sowie einem Leitstand bestehend aus Energieversorgung, Videomonitor und einer Steuerungseinheit.

Der Aufbau von ROVs besteht in der Regel aus einer offenen Rahmenkonstruktion und einer Zahl von druckdichten Gehäusen, die zur Aufnahme der Komponenten dient. Die Gehäuse werden an dem Rahmen befestigt und untereinander meist über Kabelverbindungen an die Steuerungseinheit im ROV angebunden. Mit Werkzeugen wie Master-Slave-Manipulatoren, Kettenantrieben, Greifern sowie Sensorik zur Navigation und Aufnahme von Messdaten lässt sich die Funktionalität leicht erweitern. Diese ROVs haben in der Regel große Abmessungen und werden zumeist im Offshorebereich eingesetzt. Zusätzlich stehen zum Austrimmen der Fahrzeuge Auftriebskörper und Ballastgewichte zur Verfügung.

Die norwegische Firma Hitec bietet z. B. mit ihrem ROV HiROV3000 ein universelles Fahrzeug mit komplexem Zubehör für den Offshorebereich bis 3000 m Einsatztiefe an. Neben Kompass, Sonar und mehreren Kameras sind als Zusatzausstattung auch Manipulatoren mit bis zu sieben Achsen erhältlich (Abb. 1). Zur Steuerung des Fahrzeugs steht ein kompletter Leitstand mit einer eigens entwickelten Mensch-Maschine-Schnittstelle (Man-Machine-Interface kurz MMI) vom Typ Cyberbase zur Verfügung. Zusätzlich verfügt das System über eine 3D-Visualisierung (VisiROV) für eine Überwachung während des Betriebes. Der Leitstand und die Visualisierung können auch an eine Simulationssoftware angebunden und für ein Bedienertraining genutzt werden /16/,/17/.



Abb. 1: ROV HiROV3000 (l.) mit Leitstand Cyberbase (m.) und 3D-Visualisierung VisiROV(r.)

Ein solches System deckt alle wichtigen Bereiche zur Führung, Visualisierung, Training und Kontrolle ab.

Bei zunehmend kleineren Abmessungen der Fahrzeuge reduziert sich auch die Einsatzbandbreite. Kleinbauende ROVs werden im Wesentlichen für die optische Inspektion, die Aufnahme von Messdaten sowie die Bergung und Manipulation von Kleinteilen eingesetzt.

### 2.2.2 Kleinbauende ROVs

Nachfolgend soll beispielhaft auf einige kleinbauende ROVs eingegangen werden, wobei deren Abmessungen in Länge, Breite und Höhe unter einem Meter liegen. In Tabelle 1 werden die technischen Eigenschaften für eine Auswahl an typischen kleinbauenden ROVs näher erläutert.

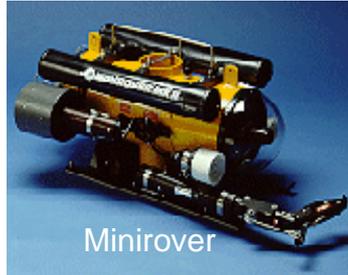
#### Fahrzeugaufbau:

Das Fahrzeug besteht in der Regel aus einem zylindrischen Grundkörper, der zur Aufnahme der Steuerungselektronik dient (Abb. 2). Am Grundkörper werden die Antriebe montiert, die sich in druckdichten Gehäusen befinden. Über Kabelverbindungen werden sie an die zentrale Steuerung im ROV angebunden. Die elektrische Antriebsleistung

reicht dabei von wenigen Watt bis zu knapp einem Kilowatt, wobei zwischen drei bis sechs Propellerantriebe zum Einsatz kommen. Alle Fahrzeuge verfügen über eine Zwangstrimmung und damit auch über eine Vorzugslage.



EROV



Minirover



Searover



Firefly



Star



Seamor

Seamor Nuclear



Scallop



Seaotter



Hyball



Spy



Phantom 150



FO

Abb. 2: Auswahl kleinbauender ROVs

Als zusätzliche Sensoren und Aktoren stehen eine oder mehrere Farbkameras sowie Scheinwerfer zur Beleuchtung zur Verfügung. Optional ist für einige Geräte ein Manipulator erhältlich, der die Handhabung von Objekten bis zu einigen Kilogramm erlaubt und sich aus der verfügbaren Antriebsleistung ergibt. Als sensorische Unterstützung stehen ein analoger oder ein elektronischer Kompass, ein Tauchtiefenmesser oder ein Ultraschall-Tiefenmesser zur Verfügung. Das Gewicht der Fahrzeuge beträgt von einigen wenigen bis zu etwa 100 kg. Durch das gewählte Fahrzeugvolumen ist das ROV im Wasser weitestgehend auftriebsneutral gelegentlich auch mit geringen Restauftrieb versehen. Die maximale Einsatztiefe liegt typischerweise zwischen 50 m und 300 m.

### Kabelverbindung:

Das Kabel versorgt das Fahrzeug mit der benötigten elektrischen Energie, überträgt die Steuersignale vom Leitstand und leitet die Videobilder an den entsprechenden Monitor weiter. An das Kabel werden dabei hohe Anforderungen gestellt. Da die Leitungslänge einige hundert Meter betragen kann, ist vor allem auf ein geringes Gewicht und auf einen kleinen Querschnitt zu achten, damit die Störkräfte des Kabels die Navigation des Fahrzeugs möglichst wenig beeinflussen. Bei großen Kabellängen kann die angeströmte Fläche des Kabels um ein Vielfaches über der des Fahrzeuges liegen.

Aus dem gleichen Grund wird das Kabel so ausgelegt, dass es im Wasser auftriebsneutral ist. Das Gewicht des Kabels wird dabei im Wesentlichen durch die Anzahl und den Durchmesser der einzelnen Leitungen bestimmt. Durch eine zusätzliche Ummantelung wird das Kabel neutral ausgetrimmt. Der Kabeldurchmesser nimmt entsprechend mit der Anzahl der Leitungen ab. Auf Grund dieser Tatsache wird das Kabel meist als Zweidrahtleitung ausgelegt, wobei entweder eine Koaxial- oder Twisted-Pair-Ausführung zum Einsatz kommt. Die Spannungsversorgung erfolgt direkt über das Versorgungskabel, während die Steuer-, Sensor- oder Videosignale aufmoduliert werden. Nachteilig bei dieser Lösung sind die zusätzlichen Kosten für die Modulatoren und Demodulatoren. Um die Kabel während des Einsatzes auf- und abwickeln zu können, kommen meist Kabeltrommeln mit Schleifringübertragern zum Einsatz.

### Leitstand und Steuerung:

Der Leitstand besteht aus der Energieversorgung, einem Monitor sowie einer Bedienkonsole (Abb. 3). Diese beinhaltet dabei meist Joysticks, Taster sowie gelegentlich ein Display. Als Videomonitor werden oft Fernseher und zur Aufzeichnung auch Videorecorder verwendet, da diese standardmäßig über einen Demodulator verfügen und somit direkt eine Rückgewinnung der aufmodulierten Videosignale erlauben.



Abb. 3: Kontrolleinheiten und Steuergeräte verschiedener Hersteller

Optional können bei einigen Geräten Systemdaten z. B. Tauchtiefe oder Kompassrichtung eingeblendet werden. Dazu werden die entsprechenden Messwerte kameraseitig aufbereitet und eingeblendet. Die Leistung der Energieversorgungen reicht von wenigen Watt bis zu etwa 3 kW, wobei meist hohe Spannungen von 130 V bis 380 V Verwendung finden, um die Kabelverluste gering zu halten. Als regelungstechnische Unterstützung werden bei einigen Geräten eine automatische Tauchtiefenregelung sowie eine Kursüberwachung angeboten.

### 2.2.3 ROVs für den Einsatz in kerntechnischen Anlagen

Da ROVs insbesondere dort eingesetzt werden, wo der Einsatz von Tauchern aufgrund der extremen Umgebungsbedingungen unwirtschaftlich ist, fanden kleinbauende Inspektionsfahrzeuge auch den Weg in die gefluteten Sektionen kerntechnischer Anlagen. Speziell für den Einsatz in kerntechnischen Anlagen wurden z. B. die ROVs E-ROV, Seamor Nuclear, Phantom150, Firefly, Dragonfly und MiniRover MK I und MiniRover MK II entwickelt. Dafür sind sie zum Teil mit strahlenresistenter Elektronik (z. B. Kameras) ausgestattet und sind konstruktiv bedingt leichter zu dekontaminieren.

Die Steuerungen haben zum Teil einen höheren Autonomiegrad. So bietet die Fa. Deep Ocean Engineering neben einer Bedienkonsole zur rein manuellen Führung auch eine automatische Tiefenregelung für die ROVs Firefly, Dragonfly und Phantom 150 an /21/.

Eine Besonderheit bildet dabei E-ROV der Fa. Benthos. Es verfügt über eine Ultraschallbasierte xyz-Positionserfassung, die im Arbeitsraum installiert wird. Dieses System gestattet die Programmierung einer Bahn und damit auch eine vollautomatische Steuerung des ROVs /24/.

In Tabelle 1 sind nachfolgend die technischen Daten einiger ROVs zusammengestellt.

Tabelle 1 : Technische Daten einiger kleinbauender ROVs (/18/-/28/)

Name Hersteller	L*B*H, Masse, Tauchtiefe	ROV-Ausstattung	Steuerung
<b>FO</b> Mariscope	1000* 500* 440 mm, 30 kg, 300 m	Licht: 2*20 W/ Antriebe: 3, 800 W horizontal, 400 W vertikal, 4 Knoten/ Farbkamera, Kompass	Energie: 230 VAC, 50 HZ/ Kabel: HF koaxial 2-pol. bis 300 m/ Monitor: TV/ Steuerung: Manuell, Bedienkonsole
<b>Spy</b> Mariscope	780* 900* 930 mm, 55 kg, 300 m	Licht: 2*50 W/ Antrieb: 4*400 W 2 horizontal, 2 vertikal, 2 Knoten/ Farbkamera, Kompass, Manipulator (optional)	Energie: 230-380 VAC, 50 Hz, 1.8 kW/ Kabel: HF koaxial 2-pol. bis 300 m/ Monitor: TV/ Steuerung: Manuell, Bedienkonsole
<b>Star</b> Mariscope	800* 500* 500 mm, 28 kg, 300 m	Licht: 50 W/ Antrieb: 800 W horizontal, 400 W vertikal, 4 Knoten/ Farbkamera, Kompass	Energie: 230 VAC, 50 HZ/ Kabel: HF koaxial 2-pol. bis 300 m/ Monitor: TV/ Steuerung: Manuell, Bedienkonsole
<b>Hyball</b> Hydrovision	535* 650* 565 mm, 42 kg, 300 m	Licht: 2 Scheinwerfer/ Antriebe: 6, 4 Knoten/ Farbkamera, Magnet- und Kreiselkompass	Energie: 240 - 110 VAC 50,60 Hz, 2.5 KW/ Kabel: bis 400 m/ Steuerung: Manuell Bedienkonsole/ automatische Tiefenregelung u. Kurskontrolle
<b>Seaotter</b> J.W. Fishers	820* 520* 380 mm, 38 kg, 80 m	Licht: 4*100 W/ Antriebe: 4/ Farbkamera/ Digital Kompass	Energie: 220 V/ Steuerung: Manuell, Bedienkonsole

Name Hersteller	L*B*H, Masse, Tauchtiefe	ROV-Ausstattung	Steuerung
<b>Seamor</b> Inuktum Ser- vices	350* 265* 225 mm, 8,1 kg, 50 m	Antriebe: 4, 1,5 Knoten/ Farbkamera/ Manipulator	Energie: 120 V, 60 Hz, 5 A Steuerung: Manuell, Bedienkonsole
<b>Phantom- 150 (*)</b> Deep Ocean Engineering	643* 356* 273 mm, 14 kg, 300 m	Licht 2*250 W/ Antriebe: 4, 2,5 Knoten/ Farbkamera, Kompass, Tauchtiefenmes- ser, Tiefenmesser, opt. Mani- pulator	Energie:120,240 V, 50/60 Hz, 1 kVA/ Steuerung: Manuell, Bedi- enkonsole; automatische Tiefenrege- lung
<b>Firefly (*)</b> Deep Ocean Engineering	190* 153* 356 mm, 5,4 kg, 46 m	Licht: 4*6 W, 2*20 W/ Antriebe: 4, 2 vertikal, 2 hori- zontal/ Strahlungsresistente Farbkamera	Energie: 120/240 V, 50/60 Hz, 1 kVA/ Kabel: 53 m/ Steuerung: Manuell, Bedienkonsole; automati- sche Tiefenregelung
<b>Dragonfly (*)</b> Deep Ocean Engineering	254* 584* 116 mm, 16 kg, 46 m	Licht: 2*500 W/ Antriebe: 5, 2 vertikal, 2 horizontal, 1 later- al/ Strahlungsresistente S/W-Kamera/ opt. Manipula- tor	Energie: 120/240 V, 50/60 Hz, 3 kVA/ Kabel: 53 m/ Steuerung: Manuell Bedienkonsole; automati- sche Tiefenregelung
<b>E-ROV (*)</b> Benthos	500* 460* 390 mm, 23.6 kg, 46 m	Licht: 2*500 W/ Antriebe: 5 / Farbkamera, opt. Manipula- tor	Energie: 115/240 V, 50/60 Hz, 3 kW/ Kabel: 76 m, koaxial, Steue- rung: Manuell oder vollautomati- sche xyz-Steuerung, Positionssystem: 2 Pinger Rov/ mehrere im Becken
<b>MiniRover MK II (*)</b> Benthos	760* 500* 420 mm, 36 kg, 304 m	Licht:: 2*150 W/ Antriebe: 4, 2 horizontal (5,4 kg), 1 links/ rechts (1,4 kg), 1 vertikal (5 kg) max. 2.9 Knoten/ Kamera, Kompass, Tiefen- messer, 2 Analog-Eingänge (5 V DC/12 bit ADC)	Energie: 100/120/208/240 VAC, 50/ 60 Hz, 1 kW/ Kabel: 152 m o. 456 m/ Steuerung: Manuell, Bedi- enkonsole
<b>Seamor Nuclear (*)</b> Inuktum Ser- vices	325* 250* 188 mm, 13 kg 45 m	Antriebe: 3/ Kamera, Manipu- lator	Energie: 120 V, 60 Hz, 5 A Steuerung: Manuell, Bedienkonsole
<b>Scallop (*)</b> Inuktum Ser- vices	350* 230* 216 mm, 3,6 kg, 76 m	Energie: 36 V/ Licht 2*20 W/ Antriebe: 3, 2 Horizontal, 1 Vertikal, 1,5 Knoten/ Farbka- mera, Kompass	Energie: 115/220 V, 60/50 Hz, 500 W auf Kabel 36 V/ Kabel: 38 m/ Steuerung: Manuell, Bedienkonsole

(\*) speziell für den Einsatz in kerntechnischen Anlagen

### 2.2.4 Vorteile modular aufgebauter Fahrzeuge

Der ausschlaggebende Faktor für die Entwicklung modular aufgebauter Unterwasserfahrzeuge ist die damit verbundene Möglichkeit, später eine einfache Anpassung an verschiedene Aufgaben vornehmen zu können. Dadurch werden zum einen die Rüstzeiten verkürzt, zum anderen wird die Effizienz eines Fahrzeugsystems entsprechend erhöht /4/.

Beim Einsatz in kerntechnischen Anlagen ergeben sich jedoch weitere Vorteile:

- Stillstands- und Ausfallzeiten werden minimiert, da defekte Funktionselemente einfach durch Austauschmodule ersetzt werden können.
- Vielfach kann auf die kostenintensive Entwicklung strahlenharter Elektronikkomponenten zugunsten konventioneller Baugruppen mit verkürzter Lebensdauer verzichtet werden.
- Exakt spezifizierte Schnittstellen ermöglichen nicht nur die nachträgliche Entwicklung weiterer Module, deren Bedarf u. U. bei der Grundkonzeption noch nicht abzusehen war, sie eröffnen auch Drittanbietern die Chance, Komponenten für das Fahrzeugsystem zu entwickeln. Dies kann wiederum die Marktposition des Grundsystems erheblich festigen.

Bei der Entwicklung der zum Einsatz kommenden ROVs müssen eine Vielzahl von Randbedingungen berücksichtigt werden:

- dekontaminationsfreundliche Konstruktion
- geringe Abmessungen
- hohe Beweglichkeit für exakte Manöver in beengten Räumen
- Wassertiefen bis maximal 30 m
- kurze Rüstzeiten
- gesicherte Rückholbarkeit des Fahrzeugs
- Sicherheit gegen Ausfall

Im Rahmen der Forschungsarbeiten wurden am Institut für Werkstoffkunde der Universität Hannover zwei unterschiedliche Ansätze für einen modularen Aufbau untersucht. Nachfolgend werden zwei Fahrzeuge näher beschrieben.

### 2.2.5 Unterwasser-Robotersystem FAUST III

FAUST III (Abb. 4) ist ein fernbedienbares Unterwasser-Handhabungssystem für Inspektions-, Montage- und Demontageaufgaben in kerntechnischen Anlagen und anderen Einrichtungen mit Unterwasser-Installationen /29/,/30/. Die maximale Tauchtiefe beträgt 50 m. Das System ist in 5 Achsen steuerbar und kann durch die 8 elektrisch betriebenen Propeller und seine lageunabhängige Trimmung in jeder Lage und Orientierung (Seitenlage, Überkopf, etc.) arbeiten. Mit dem Raupenantrieb ist ein präzises Verfahren auf Oberflächen möglich. Die Steuerung erfolgt mit einem Computer an Bord und einem Leitreechner als Schnittstelle zum Bediener. Der Bediener steuert die Bewegungen mit einer Tastatur und einer Multifunktions-Steuerkugel (6-Achsen-Joystick) am

Leitrechner, deren Auslenkungen vom Bordcomputer in die jeweiligen Steuerbefehle für die Antriebe umgerechnet werden.

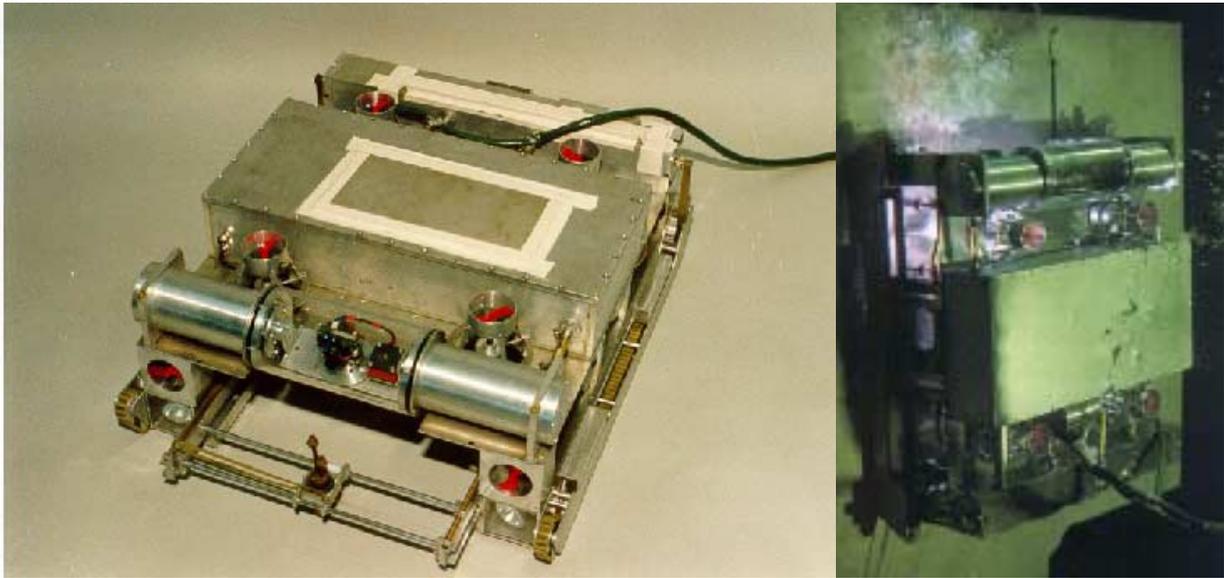


Abb. 4: ROV FAUST III in Ruheposition (l.) und beim Unterwasserplasmaschneiden (r.)

Auf der grafischen Benutzeroberfläche werden die wesentlichen technischen Daten visualisiert. Sämtliche Versorgungs- und Signalleitungen sind in einem Strang zusammengeführt, welcher bei einem Systemausfall als Bergungsleine dient. Aufgrund der kompakten Aluminiumbauweise hat das Fahrzeug bei seinen Maßen von 790 x 700 x 305 mm nur ein Gesamtgewicht von 42 kg. Ausgestattet mit Kameras, Scheinwerfern und Sensoren kann FAUST III Inspektions- und messtechnische Arbeiten unter Wasser durchführen. Der Bediener kann die Bewegungen und Arbeiten durch das zum Leitreechner übertragene Kamerabild überwachen.

An der integrierten 2-Achsen-Führungsmaschine können im Wechsel ein kleinbauender Plasmaschneidbrenner zur Zerlegung von Stahlkomponenten, diverse Sensoren oder ein Greifer mit einer max. Tragfähigkeit von 1 kg adaptiert werden.

Als Steuerungskonzept wird ein 2-Rechner-System bestehend aus Bordrechner und Leitreechner verwendet. Die Bedienung erfolgt über den Leitreechner, während die Ansteuerung der Aktoren und Sensoren über den echtzeitfähigen Bordrechner des Fahrzeugs erfolgt. Die gesamte Steuerungselektronik ist in einem zentralen Gehäuse im Unterwasserfahrzeug untergebracht. Die Spannungsversorgung erfolgt in Niederspannungstechnik, die Kommunikation und die Videodatenübertragung wird über Lichtwellenleiter realisiert. Weitere Elektronikbaugruppen können modular an die zentrale Steuerung über das Bussystem des Bordrechners angebunden werden.

### 2.2.6 Modulares Unterwasserfahrzeug $\mu$ -FAUST

Das Inspektionsfahrzeug  $\mu$ -FAUST soll für Inspektionen an schwer zugänglichen Stellen, beispielsweise im Inneren von Rohren, eingesetzt werden [4],[31]. Damit sich das Fahrzeug bei Inspektionsfahrten nicht an Hindernissen verhaken kann und der maximal zulässige Querschnitt gut ausgenutzt wird, wurde  $\mu$ -FAUST aus rotationssymmetrischen Modulen aufgebaut. Die Grundkonfiguration besteht aus dem Frontmodul, einem Elektronikmodul mit einer Mikrocontrollereinheit zur Fahrzeugsteuerung, einem Zweistrahl-

anlagenmodul und dem Heckmodul mit dem Hauptpropeller. Das Frontmodul enthält eine bewegliche CCD-Videokamera in einer Plexiglkuppel, die bündig mit dem tragenden Aluminiumsegment verklebt ist. Die einzelnen Module sind untereinander durch Dichtelemente und eine durchgängige Verschraubung in Richtung der Längsachse verbunden.

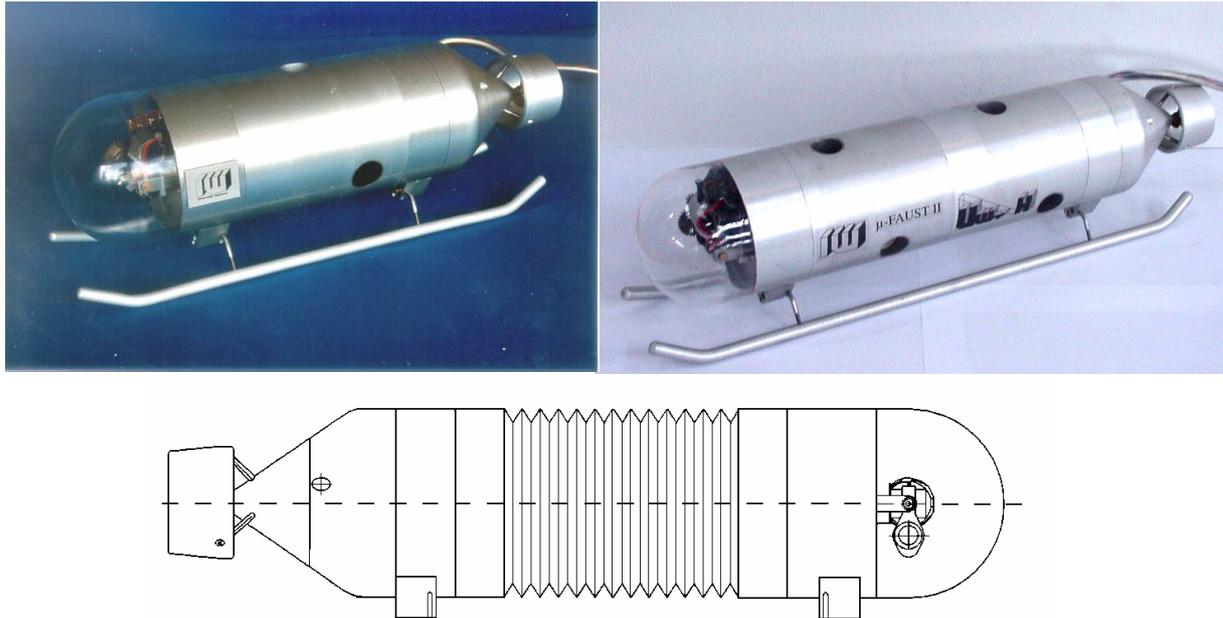


Abb. 5:  $\mu$ -FAUST mit einem Antriebsmodul (l.o.), zwei Antriebsmodulen (r.o.) und mit Balgenmodul (u.)

Das Zweistrahlanlagenmodul enthält je eine Vertikal- und Horizontalstrahlanlage. Mit ihrer Hilfe taucht das aus Sicherheitsgründen leicht positiv ausgetrimmte Fahrzeug und führt Drehungen um seine Hochachse aus. Die Horizontalstrahlanlage entfaltet ihre Kraftwirkung in Höhe des Fahrzeugschwerpunktes, so dass nahezu kein unerwünschtes Moment um die Längsachse erzeugt wird. Das Heckelement enthält die Anschlüsse für die Versorgungsleitung sowie den Antriebsmotor für den Hauptpropeller, der geschützt in einer Kortdüse am Heck des  $\mu$ -FAUST angebracht ist.

Das Fahrzeug kann modular erweitert werden, z. B. kann ein zweites Zweistrahlanlagenelement vor dem Heck eingesetzt werden, ohne den Gesamtquerschnitt zu verändern. Durch das zusätzlich vorhandene Vertikalstrahlruder sind Neigungsbewegungen um die Querachse möglich. Das zweite Horizontalstrahlruder erlaubt Seitwärtsbewegungen und die Drehung des Fahrzeugs auf engstem Raum.

Volumen und Masse der einzelnen Module wurden so aufeinander abgestimmt, dass der Trimm des  $\mu$ -FAUST bei einer Erweiterung erhalten bleibt. Soll das Fahrzeug beispielsweise zur Bergung von Kleinteilen eingesetzt werden, dann muss der Gesamtauftrieb unter Wasser automatisch reguliert werden können. Bemannte Unterseeboote verwenden hierzu Tauchtanks, die entsprechend geflutet oder gelenzt werden. Da in einer kerntechnischen Anlage mit radioaktiven Partikeln im Wasser gerechnet werden muss, wird für den  $\mu$ -FAUST eine dekontaminationsfreundlichere Lösung benötigt. Eine Möglichkeit bildet das in der Abb. 5 dargestellte Faltenbalgmodul. Es ermöglicht eine Anpassung des Fahrzeugvolumens an das veränderte Gewicht und lässt sich nach dem Einsatz von gut außen reinigen.

Da die eingesetzte Mikrocontrollereinheit nicht beliebig erweiterungsfähig ist, besteht die Notwendigkeit, bei neu hinzukommenden Komponenten auch zusätzliche Controllereinheiten anzubinden. Die Kommunikation erfolgt daher über eine serielle Ringstruktur, wobei die einzelnen Teilnehmer von der Steuerungseinheit über ein aufgesetztes Protokoll adressiert und angesteuert werden.

### **2.3 Motivation für die Neuentwicklung eines modularen ROVs**

Am Markt verfügbare kleinbauende ROVs sind geschlossene Systeme. Diese sind in der Regel nur durch den Hersteller konfigurierbar, wobei jeder Hersteller sein eigenes Steuerungskonzept einsetzt. Eine Austauschbarkeit und Verwendbarkeit von Modulen unterschiedlicher Hersteller ist damit nur in Ausnahmefällen ohne Erweiterung oder Umbau möglich. Der Endanwender kann das Fahrzeug daher typischerweise nur vom Hersteller konfigurieren und erweitern lassen.

Aber auch eine Erweiterung stellt damit immer ein Problem dar, wenn der Hersteller solche Möglichkeiten nicht von Anfang an im Konzept vorgesehen hat. Dabei lässt sich die mechanische Integration neuer Komponenten noch relativ einfach durch die Montage am Rahmen des ROVs realisieren. Auf Seiten der Steuerung muss aber ggf. eine komplette Neuentwicklung erfolgen, wenn die entsprechenden Schnittstellen nicht vorgesehen sind. Der Hauptnachteil ist das Fehlen von einheitlichen Schnittstellen zur einfachen Integration von Komponenten. Im Bereich der Automatisierungstechnik sind diese Probleme seit Jahren durch den Einsatz von Feldbussystemen gelöst. Die Steuerung des Fahrzeugs über eine Bedienkonsole ist zwar kostengünstiger als der Einsatz eines PCs, allerdings fehlt dann, gerade für die Erfassung und Speicherung von Messdaten, eine geeignete Anbindung.

Der modulare Aufbau der Systeme bezieht sich nur auf einzelne Funktionsgruppen aber nicht auf vollständige Funktionsmodule. Die zu einem Modul gehörigen Komponenten verteilen sich dabei oft über das gesamte System. Zusätzlich sind sie häufig räumlich getrennt, wodurch ein Austausch für Wartung und Reparatur zusätzlich erschwert wird. Dieser Ansatz entspricht der klassischen zentralen Steuerung von Geräten.

Auch das Unterwasserfahrzeug FAUST III hat auf Grund der fest installierten Komponenten einige Schwachstellen, die im Bereich der Wartung und Umrüstung liegen. Dies soll beispielhaft an Hand der Installation eines neuen externen, also außerhalb des Elektronikgehäuses liegenden, Sensors mit zusätzlicher Hardware erläutert werden. Dazu sind folgende Schritte notwendig:

1. Befestigung am Fahrzeugrumpf, ggf. Anbringen einer Halterung
2. Bereitstellung der notwendigen Kabeldurchführung bzw. Steckverbindung zum Elektronikgehäuse (Anbringen einer Bohrung, Einbau eines Steckverbinders)
3. Anbindung der Sensorelektronik an das nicht standardisierte, FAUST III-spezifische Bus- und Energiesystem des Fahrzeugs (Öffnen und Schließen des Deckels mit ca. 30 Schrauben)
4. Integration der zusätzlichen Steuer- und Verarbeitungssoftware im Bordrechner sowie deren Test im Zusammenspiel/Parallelbetrieb mit den übrigen Komponenten (Neuerstellung einer Epromversion des Steuerungsprogramms)

## 5. Einbindung der Sensorfunktion in die PC-Software (Programmierung der Bedienoberfläche, Kommunikation etc.)

Wie dieses einfache Beispiel zeigt, berührt die Änderung der Systemkonfiguration alle mechanischen, elektronischen, steuerungs- und softwaretechnischen Bereiche. Zusätzlich ist das Fahrzeug während der gesamten Umbauzeit oder auch bei Wartungs- und Reparaturarbeiten nicht verfügbar.

Bei  $\mu$ -FAUST wurde der Druckkörper des Trägerfahrzeugs selbst in einzelne Module unterteilt, die dann beispielsweise die erforderlichen Strahlruder mitsamt den zugehörigen Antrieben enthalten. Die Modularität bezieht sich im Wesentlichen auf Zusatzmodule der Antriebs- bzw. Auftriebstechnik. Der Aufbau muss stark an die zylinderförmige Rumpfstruktur angelehnt sein, damit das Modul integriert werden kann. Auch der Umbau gestaltet sich relativ aufwendig und zeitintensiv. Bei beiden FAUST-Fahrzeugen fehlen einheitliche bzw. standardisierte Schnittstellen.

Kein kleinbauendes Fahrzeug verfügt standardmäßig über eine 3D-Visualisierungssoftware, die den Bediener bei der Durchführung von Arbeiten unterstützt. Die Geräte weisen meist einen geringen Autonomiegrad auf und sind häufig nur rein manuell zu führen.

Im Rahmen der Arbeit soll ein hochmodulares Unterwasserfahrzeug entwickelt werden, welches eine hohe Bedienungsfreundlichkeit aufweist und durch den Bediener nahezu frei konfiguriert und in extrem kurzer Rüstzeit in Betrieb genommen werden kann. Bei dem Fahrzeug TRIBUN soll ein dezentrales Steuerungskonzept eingesetzt werden, bei dem die einzelnen Module unabhängig voneinander entwickelt, getestet und angebunden werden können. Es muss daher über technische Merkmale verfügen, die über den gegenwärtigen Stand der Technik von kleinbauenden Unterwasser-Handhabungssystemen hinausgehen.

Als Analogie sollen hier Entwicklungen im Bereich der Personal Computer (PC) aufgeführt werden. Zu Beginn des PC-Zeitalters wurden von vielen Firmen Entwicklungen auf den Markt gebracht. Zu nennen seien hier exemplarisch Commodore, Apple oder auch Atari. Diese Computer zeichneten sich im wesentlichen durch die vom Hersteller spezifizierten Eigenschaften aus. Anwender, die die Funktionalität gemäß eigener Wünsche erweitern wollten, waren zunächst auf die Entwicklung eigener Lösungen angewiesen. Dabei setzten diese Entwicklungen auf den vom Computer bereitgestellten parallelen (Drucker-Port) oder seriellen Schnittstellen auf, die aber auf Grund ihrer technischen Eigenschaften für viele Anwendungen ungeeignet waren.

Durchgesetzt hat sich auf Grund der einfachen Erweiterbarkeit daher langfristig der PC-Typ, wie er auch heute überwiegend Verwendung findet. Standardisierte Bussystem-Schnittstellen wie ISA (Fa. IBM, 1981 eingeführte standardisierte PC-Bus-Technologie) oder PCI (Fa. Intel, 1991) erlauben durch die flexible Anbindung von Einsteckkarten im sog. Motherboard die Einsatzmöglichkeiten individuell zu erweitern.

In der Anfangsphase war die Installation von Rechnern oft nur von Personen mit tiefem Grundwissen durchführbar. Zu nennen sei hier z. B. die Einstellung der Adressen oder die Belegung der Interrupts. Diese Lücken versuchte Microsoft mit dem Betriebssystem Windows 95 und der Einführung der sog. Plug&Play-Technik zu schließen. Dabei führt der Rechner beim Systemstart eine Hardwareerkennung durch und sucht nach vorhandenen oder neuen Komponenten. Werden diese gefunden, wird der Benutzer aufgefordert, die entsprechende Software (Treiber und Anwenderprogramme) zu instal-

lieren. Der Aufwand für die Installation und besonders die Fehlerrate sinkt dadurch erheblich, da die Maschine dem Menschen Auskunft über die Konfiguration des Systems gibt und nicht umgekehrt. Die Aufgabe des Betreibers reduziert sich dabei im Wesentlichen auf den Zusammenbau der Komponenten; alle weiteren Schritte werden dann nahezu vollautomatisch durchgeführt. Auch wenn dieses Konzept in der praktischen Umsetzung oft Probleme aufweist, so ist es doch als richtungsweisend für technische Systeme anzusehen.

In technischen Bereichen kommen immer häufiger Steuerungen zum Einsatz, die über Feldbussysteme miteinander verbunden werden. So können heutzutage ganze Produktionslinien aus unterschiedlichen Produkten verschiedener Hersteller aufgebaut werden. Im Gegensatz dazu finden gerade bei kleinen Handhabungssystemen häufig immer noch geschlossene Systeme Verwendung, die allein vom Hersteller erweiterbar sind.

Moderne Entwicklungen, besonders im dem Bereich der Mikroelektronik und der Feldbusteknik sowie die ständig sinkenden Preise bei gleichzeitig gesteigerter Funktionalität, lassen den Einsatz von Mikrocontrollern oder Einplatinenrechnern so auch unter dem Aspekt der Kosten lohnenswert erscheinen.

Durch die konsequente Umsetzung einer modularen Bauweise und den portierbaren Modulaufbau besteht zusätzlich der Vorteil, einzelne Module problemlos zwischen unterschiedlichen Handhabungssystemen auszutauschen.

Überträgt man die obigen Entwicklungen und Ideen aus der PC-Technik in allen Bereichen auf die Handhabungstechnik, so erhält man eine Art Baukastensystem für Handhabungssysteme bzw. im hier vorliegenden Fall für ein Unterwasserfahrzeug. Im Einzelnen bedeutet dies:

- Motherboard und Einsteckkarten: Das System soll hochmodular, bestehend aus einem Basissystem und Funktionsmodulen, aufgebaut werden.
- Adaption einer PC-Einsteckkarte: Die Komponenten müssen am Unterwasserfahrzeug einfach adaptiert werden können. Dies setzt die Entwicklung einer einheitlichen elektromechanischen Verbindungskupplung voraus.
- ISA/PCI-Steckplatz: Die Module sollen über die elektromechanische Verbindung ebenfalls steuerungstechnisch angesprochen werden. In der Automatisierungstechnik erfolgt dies in der Regel über ein Feldbussystem. Der Steckplatz soll weiterhin über eine Kennung verfügen, die vom Modul erkannt wird. Dies ist erforderlich, da die Module in Abhängigkeit von der Position unterschiedlichen Einfluss auf das Fahrzeug haben.
- Offenes System: Da die oben genannten Schnittstellen und deren Spezifikationen allgemein bekannt und zugänglich sein sollen, handelt es sich um ein offenes System.
- Plug&Play: Die Steuerung erkennt die angeschlossenen Module und stellt automatisch deren Funktionalität bereit. Als Einschränkung gegenüber der PC-Technik soll auf eine Installation von modulspezifischer Software durch den Nutzer analog zur Installation von Treibern verzichtet werden. Die Module sind Sonderentwicklungen, deren Funktionalität im System bekannt sein sollen.
- Hoher Verbreitungsgrad: Um auf kosten- und zeitintensive Eigenentwicklungen verzichten zu können, soll soweit möglich auf vorhandener Technik und gebräuchlichen

am Markt verfügbaren Lösungen aufgesetzt werden.

- **Wartung, Bedienerunterstützung:** Zur Unterstützung des Bedieners müssen für die Inbetriebnahme und Wartung komplexe Analysefunktionen implementiert werden. Als zusätzliches System soll eine 3D-Visualisierung entwickelt werden. Alle Lösungen müssen dabei dem Plug&Play-Charakter gerecht werden.

Die Konzeption, die Realisierung, die Inbetriebnahme und der Test des Unterwasser-Handhabungssystems TRIBUN wurde unter Berücksichtigung dieser Vorgaben durchgeführt.

### 3 Konzeption und Aufbau des Systems

Um die anspruchsvollen Aufgaben, insbesondere bei Wartung, Instandhaltung, Überwachung sowie beim Rückbau kerntechnischer Anlagen, zuverlässig ausführen zu können, werden die Entwicklungsarbeiten im wesentlichen in mehreren Schwerpunktgebieten parallel vorangetrieben.

Das Systemkonzept umfasst ein fernbedientes Unterwasserfahrzeug einschließlich der an Bord untergebrachten Subsysteme für Antrieb, optische Beobachtung, Lageerfassung, Energieversorgung u. ä. sowie alle notwendigen Module für die späteren Arbeitseinsätze. Die Integration der Handhabungseinrichtungen wie z. B. Manipulatoren oder Greifer vervollkommen die Spezifikation des Handhabungssystems. Die Untersuchung der Einsatzmöglichkeiten moderner Eingabegeräte zur Bedienung und Steuerung dienen der Optimierung der Handhabung.

Die Konstruktion umfasst den Entwurf des Tauchfahrzeuges sowie dessen Komponenten wie Antriebe, Roboterarme und Werkzeuge sowie die Auswahl geeigneter Materialien. Die konstruktive Auslegung erfolgt konform zu den üblichen Entwurfs- und Konstruktionsverfahren und wird innerhalb dieser Arbeit nicht näher erläutert /34/,/35/.

Die Steuerungstechnik und Architektur ist ein weiterer elementarer Bestandteil des Systems. Hierzu zählen die Auswahl und die Entwicklung der gesamten Steuerungselektronik der Steuerungskomponenten, die Auswahl eines Bussystems, aber auch die Aufteilung von Steuerungsfunktionen auf das Gesamtsystem. Die Grundlagen für den Entwurf elektronischer Komponenten und die Implementierung von Reglern werden ebenfalls nicht näher vorgestellt /36/,/37/,/38/,/39/,/40/,/41/,/42/.

Im zunehmenden Maße ist während des Einsatzes eine leistungsfähige 3D-Visualisierung erforderlich, da der Bediener in der Regel keinen direkten Einblick in den Arbeitsraum hat. Die messtechnische Erfassung der Fahrzeugposition und Lage sowie die Modellierung des Fahrzeugs und der Arbeitsumgebung sollen eine 3D-Visualisierung ermöglichen. Die 3D-Visualisierung soll auch genutzt werden, um komplexe Arbeitsabläufe und Aufgaben im Vorfeld zu simulieren und die Bediener zu trainieren. Damit soll das Risiko von Fehlbedienungen während des Einsatzes deutlich reduziert werden.

#### 3.1 Systemanforderungen

Das Handhabungssystem wird für die Durchführung von Arbeiten unter Wasser speziell in kerntechnischen Anlagen mit radioaktiv belasteten Bereichen ausgelegt. Es soll Montage-, Prüf-, Bergungs- und Beobachtungsaufgaben ausführen und den Gegebenheiten und den speziellen Anforderungen der Aufgabenstellung entsprechend aus- und umrüstbar sein. TRIBUN stellt damit die direkte Weiterentwicklung der am Titelinstitut entwickelten Fahrzeuge FAUST III und  $\mu$ -FAUST dar.

Um diese breitgefächerten Einsatzmöglichkeiten realisieren zu können, wird das Unterwasserfahrzeug TRIBUN konsequent modular aufgebaut. Die Zeitdauer der Revisionsarbeiten in kerntechnischen Anlagen bestimmt unter anderem deren Verfügbarkeit. Längere Ausfallzeiten müssen aufgrund der hohen Stillstandskosten vermieden werden. Ein modularer Aufbau des Fahrzeugs ermöglicht neben einer flexiblen, den speziellen Anforderungen entsprechenden Umrüstbarkeit auch einen schnellen Austausch defekter Funktionselemente und sichert damit eine hohe Verfügbarkeit des Handhabungssystems.

Das Unterwasserfahrzeug soll in seiner Grundausstattung über ein Kamerasystem sowie zwei Propellerdoppelantriebsmodule verfügen, die jeweils einen Antrieb für den horizontalen und den vertikalen Schub liefern. Für die Durchführung von Arbeiten in allen Raumlagen ist es notwendig, dass am Trägersystem zusätzliche Antriebsmodule adaptiert werden können.

Die konstruktiven Anforderungen an das Basissystem des TRIBUN ergeben sich aus den Arbeiten unter Wasser, speziell in kontaminierten Bereichen kerntechnischer Anlagen. Für diesen Einsatzbereich ist das System dekontaminationsfreundlich zu gestalten und ist auf Grund seines Unterwassereinsatzes gegen Korrosion auszulegen. Weiterhin sind alle Bauteile so zu konstruieren, dass ihre Oberflächen leicht zu reinigen sind und Konstruktionsspalten, Hinterschneidungen und offene Kanäle, in denen sich Flüssigkeiten sammeln können, vermieden werden.

Bei einem Totalausfall der Stromversorgung ist ein definierter Restauftrieb vorzusehen, der das Fahrzeug an die Oberfläche treiben lässt. Um das Trägersystem aus dem Gefahrenbereich zu bergen, wird das Versorgungskabel gleichzeitig als Sicherungs- bzw. Bergungsleine eingesetzt.

Die Elektronik der strahlenexponierten Baugruppen im Fahrzeug wird mittels konventioneller Komponenten realisiert, da der Einsatz von strahlenresistenten Bauelementen äußerst kostenintensiv ist. Da dadurch ein erhöhtes Ausfallrisiko verbunden ist, müssen alle Komponenten in und am Fahrzeug in sehr kurzer Zeit austauschbar sein. Hier kommt der konstruktiven Auslegung der Verschluss- und Verbindungsmechanismen verstärkte Bedeutung zu.

### **3.2 Systemkonzept**

Das grundlegende Konzept, das innerhalb dieser Arbeit realisiert wird, gliedert sich in vier Bereiche. Das eigentliche Fahrzeug besteht aus einem Grundkörper mit einigen fest installierten Komponenten. Verschiedene Funktions- und Werkzeugmodule bilden den zweiten Bereich. Das Fahrzeug soll über ein Versorgungskabel mit dem Leitstand verbunden werden. Der Leitstand, als dritter Bereich, beinhaltet dabei Systeme, wie die Energieversorgung, alle notwendigen Module zur Steuerung, wobei insbesondere bei Test und Inbetriebnahme ein PC mit entsprechenden Eingabegeräten zum Einsatz kommen soll. Unter den Aspekten einer verbesserten Fernhantierbarkeit soll der Begriff Leitstand dahingehend erweitert, dass die Komponenten der Steuerung nicht zwingend physikalisch an einem Ort installiert werden müssen. Insbesondere soll hierbei Wert auf eine Einbeziehung von netzwerk-basierten Techniken gelegt werden. Der vierte und damit letzte wichtige Bereich ist die Bedienerunterstützung durch Maßnahmen, wie steuerungs- und regelungstechnische Unterstützung, Maßnahmen zur Positions- und Lageerfassung des Fahrzeugs sowie die Anbindung einer 3D-Visualisierung an das System. Um eine Kontrolle des gesamten Systems zu gestatten, soll die Steuerung dabei auch einen Zugriff auf die Komponenten ermöglichen, die sich im Arbeitsraum befinden. Dies könnten beispielsweise zusätzliche Kameras, Beleuchtungseinheiten oder ein Positionssystem sein. Abb. 6 gibt nachfolgend einen Überblick über die Komponenten und deren Aufteilung im System. Die im Fahrzeugumpf installierten Module werden dabei als interne Module und die an den Multifunktionsleisten adaptierten als externe Module bezeichnet.

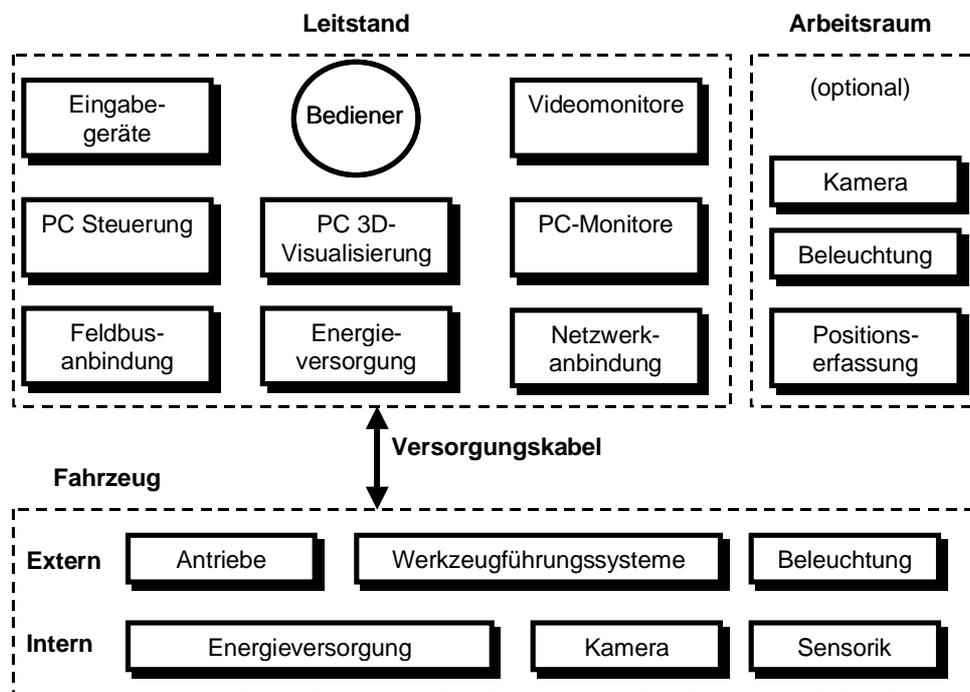


Abb. 6: Komponenten des Systems

### 3.3 Mechanisches Konzept und Aufbau des Fahrzeugs

Die äußere Geometrie basiert auf einem zylindrischen Druckkörper mit Anbauteilen. Innerhalb des Rumpfes werden einige fest installierte Elektronikmodule integriert. Der vordere Teil wird mit einer transparenten Plexiglashalbkugel abgeschlossen. Durch sie können mit einem Kamera-System Beobachtungsaufgaben durchgeführt werden. Eine Kabelverbindung versorgt das Unterwasserfahrzeug mit elektrischer Energie. Zudem werden vom und zum Leitstand Steuerungssignale an die einzelnen Module des Basissystems und die Videosignale eines Kamerasystems zum Leitstand übertragen.

Für eine gemeinsame Grundlage aller nachfolgenden Darstellungen wird zunächst ein Koordinatensystem eingeführt. Unterwasserfahrzeuge verfügen prinzipiell über die Möglichkeit der Bewegung innerhalb von sechs Freiheitsgraden. Dies sind Translationen entlang der Achsen eines kartesischen Koordinatensystems und die Rotationen um diese. Zur Beschreibung der Bewegungsvorgänge ist es sinnvoll, ein körperfestes Koordinatensystem einzuführen. Dabei werden folgende im Schiffbau übliche Bezeichnungen verwendet /43/:

- x bezeichnet die Längsachse des Fahrzeuges,
- y bezeichnet die Querachse des Fahrzeuges,
- z bezeichnet die Vertikalachse des Fahrzeuges,
- $\alpha$  bezeichnet die Drehung um die x-Achse, den Rollwinkel,
- $\beta$  bezeichnet die Drehung um die y-Achse, den Neigungswinkel und
- $\gamma$  bezeichnet die Drehung um die z-Achse, den Gierwinkel.

Der Ursprung des Koordinatensystems wird auf den Schnittpunkt der Schnittlinie der Symmetrieebenen mit der Oberfläche der Verschlussplatte des Systems gelegt. Dabei spannen die  $xy$ -,  $xz$ - und die  $yz$ -Ebenen die Symmetrieebenen auf. In Abb. 7 ist das Koordinatensystem des Fahrzeuges dargestellt.

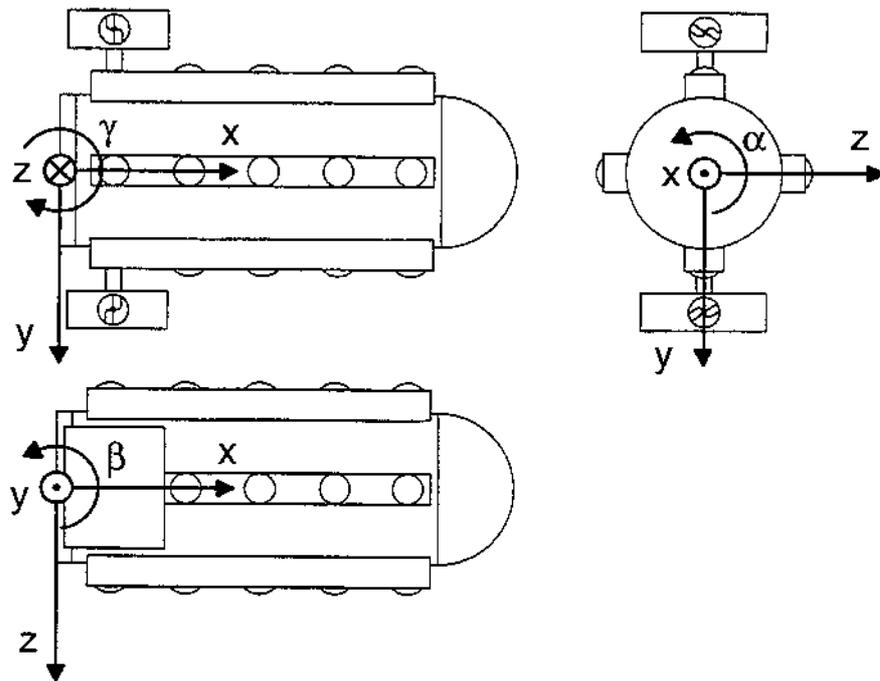


Abb. 7: Koordinatensystem des Unterwasserfahrzeugs TRIBUN

Das Basissystem (Abb. 8) hat die Aufgabe, ein Kamerasystem, die interne Sensorik und Teile der Energieversorgung aufzunehmen und vor dem Eindringen von Wasser zu schützen. Zudem dient es als Aufnahme und Transporteinheit von anforderungsspezifischen Manipulator-, Scheinwerfer- und Antriebs-Modulen. Damit bildet die Trägereinheit ein Kernstück des gesamten Systems.

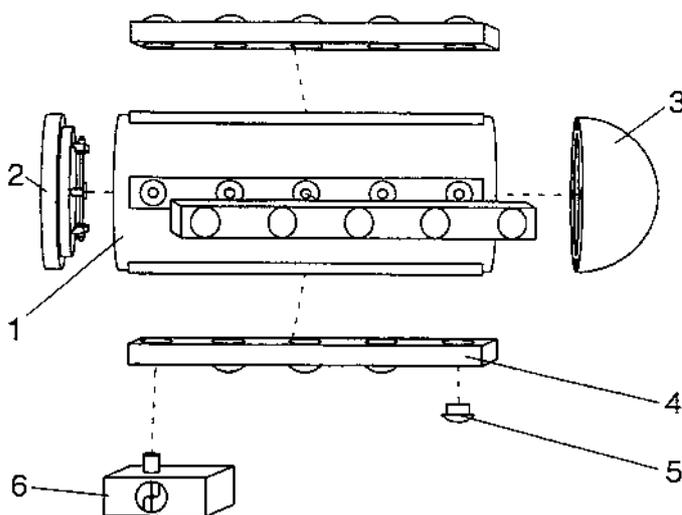


Abb. 8: Mechanischer Aufbau des Basissystems

Das Unterwasserfahrzeug soll in radioaktiv belasteten Bereichen, wie sie z. B. in kerntechnischen Anlagen vorhanden sind, in Wassertiefen bis maximal 40 m eingesetzt werden können. Der Rumpfkörper besteht daher aus einer AlMgSi-Legierung. Damit lässt sich das Unterwasserfahrzeug in unveränderter Bauform auch für den Einsatz im Salzwasser einsetzen.

Der Rumpfkörper (Pos.1) des Trägerfahrzeugs ist zylindrisch ausgeführt (Länge=500 mm, Durchmesser=200 mm). Die Stirnflächen des Zylinders schließen das Heck mit einer flachen Platte mit Schnellverschluss (Pos.2) und den Bug mit einer halbkugelförmigen Plexiglasskuppel (Pos.3) ab. Auf dem

flachen Platte mit Schnellverschluss (Pos.2) und den Bug mit einer halbkugelförmigen Plexiglasskuppel (Pos.3) ab. Auf dem

Zylinderumfang sind gleichmäßig vier Multifunktionsleisten verteilt. Sie sind längs der Rotationsachse des Zylinders angebracht und ermöglichen jeweils an fünf verschiedenen Steckpositionen die Adaption von Funktionsmodulen (Pos.6). Da die Steckplätze die elektrische Versorgung (elektrische Leistung/Steuerungssignale) der adaptierten Module über Strom führende Kontakte bereitstellt, müssen die nicht besetzten Steckplätze mit Elektroverschlüssen (Blindverschlüsse) (Pos.5) vor Verunreinigungen, Beschädigungen (mechanische/korrosive) und das Eindringen von Wasser geschützt werden. Funktionsmodule wie z. B. die Propellerantriebe werden an die Multifunktionsleiste mittels eines vereinheitlichten Schnellverschlusses gegen Abzug und Torsion gesichert und mit elektrischer Energie sowie Steuerdaten versorgt (Abb. 9). Um einen

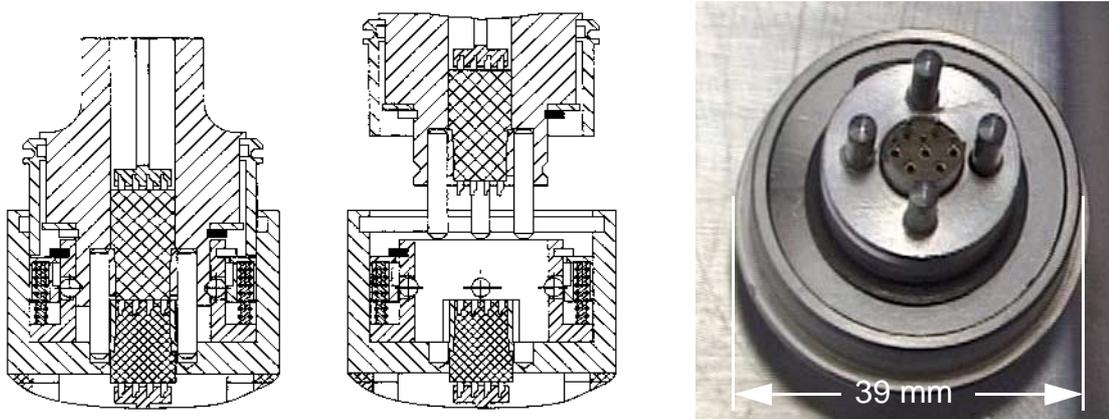


Abb. 9: Steckprinzip des Multifunktionsstecker Modul angesteckt (l), Modul abgezogen (m.)  
Steckverbindung eines Propellerantriebs (r.)

sicheren elektrischen Kontakt zwischen Modul und Trägersystem herzustellen, werden sie mit einem wasserdichten stromführenden Stecker verbunden. Jeder Steckplatz wird mit einer Kodierung versehen (siehe Kap. 3.4.6), die von dem angesteckten Modul automatisch erkannt und an die steuernde Einheit weitergeleitet wird. Es kommt dafür ein 10-poliger Steckverbinder zum Einsatz.

Die Multifunktionsleisten sind mit dem Rumpfgehäuse von innen verschraubt und damit für Wartungs- und Montagezwecke vom Basissystem trennbar. Durch diese Gestaltung werden äußere Senklöcher und Schraubenspalte vermieden, in denen sich kontaminierte Partikel oder Flüssigkeiten ansammeln könnten. Die Funktionsleiste besteht aus

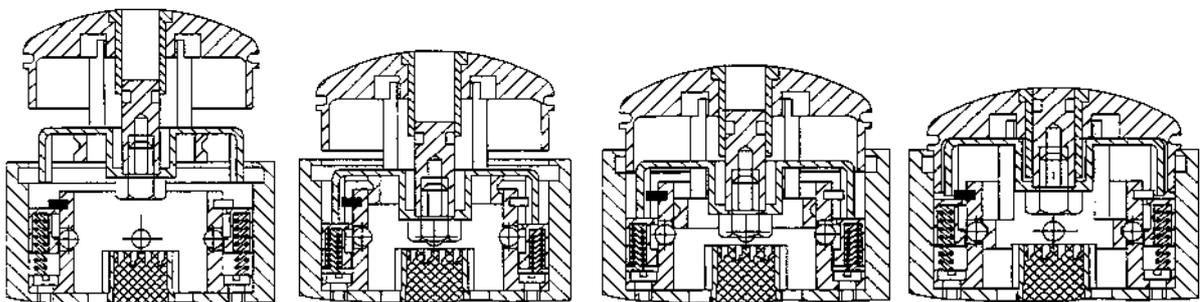


Abb. 10 : Steckprinzip der Elektroverschlüsse (Blindverschlüsse)  
Reihenfolge beim Einsetzen: v. l. n. r.; Reihenfolge beim Abziehen: v. r. n. l.

zwei Teilen, wobei Gummidichtungen zwischen dem Rumpf und dem ersten sowie dem ersten und dem zweiten Teil der Leiste die Wasserdichtigkeit gewährleisten. Die Leisten enthalten eine Kugelschlussmechanik, in der die Funktionsmodule torsions- und

abzugsicher eingeführt werden können. Um die Strom führenden Kontakte der Stecker vor einer mechanischen und korrosiven Beschädigung bzw. vor dem Kontakt mit Wasser zu schützen, werden abzugsichere und wasserdichte Elektroverschlüsse für die nicht besetzten Steckplätze eingesetzt (Abb. 10). Die Schutzkappe und der Stecker lassen sich durch einfache Translationsbewegung in die Steckeraufnahme drücken. Bei der Schutzkappe stellt sich über die Federrückstellkraft die Abzugssicherung selbsttätig ein. Die Feder- und Reibkräfte des Entsicherungssystems sind so ausgelegt, dass eine selbständige Sicherung stets gegeben ist, aber eine selbsttätige Entsicherung auch bei einer Tauchtiefe von 40 m vermieden wird.



Abb. 11: Rumpf mit Multifunktionsleisten

Für Inspektions-, Handhabungs- und Beobachtungsaufgaben ist eine Sichtverbindung zwischen Leitstand und Operationsfeld erforderlich, da durch diese bei manueller Steuerung eine genaue Positionierung des Unterwasserfahrzeugs und damit auch der Werkzeuge bzw. der Manipulatoren ermöglicht wird. Am Bug wird daher eine halbkugelförmige Plexiglas-Kuppel angeflanscht, in der ein Kamerasystem auf zwei unabhängigen, aufeinander senkrechtstehenden Rotationsachsen alle Punkte im Bereich des Sichtfeldes der Halbkugel visuell erfassen kann (Abb. 11). Die Kuppel ist mit einem hohen Sicherheitsfaktor ausgelegt und erlaubt auch versehentliche Stöße mit Hindernissen sicher abzufangen.

Im Rumpf werden zusätzlich Schienen montiert, die zur Befestigung von weiteren internen Komponenten dienen. Die Schienen sind mittig im Fahrzeug angebracht, wobei die internen Komponenten eingehängt werden. Sie haben dabei einige Zentimeter Freiraum zum Boden, so dass bei einem eventuellen Wassereintritt ein Eintauchen in das Wasser verzögert wird. Wie bereits erläutert, werden verschiedene Komponenten für das Unterwasserfahrzeug entwickelt. An dieser Stelle sei nachfolgend nur auf die entsprechenden Unterkapitel verwiesen, in denen die Komponenten mit ihrer implementierten Funktionalität, ggf. auch deren mechanischen Auslegung, ausführlich erläutert werden.

Interne Komponenten:

- ein Stereokamerasystem (Kap. 3.5.1),
- ein Lage- und Tiefensensor (Kap. 3.5.5)
- Teile der Energieversorgung (Kap. 3.4.5)

Externe Komponenten:

- die Propellerantriebe (Kap. 3.5.2),
- die Scheinwerfer (Kap. 3.5.1),
- eine Unterwasserführungsmaschine (Kap. 3.5.4),
- Manipulatoren und Greifer (Kap. 3.5.6-Kap. 3.5.3)

### 3.4 Steuerungstechnik und Steuerungsarchitektur

In den nachfolgenden Kapiteln wird die realisierte Steuerungstechnik für das Handhabungssystem TRIBUN aufgezeigt und detailliert auf die Auswahl eines Bussystems, den Aufbau der Modulstruktur, der Energieversorgung sowie der implementierten Kommunikation eingegangen.

Seit 1970 haben sich programmierbare und Rechner geführte Steuerungen in dem weiten Anwendungsfeld der Steuerungstechnik durchgesetzt. Dabei kommen im industriellen Einsatz in weiten Bereich sogenannte SPS (Speicher Programmierbare Steuerungen) zum Einsatz. Die Anbindung der Schalter, Aktoren und Sensoren erfolgt in der konventionellen Steuerungstechnik durch eine direkte Verdrahtung der Komponenten mit den Ein- und Ausgängen der SPS. Die erforderliche Verkabelung stellt dabei bei räumlich weit getrennten Baugruppen auf Grund des hohen Verdrahtungsaufwands nicht nur einen erheblichen Kostenfaktor dar, sondern ist außerdem störanfällig und wird schnell unübersichtlich. Diese Probleme werden durch den Einsatz von prozessnahen Kommunikationsnetzwerken, sogenannten Feldbussystemen, deutlich reduziert. Auf der Zellen- und Steuerungsebene können Feldbusse intelligente, dezentral angeordnete Steuerungen vernetzen oder an übergeordnete Rechner anbinden. Dabei erlaubt die Aufteilung der Steuerungsfunktion auf verschiedene steuernde Einheiten eine Vereinfachung der Softwareerstellung und des Tests. Auf Sensor- und Aktorebene dient das Feldbussystem zur Anbindung der dezentral angeordneten Komponenten /45/

Im Bereich der Automatisierungstechnik finden verschiedene SPS-Konzepte wie die Hardware-SPS, der Prozessrechner als Software-SPS, die Slot-SPS (SPS als PC-Einsteckkarte) und der PC als Software-SPS in Abhängigkeit der Einsatzkriterien Verwendung. Wenn sicherheitsrelevante Aspekte berücksichtigt werden müssen oder sogar eine redundante Ausführung erforderlich ist, kann nur eine entsprechende Hardware-SPS verwendet werden. Bei hohen Anforderungen an die Echtzeit-Reaktivität kommen Hardware- bzw. Slot-SPS oder ein Prozessrechner zum Einsatz. Sind Echtzeitanforderungen von untergeordneter Bedeutung, kann der Einsatz einer PC-Lösung dann sinnvoll sein, wenn eine umfangreiche Weiterverarbeitung der Prozessdaten in Standard-Windows-Applikationen erfolgt, da hierbei ein einfacher Datenaustausch ermöglicht wird /46/.

Eine der wesentlichen Stärken einer SPS ist die Möglichkeit, Ablaufprogramme zu erstellen und damit häufig wiederkehrende Tätigkeiten zu automatisieren. Gerade bei einer sich häufig ändernden Konfiguration des Systems und bei einer hohen Variation der Einsatzszenarien ist diese Voraussetzung nicht gegeben. Auch auf Grund fehlender Information (z. B. Positionssystemen) ist das System nicht immer oder jederzeit voll automatisierbar und muss manuell oder teilautomatisch betrieben werden.

Im Rahmen dieser Arbeit soll daher statt einer SPS eine PC-basierte Steuerung mit Feldbusanbindung zum Einsatz kommen. Die PC-Steuerungssoftware ist ein Teil dieser Arbeit und dient dabei sowohl Steuerungszwecken als auch der Visualisierung der Systemdaten. Eine solche Lösung ist deutlich flexibler als eine SPS und weist einfache Erweiterungsfähigkeit z. B. durch PC-Einsteckkarten auf. Weiterhin ist mittels des PCs auch eine Netzwerkanbindung realisierbar. „Harte“ Echtzeit ist auf PC-Ebene nicht zwingend erforderlich, da echtzeit-relevante Vorgänge auf Mikrocontroller ausgelagert werden können (siehe auch Kap. 3.4.4 und Kap. 3.7.6).

### 3.4.1 Anforderungen an ein Bussystem für TRIBUN

Ausgehend von der verwendeten Zweirechner-Steuerung und der in Kap. 2.3 beschriebenen Problematik soll von der Zweirechner-Lösung, wie sie bei den Vorgängerfahrzeugen FAUST III und  $\mu$ -FAUST eingesetzt wird, abgerückt und der Einsatz einer dezentralen Steuerung realisiert werden. Die Kommunikation der einzelnen Komponenten des Handhabungssystems wird über den Einsatz eines Feldbussystems verwirklicht /47/,/48/.

Zusammengefasst lassen sich folgende Charakteristika und Anforderungen an Feldbussysteme spezifizieren /47/:

- Effektive Verkabelungstechnik, Reduktion des Verdrahtungsaufwands
- Erhöhung der Zuverlässigkeit des Gesamtsystems
- Reduzierung der Wartungsaufgaben
- Verbesserte und effizientere Diagnosefähigkeit des Systems
- Erhöhte Flexibilität bei der Planung des Gesamtsystems
- Einfache Parametrierung der Netzwerkteilnehmer von jeder beliebigen Stelle des Netzwerks unabhängig vom Ort des Teilnehmers
- Bereitstellung einer ausreichend hohen Bandbreite für die Kommunikation
- Bereitstellung von Echtzeitfähigkeit gemäß der Anforderungen

Nachfolgend sollen zunächst die speziellen Anforderungen an ein Feldbussystem aufgestellt werden, die für das System TRIBUN notwendigerweise erfüllt werden müssen. Diese sind:

#### **Anzahl und Art der Busteilnehmer:**

Aus Gründen der Flexibilität ist es vorteilhaft, wenn mehrere steuernde Einheiten existieren können, der Bus also Multimaster-Fähigkeit aufweist. So können z. B. verschiedene Steuerungen für Roboterarme und Greifer, für die Werkzeuge, für die Fahrzeugbewegung (Propellerantriebe) oder auch für das Kamerasystem zum Einsatz kommen. Es müssen dabei typischerweise 10-20 intelligente Komponenten miteinander verknüpft werden. Bei einer dezentralen Steuerung entfällt damit auch der Aufwand, Änderungen an einer zentralen Steuerungssoftware für jeden neuen Teilnehmer durchführen zu müssen. Das Anbinden neuer Teilnehmer kann dadurch mit deutlich geringerem Aufwand realisiert werden. Der Bus sollte Ereignis orientiert ausgelegt sein, damit z. B. schnell auf Fehlermeldungen reagiert werden kann. Dabei ist es notwendig, dass alle Teilnehmer in gleicher Weise aktiv auf den Bus zugreifen können.

#### **Ausdehnung und Struktur:**

Die räumliche Ausdehnung des Systems wird im Wesentlichen durch die Länge der Kabelverbindung zwischen Leitstand und Fahrzeug sowie durch die zu vernetzten Geräte im Arbeitsraum (z. B. Beleuchtung, Positionssysteme) bestimmt. Daher soll das Bussystem eine Vernetzung über mehrere hundert Meter gestatten.

Als physikalische Struktur ist ein Ring denkbar. Dies ist aber mit höheren Verdrahtungsaufwand verbunden. Insbesondere müssen bei einer Ringstruktur Blindstecker vorgesehen werden, falls kein Modul am Steckplatz angeschlossen ist, um den

Kommunikationsfluss zu gewährleisten. Vorteilhaft ist es daher, wenn als Topologie ein Bus gewählt werden kann. Die Vernetzung soll generell über ein Kabel erfolgen, da Lichtwellenleiter zu aufwendig in der Installation und der Verbindungstechnik sind.

#### **Übertragungs- und Störsicherheit:**

Besondere Anforderungen werden an die Übertragungs- und Störsicherheit gestellt. Wichtig ist dabei, dass der Ausfall einer Komponente nach Möglichkeit nicht das Versagen des gesamten Systems impliziert. Störungen wie elektrische oder magnetische Felder, wie sie in technischen Anlagen vorkommen, müssen in gewissen Umfang toleriert werden.

#### **Kosten und Verfügbarkeit:**

Wünschenswert ist dabei auch ein hoher Verbreitungsgrad am Markt, um auf Kauflösungen zurückgreifen zu können und damit kosten- und zeitintensive Eigenentwicklungen zu vermeiden. Kauflösungen verfügen allerdings nicht immer über die gewünschte Funktionalität und bei kleinbauenden Systemen häufig auch nicht über geforderten Abmessungen. Für die Entwicklung von eigenen Hardware-Lösungen müssen entsprechende ICs für die Busanbindung zur Verfügung stehen.

#### **Geschwindigkeit, Datenraten:**

Notwendigerweise muss die Datenrate auf dem Bus eine ausreichende Bandbreite für alle Teilnehmer aufweisen. Daher sollte der Bus eine möglichst hohe Übertragungsrate aufweisen, um der Erweiterungsfähigkeit des Systems gerecht zu werden.

Für die manuelle Steuerung ist erfahrungsgemäß eine Aktualisierung neuer Stellwerte (z. B. für die Propellerantriebe) im Bereich 5 bis 10 Hz ausreichend, um den Bediener eine gute Handhabung zu garantieren.

#### **Dienste:**

Die Bereitstellung von Diensten ist für diesen Anwendungsfall nicht zwingend erforderlich. Vorteilhaft wirkt sich generell ein Standard oder eine Norm aus.

### **3.4.2 Etablierte Feldbussysteme im Überblick**

Am Markt sind momentan Bussysteme verschiedener Hersteller und Vereinigungen verfügbar, die zum Teil auch genormt sind. Um hier einige gebräuchliche zu nennen seien hier PROFI-Bus, CAN-Bus, DeviceNet, Safetybus, P-Net, FIP, InterbusS, LON oder auch der DIN-Messbus erwähnt. Grundsätzlich soll hier eine klare Trennung zwischen „echten“ Feldbussen und Aufsätzen durchgeführt werden. So stellen z. B. DeviceNet oder SafetyBus kein eigenes Bussystem dar, sondern setzen auf dem CAN-Bus auf.

Unter Berücksichtigung der oben gestellten Anforderungen werden nachfolgend die wichtigsten technischen Eigenschaften /47/ und die Einsatztauglichkeit der Bussysteme für TRIBUN untersucht. Das Bussystem soll das Steuerungssystem dabei in seiner Flexibilität möglichst minimal einschränken. Für den hier vorliegenden Einsatzfall mit mehreren Steuerungen bzw. steuernden Einheiten soll der Bus weiterhin Multimaster-Fähigkeit aufweisen.

**Reine Master-Slave-Busse:**

Unter diesen Voraussetzungen scheiden damit die reinen Master-Slave-Systeme wie **ASI** (Actuator/Sensor-Interface) oder auch der **Bitbus** (Fa. Intel) aus. Der **DIN-Messbus** ist ebenfalls ein serieller Master-Slave-Bus, der ursprünglich für die Datenübertragung im Bereich Mess- und Prüftechnik entwickelt wurde, wobei mittlerweile das Einsatzgebiet auch auf den Sensor-/Aktorbereich ausgedehnt wurde. Der **Interbus-S** ist ein firmenspezifisches Kommunikationsprotokoll der Firma Phoenix Contact GmbH & Co. (Deutschland) und zeichnet sich durch einen hohen Verbreitungsgrad aus. Interbus-S basiert allerdings auch auf einer Single-Master/Multiple-Slave-Architektur, wobei alle Komponenten in einer Ringtopologie angeordnet sind.

**Multimasterfähig aber Master-Slave-orientierte-Busse:**

**LON** (Local Operating Network), eine Entwicklung der Fa. Echelon (USA), wird bisher hauptsächlich in der Gebäudeautomatisierung verwendet. Die Kommunikation ist allerdings auch Master-Slave-orientiert und damit ebenfalls ungeeignet.

Die grundlegende Kommunikationsprozedur von **FIP** (Factory Instrumentation Protocol, Frankreich) basiert auf einer Single-Master/Multiple-Slave-Konfiguration mit zyklischem Abfragen durch den Master. Nachdem das Netzwerk initialisiert und gestartet ist, fragt der Master alle Slaves der Reihe nach gemäß einer vorher festgelegten Abfrageliste ab. Ein Slave, der durch den Master ausgewählt ist, kann dann seine Daten an mehrere andere Slaves, die an dieser Information interessiert sind, gleichzeitig verschicken. Über das zyklische Abfragen hinaus gibt es auch eine ereignisgetriebene sowie eine spezielle Punkt-zu-Punkt Kommunikationsprozedur, die durch die Slaves bei dem Master beantragt und zugewiesen werden kann. Alle Teilnehmer werden in der Startphase dem Master statisch bekannt gemacht. Dementsprechend tritt ein Problem auf, wenn ein Teilnehmer zusätzlich zum System dazugefügt wird oder entfernt wird, während das Netzwerk in Betrieb ist und ist daher für Plug&Play-Betrieb ebenfalls ungeeignet.

**P-NET** (Process Network) ist ein Protokoll, das für universelle Prozesssteuerungsanwendungen gedacht ist. P-NET ist grundsätzlich ein Master-Slave orientiertes System. Der Master fragt die Slaves zyklisch ab. Nachteilhaft ist hierbei, dass eine ereignisgetriebene Kommunikation ausgeschlossen ist. In einem physikalischen Netzwerk sind jedoch mehrere Master-Slave-Gruppen gleichzeitig möglich. Jede Gruppe repräsentiert quasi einen logisch separaten Kommunikationskanal. Der Zugriff dieser multiplen Master auf das gemeinsame Medium wird durch ein Token-Passing-Protokoll geregelt. Es sind 125 Knoten pro Netzwerk erlaubt, wobei maximal 32 Master zugelassen sind. Jeder Master kann bis zu 32 Slaves steuern. Die Bitrate ist konstant auf 76,8 kBit/s festgelegt.

Die Entwicklung des **Profibus** wurde durch die deutsche Regierung als gemeinsames Projekt zwischen Universitäten und Firmen gefördert. Profibus FMS (Process Fieldbus Message Specification) wird als universelle Lösung für allg. Feldbusapplikationen angesehen, während Profibus DP als Standard für Low Cost Sensor/Aktuator-Netzwerke eingesetzt wird. Der Profibus hält einen Satz von Diensten bereit, wie sie zum Ansprechen von intelligenten Steuerungen sowie Sensoren und Aktoren benötigt werden. Diese Dienste umfassen die Einrichtung und Verwaltung von Verbindungen zwischen Partnern und den Austausch von Objekten. Da der Profibus die Kommunikation in heterogenen Systemen zulässt, müssen alle Knoten, bevor sie die eigentliche Kommunikation beginnen, ihre Eigenschaften in der Startphase des Netzwerks zunächst mitteilen. Für

einen Plug&Play-Betrieb, bei dem im Betrieb Einheiten hinzugefügt oder entfernt werden können, ist eine Reinitialisierung des Netzwerkes ebenfalls nachteilig.

Es werden mehrere Master zugelassen, wobei eine Token Passing Technik verwendet wird, um die Zugriffsrechte auf das Kommunikationsmedium zu regeln. Der jeweils aktuelle Master darf das gesamte System für eine bestimmte Zeitspanne beherrschen. Die Slaves können nur antworten und quittieren. Wegen der grundsätzlichen Single-Master-Ausrichtung verfügt der Profibus nur über begrenzte ereignisbasierende Kommunikationsmöglichkeiten. Werden zusätzlich softwareimplementierte Protokollversionen verwendet, wird die zusätzlich Echtzeitfähigkeit begrenzt.

### **Multimaster-Busse:**

Der Multi-Master-fähige **eBUS** wurde durch die Fa. Dungs (Deutschland) insbesondere für Heizungsapplikationen entwickelt. Die Anwendung zielt auf die Kommunikation zwischen Steuergeräten ab, z. B. Heizungssteuerungen, Regler für verschiedene Heizkreise, Sensoren und Aktoren wie Mischerantriebe, Pumpen, Gebläse, Temperaturfühler, Sollwertgeber oder Anzeigen. Der eBus gestattet maximal 25 Master und 228 Slaves und wäre für die erwartete Teilnehmerzahl durchaus ausreichend, hat aber den Nachteil, dass die Kommunikation auf eine sehr geringe Baudrate von 2,4 kBit/s festgelegt ist und die Buslänge wegen der Begrenzung des Spannungsabfalls auf Standardleitungen typisch mit 100 m spezifiziert wird, was für diesen Einsatzfall je nach Konfiguration des Systems zu kurz sein kann.

Das **EIB**-Protokoll (European Installation Bus) wurde von der Fa. Siemens (Deutschland) für die Gebäudeautomatisierung entwickelt. Diverse Firmen bieten zum Beispiel Produkte wie Schalter, Steckdosen, Lampen, Rollläden, Motoren etc. mit dem Protokoll an. Das EIB-Protokoll basiert auf der CSMA/CA-Medium-Zugriffstechnik. Damit kann jeder Knoten nach erfolgreicher Arbitrierung Master sein. EIB unterstützt hierarchische Netzwerkstrukturen, die aus Teilnehmern, Unternetzen und Bereichen - Domains - ein ganzes System ergeben. Ein gesamtes Netzwerk darf daher bis zu maximal 11520 Teilnehmer haben. Hauptnachteil ist die sehr geringe Baudrate von 9600 Baud.

Der **CAN-Bus** (Controller Area Network) wurde für den Automobilbereich entwickelt und ist ebenfalls als Multi-Master-Bus ausgelegt. Er wurde anfangs nur für den Einsatz in Kraftfahrzeugen konzipiert, um den Verdrahtungsaufwand der ständig steigenden Zahl an elektronischen Komponenten zu reduzieren, hat sich aber im Laufe der Zeit auch in der Industrie (Sensor-/Aktor-Bereich) zu einem konkurrenzfähigen Feldbussystem entwickelt. Die gesamte Datenübertragung kann über eine serielle Busleitung bewerkstelligt werden, so dass zu den elektrischen Komponenten nur noch Versorgungs-, aber keine Signalleitungen mehr verlegt werden müssen.

Der CAN-Bus erlaubt in der CAN-Spezifikation 2.0A 2<sup>11</sup> in der Version 2.0B 2<sup>29</sup> Kommunikationspartner bzw. priore Botschaften, wobei ca. 30-60 Knoten, in Abhängigkeit der verwendeten Leistungstreiber, physikalisch an einem Bus angeschlossen werden können. Durch den Einsatz von Repeatern kann die Anzahl weiter erhöht werden.

Weitere grundsätzliche Leistungseigenschaften sind:

- Multimaster Buszugriffstechnik,
- Kanalzugriffstechnik CSMA/CD + AMP (Carrier Sense / Multiple Access with Collision Detection + Arbitration on Message Priority),

- kurze Botschaftslängen mit entsprechend kurzen Übertragungsverzögerungen,
- variable Bruttoübertragungsraten im Bereich von 10 kBit/s bis zu 1 MBit/s,
- maximale Buslänge bei 1 MBit/s begrenzt auf 40 m (In-Bit-Response-Protokoll),
- einfache Anbindung an den steuernden Mikrorechner,
- selbstsynchronisierende Bitcodierung und
- hohe Störsicherheit bei äußeren und inneren Fehlern.

Durch die Bewährung unter rauen Umgebungsbedingungen in Automobilen (Temperaturen von  $-25^{\circ}\text{C}$  bis  $+70^{\circ}\text{C}$ , Schüttelbelastungen bis zum hundertfachen der Erdbeschleunigung, starke elektromagnetische Störfelder im Motorenbereich) und den immer wieder geforderten geringsten Abmessungen bietet sich der CAN-Bus auch für den Einsatz in der Industrie an. Hinzu kommt der äußerst günstige Knotenpreis und die hohe Verfügbarkeit der Bauteile, die durch die Massenfertigung für die Automobilindustrie erreicht werden können. Durch die weite Verbreitung von CAN-Bus basierten Systemen steht dem Entwickler mittlerweile ein breites Spektrum an leistungsfähigen und preisgünstigen Entwicklungswerkzeugen zur Verfügung.

Eine Vernetzung und Steuerung der Fahrzeugmodule über das **Ethernet** wird im Wesentlichen für diesen Einsatzfall nicht in Erwägung gezogen, da unter dem Aspekt der Echtzeit und der Vernetzung Probleme zu erwarten sind. So sind bei der Verwendung einer sternförmigen Busstruktur mehrere Verteiler (Hubs) für die Netzwerk-Kommunikation im Rumpf des Fahrzeugs notwendig. Der Aufwand für die Verkabelung und der Platzbedarf der Hubs ist dabei relativ hoch. Bei einer Verdrahtung über BNC-Steckverbindungen können gerade im Bereich der Multifunktionsleisten leicht Schwierigkeiten mit der Terminierung auftreten. Ein weiterer Nachteil ist, dass zu Beginn dieser Arbeit am Markt keine kostengünstigen und kleinbauenden Module zur Verfügung stehen.

Auf Grund der Eigenschaften und unter Berücksichtigung der oben genannten Anforderungen wird der CAN-Bus ausgewählt. Er weist eine hohe Störunanfälligkeit und die höchste Fehlererkennung mit einer Hamming-Distanz von 6 auf. Nachteilig ist, dass er nur eingeschränkt echtzeitfähig ist, da höher priorisierte Teilnehmer durch ständiges Senden nieder priorisierte Teilnehmer blockieren können. Es muss daher erhöhte Aufmerksamkeit auf die Programmierung der Kommunikation gelegt werden.

### 3.4.3 Modulelektronik und Modulaufbau

Die Modulelektronik lässt sich in drei logisch abgegrenzte Bereiche unterteilen wie in Abb. 12 zu sehen ist. Zum einem besteht sie aus einer Mikrocontroller-Einheit mit CAN-Busanbindung, die die notwendigen Funktionen für die Kommunikation, die Ansteuerung, die Regelung und die Überwachung des Moduls übernimmt. Der zweite Teil besteht aus der Sensor/Aktor-spezifischen Elektronik, wie z. B. einer Leistungselektronik zur Anbindung von Gleichstrommotoren oder der Sensorelektronik. Beide Gruppen werden über den Energieversorgungsteil mit den benötigten Spannungen versorgt.

Die Energieversorgung der einzelnen Module soll grundsätzlich über eine zentrale Versorgungseinheit im Rumpf des Fahrzeugs erfolgen. Dabei ist es aus praktischen und funktionalen Gründen sinnvoll, alle Module einheitlich an die Energieversorgung und das Bussystem anzubinden. Da eine Vielzahl von unterschiedlichen Modulen aus dem Bereich der Antriebstechnik und Sensorik zum Einsatz kommen können, ist es nur

schwer abzuschätzen, welche Spannungen mit welcher Leistung in Abhängigkeit der aktuellen Konfiguration bereitgestellt werden müssen. Eine Bereitstellung von mehreren gebräuchlichen Spannungen führt zwangsläufig zu einer schlechten Auslastung, da in der Regel nicht alle vorhandenen Spannungen auch erforderlich sind. Weiterhin muss ein erhöhter Verdrahtungsaufwand in Kauf genommen werden, was sich gerade innerhalb der Multifunktionsleisten und des Steckersystems nachteilig auswirkt.

Das hier entwickelte Konzept für die elektrische Anbindung sieht daher vor, dass lediglich zwei Leitungen mit 24 V Gleichspannung der Zuführung der elektrischen Energie sowie zwei weitere Leitungen zur Busanbindung dienen. Alle benötigten Spannungen müssen in den einzelnen Modulen über geeignete Spannungswandlung (DC-DC-Wandler) erzeugt werden. Nachteilig ist hierbei natürlich der erhöhte finanzielle Aufwand für die Spannungswandler sowie der zusätzlich benötigte Bauraum.

Die einheitliche Versorgungsspannung von 24 V wurde gewählt, da sie in der Automatisierungs- und Steuerungstechnik eine hohen Verbreitungsgrad aufweist und daher eine große Zahl Komponenten direkt, ohne weitere Spannungswandler, angebunden werden können. Vorteilhaft ist ebenfalls, dass an allen Steckverbindungen nur ungefährliche Niederspannung anliegt.

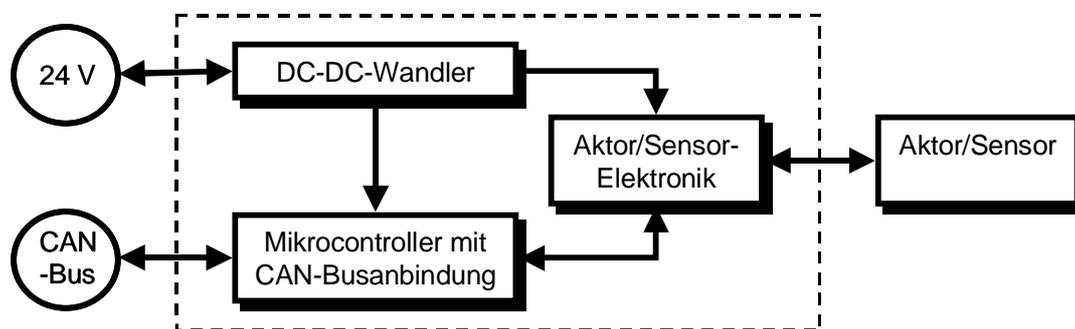


Abb. 12: Aufbau der Modulelektronik

Der Markt bietet Wandlertypen mit verschiedenen Leistungsmerkmalen an, so dass sowohl symmetrische als auch asymmetrische Versorgungen in Ausführungen als Aufwärts- und Abwärtssteller bereitstehen, die ausreichend kleinbauend sind, einen hohen Wirkungsgrad aufweisen und damit geringe Abwärme erzeugen.

Durch die An- und Abkoppelbarkeit der Module an den Feldbus und die zentrale Energieversorgung auch während des Betriebes (Hot Plug&Play) wird vor allem eine kurze Rüstzeit und eine hohe Wartungsfreundlichkeit garantiert.

Bei den Modulen kommt eine Sandwich-Bauweise zum Einsatz, wobei eine Aufnehmerplatte in der Regel die modulspezifische Energieversorgung und Aktorik bzw. Sensorik enthält. Die Mikrocontroller-Einheit wird steckbar über eine entsprechende Stiftleistenverbindung integriert. Dieser Aufbau hat den Vorteil, dass defekte Baugruppen sehr schnell ausgetauscht werden können und auf Kabelverbindungen weitestgehend verzichtet werden kann.

#### 3.4.4 Auswahl der Mikrocontroller-Familie

Grundsätzlich besteht nicht die Notwendigkeit, sich an einen bestimmten Prozessor bzw. eine Prozessorfamilie zu binden. Als einzige Bedingung müssen die Mikrocontrol-

ler entweder über ein On-Chip-CAN-Controller verfügen oder aber über die bereitgestellten I/O-Ports eine Anbindung an einen externen CAN-Controller gestatten.

Aus wirtschaftlichen und funktionalen Gründen ist es aber meistens nicht sinnvoll, unterschiedliche Prozessoren einzusetzen, da zur Programmentwicklung spezielle Werkzeuge (z. B. Compiler) ebenfalls angeschafft werden müssen. Dabei sind die Entwicklungswerkzeuge typischerweise auf bestimmte Controller-Familien zugeschnitten. Neben den in der Regel hohen Anschaffungskosten steigt daher mit jedem neuen Werkzeug natürlich auch die Einarbeitungszeit und der Aufwand für die Wartung oder die Modifikation der erstellten Software.

Ausgewählt wurden Derivate der HC11-Familie der Fa. Motorola, da damit bereits in mehreren Projekten gute Erfahrungen gesammelt wurden und die entsprechenden Entwicklungswerkzeuge, wie Compiler und HW-Entwicklungssystem, bereitstehen. Die Rechenleistung der Controller ist für die zu erwartenden Aufgaben (Kommunikation, Implementierung einfacher PID-Regler) ausreichend. Die Controller verfügen standardmäßig über mehrere digitale I/O Ports, einen 8-Kanal 8-Bit AD-Wandler, verschiedene programmierbare Timer, eine serielle RS 232 Schnittstelle sowie frei programmierbare Interrupts.

Eine Marktrecherche ergab, dass es für diese Controller auch vollständige Module mit CAN-Bus-Anbindung gibt, so dass auf reine Eigenentwicklungen verzichtet werden kann. Zum Einsatz kommt ein sog. Scheckkartenrechner der Fa. SIE (Platine hat etwa die Größe einer Scheckkarte). Als entscheidendes Auswahlkriterium verfügt sein implementiertes Betriebssystem über die Möglichkeit, die erstellte Software zu Entwicklungszwecken direkt über den CAN-Bus in den Scheckkartenrechner zu laden und auszuführen. Damit entfällt in der Entwicklungsphase die Notwendigkeit, für jede neue Softwareversion ein EPROM zu erstellen. Nach erfolgreichen Funktionstest kann dann das Nutzerprogramm und das Betriebssystem in ein EPROM gebrannt werden. Die Programmierung der Controller erfolgt in der Programmiersprache C /49/.

Nachfolgend werden die wichtigsten technischen Eigenschaften kurz zusammengefasst.



Abb. 13: Scheckkarten-Rechner mit CAN-Busanbindung

Der Scheckkartenrechner mc68 stellt eine auf dem Mikrocontroller Motorola 68HC11 (Prozessor A8) basierende Anbindung an den CAN-Bus dar /50/. Auf der I/O-Seite verfügt er über analoge Eingänge, fest konfigurierte TTL-Ein- und Ausgänge sowie über eine konfigurierbare User-Hardware-Schnittstelle. Eine Digital/Analog-Wandlung kann über ein PWM-Signal (Software-PWM) ausgegeben werden, eine Möglichkeit zur A/D-Konvertierung ist ebenfalls

vorgesehen. Daten- und Adressbus sind über Steckerleisten ausgeführt und bieten die Möglichkeit vieler Erweiterungen.

Ein BasicCAN-Controller 82C200 von Philips übernimmt mittels einer ISO-DIS 11898 - Schnittstelle die Verbindung mit dem Netzwerk. Er deckt alle möglichen CAN-Baudraten ab. Zum Schutz des Einplatinenrechners und zur Verhinderung von EMV-Problemen kann der CAN-Bus zusätzlich noch optisch entkoppelt werden. Zusätzlich ist auch ein Watchdog-Timer vorhanden, der bei sicherheitsrelevanten Anwendungen eine Überwachung des ordnungsgemäßen Betriebes des Rechners ermöglicht.

Der Rechner wird mit einem herstellerspezifischen Betriebssystem auf EPROM ausgeliefert. Zur Unterstützung möglichst vielfältiger Anwendungsfälle können Benutzerprogramme mit unterschiedlichen Aufgabenbereichen getrennt definiert und aufgerufen werden. Die Programme können dabei auf ein breites Spektrum von Befehlen des Betriebssystems zurückgreifen, wie Instruktionen zur Behandlung analoger und digitaler Ein- und Ausgabe, Datenausgabe auf dem CAN-Bus und Speicherzugriffe (siehe auch Kap. 3.4.8). Der Scheckkartenrechner stellt somit ein eigenes Kleinrechnersystem dar, das auch Echtzeitanforderungen genügt und durch seine umfangreiche Funktionalität für anspruchsvolle Aufgaben im Steuer- und Regelungsbereich eingesetzt werden kann. Das Mikrocontrollermodul kann daher auch alle Funktionen zur Überwachung und Regelung der modulspezifischen Hardware übernehmen, so dass die ansteuernde Einheit nur Stellbefehle (Sollgrößen) vorgibt, aber nicht aktiv in Regelvorgänge eingreifen muss.

Ein weiterer zum Einsatz kommender Platinentyp ist eine Eigenentwicklung auf 68HC11K4-Basis der Fa. Motorola (Abb. 14). Der Entwurf ist in Hardware, Software und Kommunikation an die SIE-Scheckkarte angelehnt.

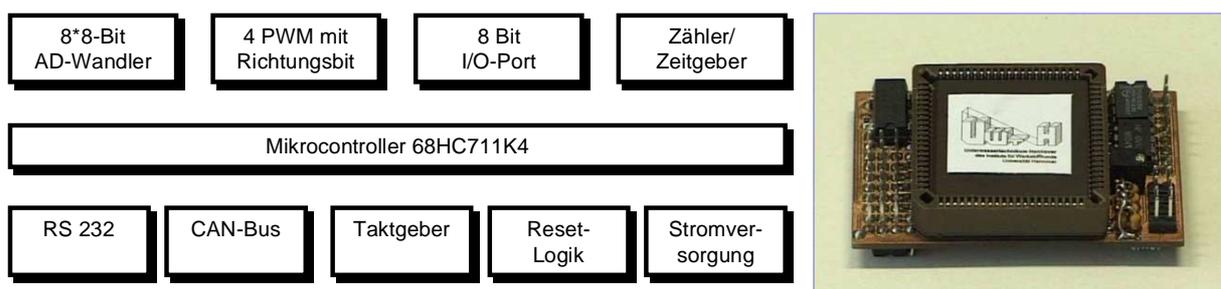


Abb. 14: Hardware des Einplatinenrechners (l.) und aufgebautes Modul (r.)

Die Platine hat jedoch mit 38\*60 mm kleinere Abmessungen und verfügt hardwaremäßig zusätzlich über vier Hardware-PWM-Ausgänge sowie Interrupt-gesteuerte Zähler-eingänge, so dass sie sich speziell zur Ansteuerung und Regelung von Gleichstrommotoren eignet.

### 3.4.5 Energieversorgung und Versorgungskabel

ROVs werden über eine Kabelverbindung vom Leitstand mit der benötigten elektrischen Energie versorgt. Um die Handhabung des Fahrzeugs nicht zu beeinträchtigen, soll der Kabeldurchmesser möglichst gering sein, da sonst der Strömungswiderstand aber auch die Masse des Kabels die Steuerung des Bootes unzulässig hoch beeinträchtigen würde.

Tabelle 2: Abschätzung des max. Strombedarfs bei 24 V Betriebsspannung

Bezeichnung	Stromaufnahme [A]	Anzahl	Gesamtstrom [A]
Antriebsmotor Propeller	3,5	8	28
Scheinwerfer	5	2	10
Kameraservos	1	3	3
Kameraelektronik	0.5	1	0.5
Lagesensor	0.5	1	0.5
Schrittmotor Führungsmaschine	3	2	6
Pumpe Führungsmaschine	2	1	1
restliche Elektronik	1	1	1
<b>Gesamt</b>			<b>50</b>

Die maximale Leistungsaufnahme im ROV TRIBUN beträgt 1,2 kW bei einer Bordspannung von 24 V und einem Strom von 50 A (Tabelle 2). Da die Versorgungsspannung im ROV für die verwendeten Module konstant gehalten werden muss, ist es erforderlich, die Versorgungsspannung unabhängig vom Lastfall und damit auch von den auftretenden Kabelverlusten zu gewährleisten. Es sind nun zwei Konzepte denkbar, um die sich ergebenden Kabelverluste zu kompensieren.

Wird die Spannung im ROV vom Leitstand aus geregelt, ist es notwendig, die lastabhängigen Kabelverluste durch schaltungstechnische Maßnahme zu kompensieren. Eine solches Versorgungsprinzip wurde bei dem Vorgängerfahrzeug FAUST III verwendet. Der Vorteil ist, dass im Fahrzeug weniger Einbauraum benötigt wird sowie weniger Masse bewegt werden muss. Nachteilig wirken sich jedoch die hohen Kabelverluste gerade bei Vollast aus, die leicht ein mehrfaches der benötigten Leistung im ROV betragen können.

Praxisgerechter ist es, das ROV mit einer höheren Versorgungsspannung bei geringem Strom zu versorgen und die benötigte Betriebsspannung über DC-DC-Wandler im Boot selbst zu erzeugen. Auf dem Markt sind kleinbauende Geräte verfügbar, die mit einer Eingangsspannung von ca. 200-400 V Gleichspannung arbeiten. Im Leitstand erfolgt die Gleichrichtung und Glättung der Netzspannung (230 V~) (Abb. 15). Die Nachteile, die durch erhöhtes Gewicht, höheren Platzbedarf und Sicherheitsaufwand auftreten, werden durch leichteres Kabel und geringere Verlustleistung und damit verbesserte Handhabbarkeit des ROVs kompensiert.

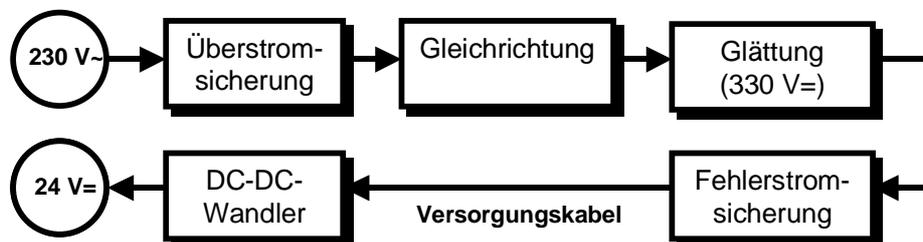


Abb. 15: Prinzip der Spannungsversorgung im ROV

Das Versorgungskabel ist als Mehrdrahtleitung ausgelegt, wobei ein dreidrahtiges Kabel für die Energieversorgung, ein zweidrahtiges geschirmtes Kabel für den CAN-Bus und zwei Koaxialkabel für die Videoübertragung verwendet werden. Alle Kabel sind in einem

Schlauchpaket eingezogen, wobei Masse und Auftrieb des Kabels so aufeinander abgestimmt sind, dass das Kabel im Wasser eine Restauftriebskraft von ca. 0,04 N/m aufweist und somit das Fahrzeug in seiner Bewegung nur minimal beeinflusst.

### 3.4.6 Plug&Play-Konzept mit Steckplatzkennung

Zur Identifizierung der Steckplätze der einzelnen Module des ROVs wird eine Vereinbarung gemäß Abb. 16 getroffen. Die Nummer eines Steckplatzes ergibt sich aus der Kombination von Leistennummer und Position auf dieser Leiste. Die Leisten sind oben beginnend von eins bis vier im Gegenuhrzeigersinn durchnummeriert (Abb. 16). Die Steckplätze auf einer Leiste werden von Heck aus zum Bug gezählt.

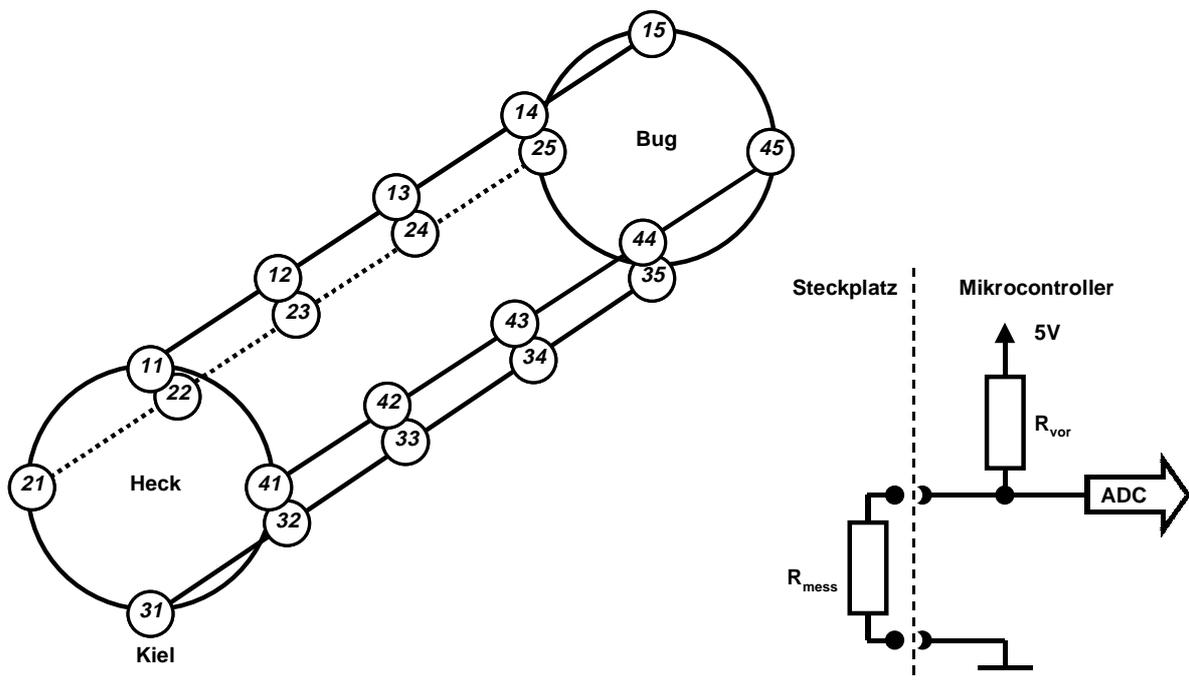


Abb. 16: Steckplatznummerierung TRIBUN (l.) und Messschaltung Steckplatzerkennung (r.)

Die angesteckten Module erkennen den Steckplatz automatisch. Um den Aufwand für Kodierung- und Dekodierung sowohl im Steckplatz als auch im angesteckten Modul gering zu halten, wird eine analoge Kennung gewählt. Realisiert wird diese durch eine Widerstandsmessung, wobei jedem Steckplatz ein bestimmter Widerstand zugeordnet wird, der über zwei Kontakte im Stecker zur Messung zugänglich ist. Dadurch ergibt sich eine einfache Messschaltung nach Abb. 16. Bei 5 V Messspannung soll der Gesamtwiderstand  $R_{ges}=5\text{ k}\Omega$  betragen. Die eingesetzten Mikrokontrollertypen besitzen einen internen 8 Bit Analog-Digital-Wandler (ADC). Damit ergeben sich  $N=2^8 - 1=255$  mögliche Messwerte und bei 20 Steckplätzen ein rechnerischer Wert von 12,8 Messwerten/Steckplatz. Reduziert man jeden Bereich auf 10 Messwerte pro Steckplatz, kann man die 20 Bereiche symmetrisch auf den gesamten Messbereich verteilen. Weiterhin stehen damit noch fünf Steckplatzkennungen bereit, die für weitere Steckplätze genutzt werden können. Somit liegt der größte vom ADC gewandelte Wert bei  $R_{dezimal,max}=220$ . Über die Spannungsteilerregel ergibt sich der entsprechende Widerstandswert.

$$R_{220} = \frac{R_{dezimal,max}}{N} R_{ges} = 4317\text{ k}\Omega \quad (Gl. 1)$$

Dadurch wird auch der Messvorwiderstand festgelegt auf

$$R_{\text{vor}} = R_{\text{ges}} - R_{220} = 687\Omega \quad (\text{Gl. 2})$$

Damit berechnen sich die Widerstände  $R_{nm}$  für jeden Steckplatz auf der Leiste  $n$  mit der Position  $m$  auf dieser Leiste über

$$R_{nm} = \frac{R_{\text{vor}}}{\left(\frac{N}{R_{\text{dezimalnm}}} - 1\right)} \quad (\text{Gl. 3})$$

Wobei  $R_{\text{dezimalnm}}$  den Wert bezeichnet, den der ADC als Ergebnis einer Analog-Digital-Wandlung liefert angibt. Er berechnet sich zu

$$R_{\text{dezimalnm}} = 20 + 10m + 50(n - 1) \quad (\text{Gl. 4})$$

Widerstände werden industriell üblicherweise in Normreihen, den sog. E-Reihen, gefertigt. Zur eindeutigen Kodierung der Steckplätze sind Standard-Filmwiderstände der E24-Reihe mit einer Toleranzklasse von 5% ausreichend.

### 3.4.7 Spezifische Motormodulkennwerte

Jeder Antrieb übt eine gerichtete Kraft auf das ROV aus. Damit nicht jede Motormodulbauart eine eigene Modulkenning erhalten muss, werden Motormodule mit einer einheitlichen Kennung ausgestattet. Als zusätzliche Parameter übertragen die Module die Richtung dieser Kraft sowie deren Angriffspunkt am Grundkörper des Bootes an die Steuerung. Daraus kann diese die Wirkung der Motorkraft des jeweiligen Antriebsmo-

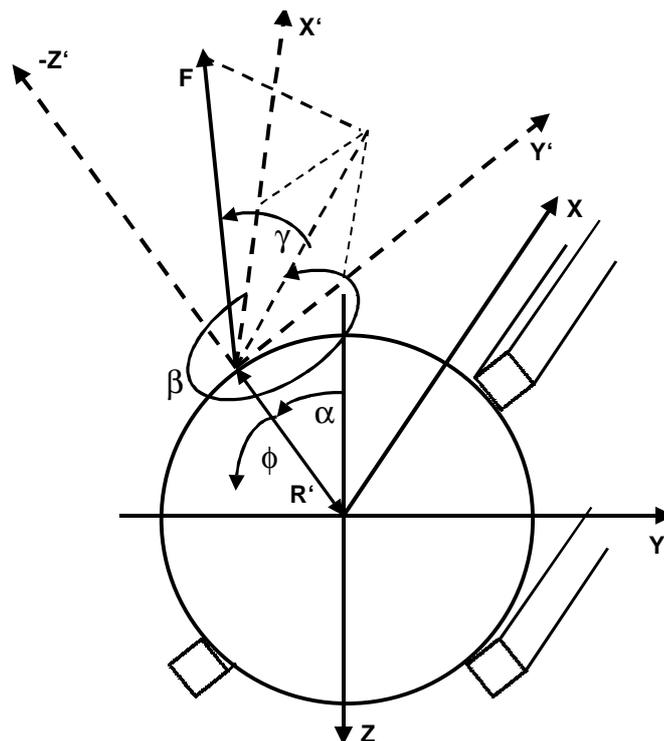


Abb. 17: Koordinatensysteme der Steckplätze

duls auf TRIBUN berechnen. Standardmäßig wird davon ausgegangen, dass die Drehzahl der einzelnen Motoren vorgegeben werden kann.

Um größtmögliche Flexibilität und Erweiterbarkeit des Gesamtsystems zu erhalten, wird eine allgemeine Vereinbarung getroffen.

Die Kraft  $F$  eines Motors wird in ihrer Richtung relativ zum Steckplatz mit den zwei Winkeln  $\gamma$  und  $\beta$  angegeben (Abb. 17).  $\gamma$  ist der Höhenwinkel, also die Wirkung in Richtung der Z'-Achse, und  $\beta$  der Drehwinkel, also die Lage der Kraft in der X'-Y'-Ebene. Um alle möglichen Positionen angeben zu können, darf  $\gamma$  Werte von Null bis 90 Grad und  $\beta$  von Null bis 360 Grad annehmen. TRIBUN kann zusätzlich noch um einen Winkel  $\alpha$  aus der Nulllage (Leiste 1 oben) gedreht werden. Technisch sinnvolle Lagen sind allerdings nur  $\alpha = 0^\circ$  und  $\alpha = 45^\circ$ . Aus diesen Angaben kann die Steuerung nun die Wirkung des Antriebes auf das Boot berechnen.

Zur Vereinfachung der Berechnungen kann die maximale Motorkraft  $F$  auf Eins normiert, da nur Antriebe mit gleicher Leistung verfügbar sind und im Wesentlichen die Krafrichtung für eine Ansteuerung von Bedeutung ist. Zunächst wird die Gesamtkraft  $F$  in die Komponenten  $F_x$ ,  $F_y$  und  $F_z$  im Koordinatensystem des Steckplatzes aufgeteilt.

$$F_{z'} = \sin(\gamma) \quad (Gl. 5)$$

$$F_{h'} = \cos(\gamma) \quad (Gl. 6)$$

$$F_{y'} = -F_{h'} \sin(\beta) = -\cos(\gamma) \sin(\beta) \quad (Gl. 7)$$

$$F_{x'} = F_{h'} \cos(\beta) = \cos(\gamma) \cos(\beta) \quad (Gl. 8)$$

Daraus lassen sich dann mit dem Lagewinkel  $\phi$  (gezählt von der negativen Z-Achse) der Leiste des Steckplatzes relativ zur Z-Achse die Kraftkomponenten im Koordinatensystem des Bootes berechnen.

$$F_x = F_{x'} = \cos(\gamma) \cos(\beta) \quad (Gl. 9)$$

$$F_y = F_{y'} \cos(\phi) + F_{z'} \cos(90^\circ - \phi) = -\cos(\gamma) \sin(\beta) \cos(\phi) + \sin(\gamma) \sin(\phi) \quad (Gl. 10)$$

$$F_z = -F_{y'} \sin(\phi) + F_{z'} \sin(90^\circ - \phi) = \cos(\gamma) \sin(\beta) \sin(\phi) + \sin(\gamma) \cos(\phi) \quad (Gl. 11)$$

Aus diesen Komponenten ergeben sich auch die resultierenden Momente um die Hauptachsen X, Y und Z. Die Momente  $M_x$ ,  $M_y$ ,  $M_z$  berechnen sich aus den Abständen der Komponenten zu den Drehachsen (Abb. 18). Es wird dabei vereinfachend davon ausgegangen, dass das Fahrzeug um die Fahrzeugmitte dreht (Achsen  $D_x$ ,  $D_y$ ,  $D_z$ ). Die Kraft  $F$  eines Motors greift dabei mit dem Abstand  $R'$  senkrecht zur Fahrzeuglängsachse an.  $W$  sei dabei der Abstand zwischen zwei Steckplätzen einer Leiste.

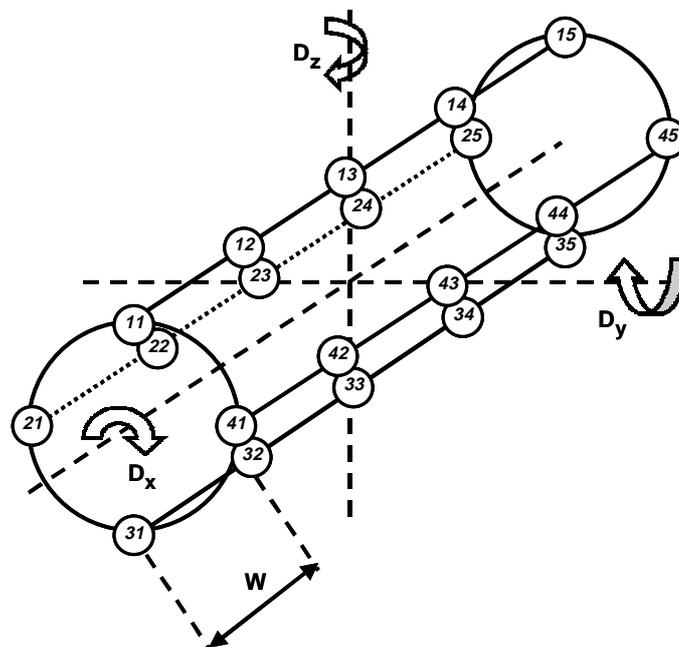


Abb. 18: Drehachsen und Kraftabstände

Somit ergeben sich die einzelnen Momente zu:

$$M_x = F_y R' \cos(360 - \phi) + F_z R' \sin(360 - \phi) = F_y R' \cos(\phi) - F_z \sin(\phi) \quad (\text{Gl. 12})$$

$$M_y = -F_x R' \cos(360^\circ - \phi) + F_z n W = -F_x R' \cos(\phi) + F_z n W \quad (\text{Gl. 13})$$

$$M_z = -F_x R' \sin(360 - \phi) - F_y n W = F_x R' \sin(\phi) - F_y n W \quad (\text{Gl. 14})$$

Bei den gebräuchlichen Bauformen der Module, also Verwendung von rechten Winkeln, ergeben sich noch einige weitere Vereinfachungen (Abb. 19). Diese sind zur leichteren Unterscheidbarkeit nachfolgend benannt und ihre spezifischen Parameter angegeben (Tabelle 3).

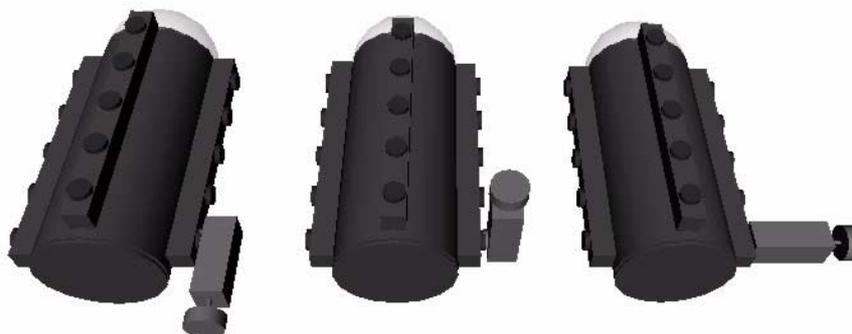


Abb. 19: Verschiedene Motortypen; Längsmotor (l.); Quermotor (m.); Kopfmotor (r.)

Tabelle 3: Gewichtete Kraftanteile für verschiedene Modultypen

Bezeichnung	$\beta$	$\gamma$	x	y	z	$M_x$	$M_y$	$M_z$
Kopfmotor	$0^\circ$	$90^\circ$	0	$\sin \phi$	$\cos \phi$	0	$nW \cos \phi$	$-nW \sin \phi$
Quermotor	$270^\circ$	$0^\circ$	0	$\cos \phi$	$-\sin \phi$	$R'$	$-nW \sin \phi$	$-nW \cos \phi$
Längsmotor	$0^\circ$	$0^\circ$	1	0	0	0	$-R' \cos \phi$	$R' \sin \phi$

Für einen Plug&Play-Betrieb ist es erforderlich, dass das entsprechende Modul seine charakteristischen Daten  $\beta$  und  $\gamma$  und die Steckplatzkennung an die Steuerung überträgt.

### 3.4.8 CAN-Kommunikationsprotokoll

Für die Implementierung eines Geräteprofils bietet sich das CANopen-Profil der CiA an / 51/,/52/,/53/. Nachteilig dabei ist jedoch der sehr hohe Programmieraufwand, um alle von CAL unterstützten Protokolle und die dabei auftretenden Zustände zu implementieren. Hinzu kommt, dass es sich bei diesem Handhabungsgerät generell nicht um eine stationär installierte Anlage handelt, sondern dass das System an wechselnden Einsatzorten und mit verschiedenen steuernden Einheiten betrieben werden kann. Das wiederum schließt fast gleichzeitig ein Benutzen des CANopen-Profiles aus, da die vier Services des CAL auf einer Master-Slave-Struktur basieren, wobei der Master in der Regel jedoch als PC ausgeführt, um die zahlreichen Konfigurationsmöglichkeiten nutzen zu können.

Verwendung findet das auf dem 68HC11-Scheckkartenrechner implementierte Betriebssystem mit dem firmenspezifischen IBS-Protokoll. Das Protokoll unterscheidet Nachrichten, die zwischen Knoten ausgetauscht werden, und Nachrichten, die von einem Knoten an alle Teilnehmer gesendet werden (Broadcast-Nachrichten). Dabei werden diese Gruppen nochmals in quittierte und nicht quittierte Nachrichten eingeteilt. Um die gezielte Kommunikation zwischen zwei Knoten zu gewährleisten, muss jedem Knoten eine eindeutige Adresse (1-240) zugeordnet werden. Der Bereich über 240 ist reserviert. Der Adresse 0 kommt eine spezielle Bedeutung zu, mit ihr werden alle Knoten gleichzeitig adressiert (Broadcast-Nachrichten).

Das Protokoll erlaubt, Datenpakete mit einer Länge von 0 bis zu 5 Bytes zu senden. Komplexe Datentypen werden von dem Protokoll nicht spezifiziert. Drei Bytes in jedem Datenpaket dienen zur Benennung von Empfänger, Absender und Befehlscode der Botschaft. Da das IBS-Protokoll keine Aussagen darüber macht, wie komplexe Datentypen oder längere Botschaften als 5 Bytes zu senden sind, muss das Datenpaket von der Applikation des Senders selbst in mehrere kleine CAN-Botschaften segmentiert und vom Empfänger in der richtigen Reihenfolge wieder desegmentiert werden. Die CAN-Nachricht eines Kommandos setzt sich in einer Bytefolge gemäß Tabelle 4 zusammen.

Tabelle 4: Datenrahmen des IBS-Protokolls

ID0	ID1	Ziel	Quelle	Befehl	Data 0	Data 1	Data 2	Data 3	Data 4
-----	-----	------	--------	--------	--------	--------	--------	--------	--------

Das erste Byte enthält immer den Empfänger des Kommandos, das zweite den Absender. Das dritte Byte spezifiziert gleichzeitig das Kommando und die Kommandoklasse. Dafür wurde der Kommandobereich von 0 bis 255 eingeteilt. Besondere Bedeutung kommt bei CAN dem Identifier jeder CAN-Nachricht zu. IBS sieht vor, den Identifier nach der Formel Identifier = 0x333+Knotenadresse zu berechnen. Damit wird jedem Knoten, nicht wie CAN es ursprünglich vorsieht, jedem Ereignis im System eine bestimmte Priorität zugeordnet. Die Identifier sind so gewählt, dass es möglich ist, IBS in ein CAL-Netzwerk zu integrieren.

Tabelle 5: Einteilung der Kommandos im IBS-Protokoll

Kommando-bereich	Befehlsklasse
0x00 – 0x2F	Befehle für allgemeine Funktionen, wie „RAM lesen“ und „RAM schreiben“
0x30 – 0x37	Broadcast-Nachrichten, die nicht quittiert werden
0x38 – 0x3F	Broadcast-Nachrichten, die eine Antwort der angesprochenen Knoten erfordern
0x41 – 0x63	frei verwendbarer Bereich
0x64 – 0x93	für Antworten auf Kommandos aus dem Bereich 0x00 - 0x2F
0x9C – 0xA3	für Antworten auf Kommandos aus dem Bereich 0x38 - 0x3F
0xA5 – 0xC7	für Antworten auf Kommandos aus dem Bereich 0x41 - 0x63
0xC8	positive Quittung
0xC9	negative Quittung

Die Befehlseinteilung zur Kommunikation gemäß Tabelle 5 bietet für die Erstellung von anwenderspezifischen Applikationen eine ausreichende Anzahl Befehle.

### 3.4.9 CAN-Kommunikationsverhalten der Module

Auf Grund der vorliegenden Erfahrung mit diesem Protokoll wird es als Kommunikationsgrundlage für die Scheckkartenrechner gewählt und ebenfalls für die HC11 K4-Variante implementiert. Aufbauend auf dem Protokoll wird ein Geräteprofil festgelegt, um möglichst viele Vorteile des CANopen zu übernehmen.

Im Wesentlichen basiert das innerhalb dieser Arbeit entworfene Profil darauf, in einem definierten Zeitrahmen (spätestens alle 2 s) eine Broadcast-Statusmeldung abzusetzen, die Knotenfunktion, Knotenstatus und Knotenadresse enthält. Diese Statusmeldung besitzt im ganzen CAN-System die niedrigste Priorität, um vorrangige Steuermeldungen nicht zu verzögern. Die Kanalkapazität wird durch diese zyklischen Statusmeldungen nur relativ gering belastet. Dem 16-Bit-Funktionscode in der Statusmeldung ist eine eindeutige Funktions-ID zugeordnet und dient den anderen Knoten zur Identifizierung.

Auf der Quittierung der einzelnen Kommandos und eine definierte Reaktion der Knoten bei Ausfall eines Kommunikationspartners liegt besonderer Wert, um dem Einsatz der Handhabungsfahrzeuge in sicherheitsrelevanten Umgebungen gerecht zu werden. Auf eine Überwachung der einzelnen Knoten nacheinander mittels Master-Slave-Prinzip

wird verzichtet, alle Knoten haben innerhalb eines bestimmten Zeitraums eine Statusmeldung abzusetzen. Geschieht das nicht, so muss das restliche System davon ausgehen, dass der Knoten nicht mehr an der Kommunikation teilnehmen kann.

Ein weiterer Schwerpunkt wird darauf gelegt, dass das Netzwerk sehr einfach zu konfigurieren und zu warten ist. Soll z. B. den Kommandos eines Knotens im Nachhinein eine höhere Priorität zugewiesen werden, so genügt es, die Knotenadresse zu verringern. Durch die zyklisch gesendete Statusmeldung dieses Knotens können dann alle auf Daten von ihm angewiesenen Knoten seine neue Adresse registrieren und die Kommunikation wieder gezielt mit ihm aufnehmen. Damit wird ein aufwendiges manuelles Rekonfigurieren aller abhängigen Knoten auf eine veränderte Adresse überflüssig. Die Knotenprioritäten können auf diese Weise auch im laufenden Betrieb geändert werden, wodurch ein kompletter Neustart entfallen kann.

Tritt im Knoten ein Fehler auf, wird zum Zeitpunkt des Auftretens und für die Dauer des Fehlers zyklisch mit der Statusmeldung eine Fehlermeldung abgesetzt. Die anderen Knoten haben autonom zu entscheiden, ob sie von diesem Fehler betroffen sind oder nicht und geeignete Maßnahmen einzuleiten.

Von jedem Modul können über die zyklisch gesendete Statusmeldung hinaus noch weitere Informationen abgefragt werden. Diese Informationen erleichtern im späteren Betrieb die Konfiguration oder helfen Inkompatibilitäten auf Grund der Verwendung von Modulen mit alten Softwareversionen aufzudecken. Alle Knoten verfügen über mindestens zwei Zustände, die über den CAN-Bus angesteuert werden können. Dabei handelt es sich um die Zustände *standby* und *operational* (Abb. 20). Ein Modul wird sich nach dem Reset automatisch initialisieren und dann in den Zustand *operational* gehen. Auf Anforderung von einem anderen Netzwerkteilnehmer kann ein Modul gestoppt, also in den Zustand *standby* versetzt werden. In diesem Zustand ist eine Änderung der Betriebsparameter möglich. Geht das Modul dann wieder in den Zustand *operational* über, so wird es zuerst neu initialisiert und nimmt dann wieder an der Kommunikation teil. Diese zwei Zustände erleichtern besonders in der Testphase das Arbeiten, um gezielt Betriebsparameter zu verändern.

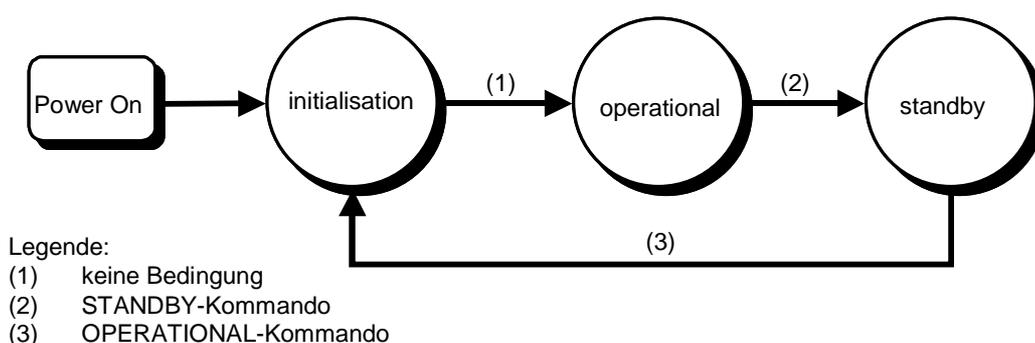


Abb. 20: Modulbetriebszustände

Im Zustand *operational* reagiert das Modul auf seinen kompletten Befehlssatz. Im Zustand *standby* hingegen werden nur die für die 68HC11-Scheckkarten gültigen Betriebssystem-Befehle akzeptiert und als einziger weiterer Befehl das OPERATIONAL-Kommando, um wieder in den Zustand *operational* zu gelangen /54/. Sollten bei zukünftigen komplexeren Knoten noch mehr Zustände benötigt werden, so können diese problemlos hinzugefügt werden.

## 3.5 Komponenten des Unterwasserfahrzeugs TRIBUN

Der grundlegende mechanische Aufbau wurde bereits in Kap. 3.3 vorgestellt. In den nachfolgenden Abschnitten wird nun auf die internen und externen Komponenten und Einbauten näher eingegangen.

### 3.5.1 Kamerasystem

Ein wesentlicher Bestandteil jedes ROVs ist die integrierte Videokamera. Mit ihrer Hilfe navigiert der Bediener das Fahrzeug, da er in der Regel keinen direkten Sichtkontakt zum Fahrzeug hat. Daher kommt der Kamera und deren Steuerung besondere Bedeutung zu.

Das Kameramodul umfasst zwei Kameras, die die gleiche Szene aus zwei leicht unterschiedlichen Positionen und Blickrichtungen aufnehmen und damit die Grundlage für ein stereoskopisches Sehen bilden. Um die Bewegung der Kameras gezielt beeinflussen zu können, muss ein mechanischer Kameraträger mit den entsprechenden elektrischen Stellmotoren und die dafür notwendige Ansteuerelektronik bereitgestellt werden.

Einen weiteren Schwerpunkt des Kamerasystems stellt die Datenübertragung der Kamerapositionsdaten vom Bediener zum Kameramodul und der Videobilder von den Kameras zum Bediener dar. Die Videobilder können aufgrund ihrer großen Bandbreite (bis 5 MHz) nicht über den CAN-Bus übertragen werden, deshalb sind dafür getrennte Leitungen vorgesehen. Alle anderen Steuerinformationen zwischen Kamera- und Trackingmodul werden über den CAN-Bus übertragen.

Zur besseren räumlichen Orientierung ist es sinnvoll, dem Bediener einen stereoskopischen Eindruck zu vermitteln. Oft wird dies über ein sog. Head-Mounted-Display erreicht (Kap. 3.6), wobei dem Bediener zwei vor den Augen befindliche Monitore bereitgestellt werden. Die Steuerung wird idealerweise über die Erfassung der Kopfbewegung mittels eines Head-Trackers realisiert.

Die Kamera kann dabei über das PC-Steuerungsprogramm überwacht, gesteuert und konfiguriert werden. Die Verwendung des PC ist allerdings optional, d. h. Kamera- und Trackingmodul können nach abgeschlossener Konfiguration eigenständig arbeiten.

Moderne CCD-Kameras verbinden dabei ausreichendes Auflösungsvermögen des Bildes bei kompakter und leichter Bauweise. Sie führen den Weiß- und Schwarzabgleich automatisch durch und regeln elektronisch die Lichtempfindlichkeit nach. Mit einer typischen Auflösung von 499\*582 Farbbildpunkten liegt ihre Wiedergabequalität weit über der Auflösung der meisten HMDs. Für das Kamerasystem werden zwei CCD-Kameramodule (Fa. Conrad Elektronik) eingesetzt. Mit ihrer extrem geringen Baugröße (30\*30 mm) und einem Gewicht von je 20 g eignen sie sich ideal für den Einbau in der Frontkuppel des Fahrzeugs. Die Auslieferung erfolgt mit einem Weitwinkelobjektiv (48° Bilderfassungswinkel). Falls eine andere Brennweite gefordert wird, können wahlweise Linsen mit Bilderfassungswinkeln von 26° oder 150° nachgerüstet werden.

Die Kameras müssen zum Betrieb von einer 12 V Gleichspannungsquelle gespeist werden und nehmen dabei ca. 160 mA auf. Als Ausgangssignal liefern sie ein FBAS-Signal im PAL-Format und können somit direkt an einem Fernsehmonitor angeschlossen werden.

Als Antriebstechnik für den Kameraträger werden Servos verwendet. Die aus dem Modellbaubereich bekannte Einheiten bestehen aus einem Gleichstrommotor, einem Getriebe und einer integrierten Elektronik, um die Abtriebsachse auf eine vorgegebene Position drehen zu lassen. Als Eingangsgröße dient ein PWM-Signal, dessen PWM-Länge proportional zum Soll-Drehwinkel der Abtriebsachse ist. Im Kameraträger werden sie eingesetzt, um die Kameras direkt zu bewegen. Gegenüber dem Einsatz von Schrittmotoren benötigen sie keine weiteren elektronischen Komponenten. Vorteilhaft ist auch, dass sie nach dem Einschalten nicht referenziert werden müssen.

Die Servos beeinflussen das Bildergebnis signifikant: Von ihrer Stellgenauigkeit hängt im Wesentlichen der räumliche Eindruck ab. Weichen sie mehr als  $2^\circ$  von ihrer Sollposition ab, werden vom Gehirn zwei scheinbar überlagerte Bilder wahrgenommen und der stereoskopische Eindruck geht verloren /55/.

Die Stellgenauigkeit wird von zwei Faktoren beeinflusst: von der Qualität der eingebauten Elektronikkomponenten und dem mechanischen Spiel des Getriebes. Das Servogetriebe ist in der Regel in Metall- oder Kunststoffausführung erhältlich. Generell gilt, dass Metallgetriebe größere Stellmomente aufnehmen können, während Kunststoffgetriebe das geringere Spiel aufweisen. Da in dem Kamerasystem nur sehr geringe Momente zur Positionierung der Kameras aufgebracht werden müssen (die Trägheit der 20 g schweren Kamera kann vernachlässigt werden), werden Servos mit Kunststoffgetriebe ausgewählt werden.

Ein weiteres Kriterium, das die Qualität der Darstellung beeinflusst, ist die Stellgeschwindigkeit, d. h. die Zeit, die vergeht bis ein Servo von der aktuellen Position die nächste Sollposition erreicht hat. Die Winkelgeschwindigkeit eines Servomotors ist entsprechend seiner Getriebeübersetzung reziprok zum Stellmoment. Folgen die Kameras zu langsam der Sollposition, so läuft das Bild der vom Gehirn erwarteten Ansicht hinterher. Diese Verzögerungszeit ist von anderen Virtual Reality Anwendungen und professionellen Simulatoren bekannt. Ist die Zeit zu groß, kann dies leicht zu Übelkeit und Schwindelgefühlen des Benutzers führen. Bei der Auswahl der Servos wurde deshalb besonders auf eine hohe Stellgeschwindigkeit geachtet.

Um dem Bediener einen möglichst realistischen Eindruck der Kameraszene zu vermitteln, muss die Kamerabewegung der Kopfbewegung so weit wie möglich folgen können. Hierbei sind allerdings alle auf dem Markt verfügbaren Servos durch ihre technischen Daten auf einen Bereich von  $\pm 45^\circ$  Drehwinkel eingeschränkt. Versuche haben allerdings gezeigt, dass die in dieser Arbeit eingesetzten Servos problemlos in einem Stellbereich von  $\pm 60^\circ$  linear arbeiten. Die Soft- und Hardware des Kameramoduls kann zum Erzeugen des entsprechend modifizierten PWM-Signals angepasst werden.

Die Servos vom Typ Micro 3 BB der Firma Multiplex erfüllen alle oben genannten Bedingungen hinreichend, wobei eine Stellgeschwindigkeit von  $260^\circ/\text{s}$  mit einem Stellfehler von max.  $1^\circ$  erreicht wird.

Zur Bewegung der Kameras werden grundsätzlich mindestens drei Servos benötigt, wobei sich zwei verschiedene Lösungen (Abb. 21) anbieten.

In Variante a) steuert jeweils ein Servo die Bewegung der Kameras um die horizontale und die vertikale Achse. Ein dritter Servo stellt über ein zusätzliches Getriebe den Konvergenzwinkel ein, wobei gleichzeitig die Achsen 1 und 2 bewegt werden. Vorteil ist hierbei, dass der Konvergenzwinkel fest eingestellt bleibt, wenn die Kamera horizontal geschwenkt werden.

Dem gegenüber stellt Variante b) einen optimierten Entwurf bezüglich des mechanischen Aufbaus dar. Dabei wird auf den Servo zum Drehen des Kameraträgers um die vertikale Achse verzichtet und stattdessen jede Kamera direkt auf einem Servo montiert. Die beiden Servos (Achse 1, Achse 2) stellen dann für jede Kamera gleichzeitig die Blickrichtung und den Konvergenzwinkel ein. Daraus resultiert ein stark vereinfachter mechanischer Aufbau, da das zusätzliche Getriebe entfällt. Dafür fallen einige Teile der Software komplexer aus.

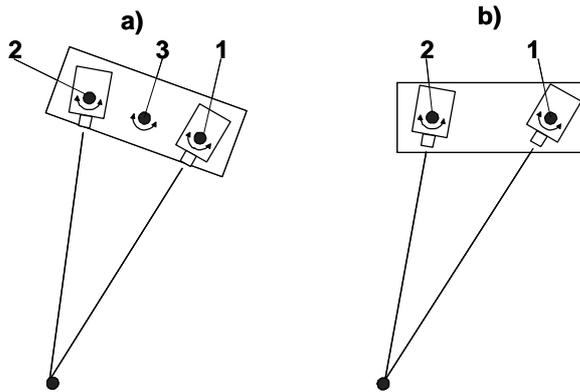


Abb. 21: Mechanischer Aufbau (horizontal)  
a) 3-Achsen  
b) 2-Achsen

Abb. 21 verdeutlicht außerdem, dass die Drehung des gesamten Kameraträgers in der Kuppel des TRIBUN mehr Freiraum beansprucht, als die vereinfachte Lösung. Die Software wurde dennoch so konzipiert, dass die tatsächliche mechanische Umsetzung zu der Schnittstelle zum CAN-Bus gekapselt ist. Der Benutzer des Kameramoduls kann den Konvergenzwinkel und die vertikale und horizontale Blickrichtung festlegen. Die Berechnung der Bewegung der einzelnen Servoachsen wird dann, abhängig vom mechanischen Aufbau, von der Software vorgenommen.

Die Bildübertragung der beiden Kameras zum Leitstand kann auf Grund der großen Bandbreite der Videosignale (5 MHz/Kamera) nicht über den zur Kommunikation vorgesehenen CAN-Bus erfolgen. Sie erfolgt daher über zwei Koaxialkabel mit einem Wellenwiderstand von 75 Ohm. Soll der daraus entstehende Verkabelungsaufwand reduziert werden, so ist es möglich, die beiden Videosignale moduliert über ein Koaxialkabel zu übertragen und vor dem Einspeisen in das HMD wieder zu demodulieren. Für die Modulation bietet sich auch die in der Fernsehtechnik eingesetzte Amplitudenmodulation an, da für dieses Verfahren schon fertige Modulatoren und Demodulatoren verfügbar sind.

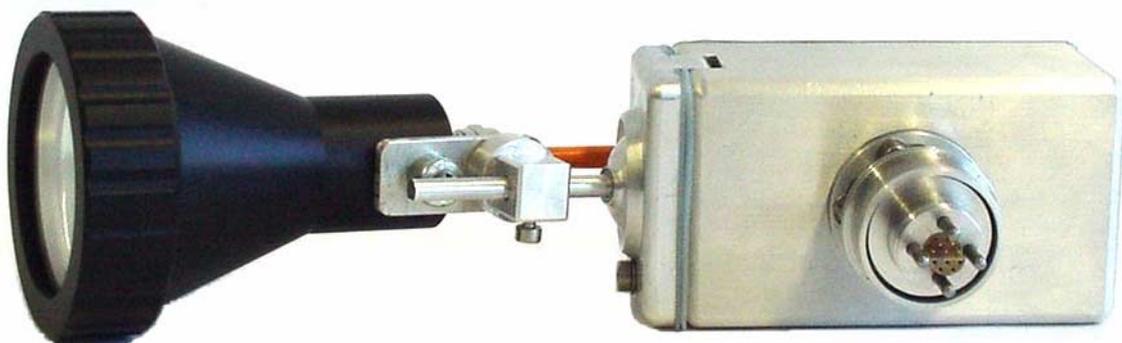


Abb. 22: Beleuchtungseinheit

Das Fahrzeug benötigt für den Einsatz unter Wasser eine Beleuchtungseinrichtung, um die Qualität der Bilder des Kamerasystems jederzeit zu gewährleisten. Die entwickelte Beleuchtungseinheit mit einer Leistung von 50 W kann den Arbeitsraum vor den Plexi-

glaskuppel vollständig ausleuchten und lässt sich über die Verstelleinrichtungen vom Bediener in vier verschiedenen Achsen mit wenigen Handgriffen frei justieren (Abb. 22).

### 3.5.2 Propellerantriebe

Für das Trägerfahrzeug wurden zunächst vier Propellerantriebsmodule aufgebaut. Diese enthalten im Gehäuse jeweils zwei Antriebsmotoren, die Leistungselektronik sowie die Mikrocontroller-basierte Steuerung mit CAN-Busanbindung. Die Antriebe sind um 90 Grad versetzt angeordnet, drehzahl geregelt und erlauben es das Fahrzeug mit definierten Schubkräften zu bewegen.

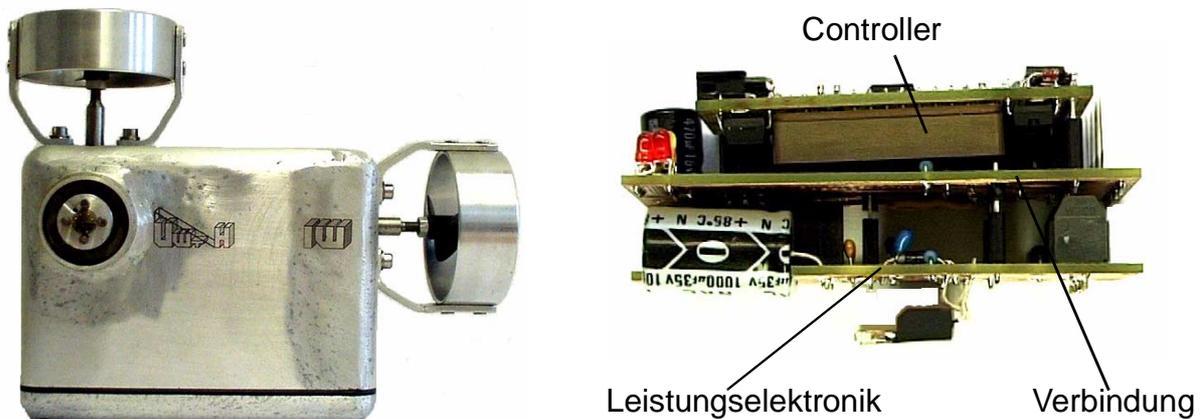


Abb. 23: Propeller-Doppelantrieb (l.) und Modulelektronik für zwei Propellerantriebe (r.)

Die Steuerelektronik in Sandwich-Bauweise (Abb. 23) verfügt über eine automatische Steckplatzerkennung und meldet die Position des Antriebsmoduls an die steuernde Einheit. Diese Information wird dann im Monitoringbetrieb zur Visualisierung der Fahrzeugkonfiguration genutzt. Weiterhin kann sie auch für die Regelung verwendet werden, da die Anzahl der steuer- bzw. regelbaren Freiheitsgrade des Fahrzeugs durch jeden vorhandenen Antrieb beeinflusst wird.

### 3.5.3 Manipulator mit Schnellkupplung

Für die Durchführung von Manipulations- sowie Bergungsaufgaben ist es erforderlich, neben der Position auch die Orientierung der Sensoren oder Werkzeuge einstellen zu können. Eine Positionierung in allen Freiheitsgraden allein über die Propellerantriebe ist relativ schwierig. Diese Problematik wird durch den Einsatz eines Manipulators vereinfacht. Er wird am Kiel des Fahrzeugs adaptiert, wodurch sich der Schwerpunkt nach unten verschiebt und es über eine stabile Vorzugslage verfügt. Das Fahrzeug dient zum Anfahren der Position und erlaubt Drehungen um die z-Achse. Es wurde daher ein dreiachsiger Manipulator in Drehgelenkarmkonfiguration entwickelt, der Drehungen um die x-Achse und die z-Achse gestattet (Abb. 24). Zur Aufnahme von verschiedenen Endeffektoren ist an der vordersten Drehachse eine Arretierungseinheit befestigt, an die ein entsprechendes Gegenstück gekuppelt werden kann (Abb. 25). Dieses Kupplungsstück wird fest mit dem zu handhabenden Gerät oder auch einem Werkzeug verbunden. Es wurde eine Kinematik mit drei Freiheitsgraden gewählt, wobei die erste Achse nur manuell vor Inbetriebnahme eingestellt werden kann. Die einzelnen Achsen verfügen über Achsgeber, die zur Positionserfassung dienen.



Abb. 24: Manipulator mit Arretierungseinheit (l.) und Arretierungseinheit mit Kupplung (r.)

Damit lassen sich auch definierte Achspositionen anfahren. Für eine automatische Positionierung werden nachfolgend die Berechnungsvorschriften für die direkte und die indirekte Kinematik hergeleitet.

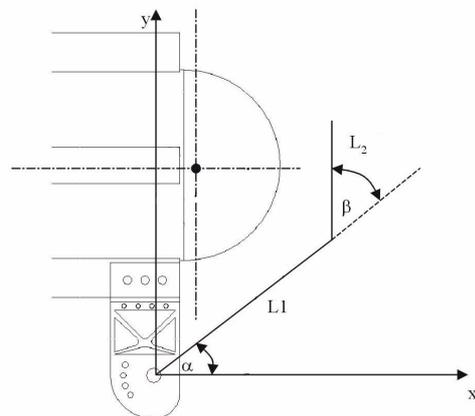
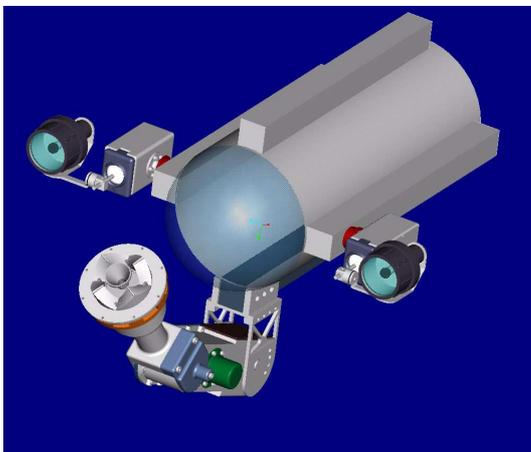


Abb. 25: TRIBUN mit Manipulator und Scheinwerfern (l.); Geometrie des Manipulators (r.)

Die direkte Kinematik hat die Aufgabe, den vorgegebenen Gelenkstellungen eine Endeffektorstellung zuzuordnen. Die Aufstellung, der das System beschreibenden Denavit-Hartenberg-Matrizen, erfolgt nach den in /58/ genannten Verfahrensschritten.

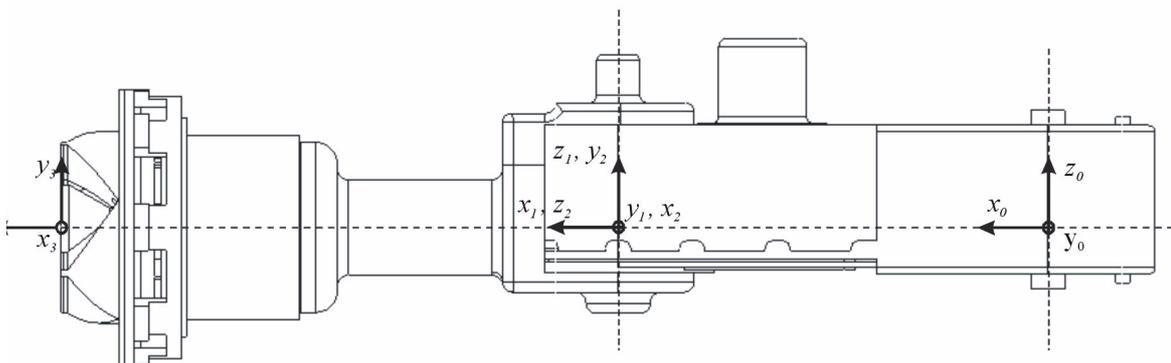


Abb. 26: Strukturmodell des Manipulators

Nach Positionierung der Koordinatensysteme in dem Strukturmodell des Manipulators (Abb. 26) werden die D.-H.-Parameter dem Modell entnommen und sind zusammengefasst in Tabelle 6 wiedergegeben.

Tabelle 6 : Denavit-Hartenberg Tabelle für 3-Achsen-Manipulator

Achse i	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	0	$L_1$	0	$q_1$
2	$\pi/2$	0	0	$q_2 + \pi/2$
3	0	0	$L_2 + L_e$	$q_3$

Mit der allgemeinen Vorschrift zum Berechnen von D.-H.-Matrizen

$$A^{i-1}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Gl. 15)$$

ergeben sich, für die aus dem D.-H.-Tabelle verwendeten Werte, die einzelnen Transformationsvorschriften der Teilkörper zu:

$$A_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & L_1c_1 \\ s_1 & c_1 & 0 & L_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2^1 = \begin{bmatrix} -s_2 & 0 & c_2 & 0 \\ c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Gl. 16)$$

mit den Abkürzungen:

$$s_1 = \sin(q_1) \quad \text{und} \quad s_2 = \sin(q_2) \\ c_1 = \cos(q_1) \quad \text{und} \quad c_2 = \cos(q_2)$$

Mittels der Matrizen ist es nun möglich, die homogene Transformation aus dem Endeffektorsystem  $(KS)_E$  in das Inertialkoordinatensystem  $(KS)_0$  zu berechnen:

$$T_E^0 = A_1^0 A_2^1 A_3^2 \quad (Gl. 17)$$

Unter Anwendung der Additionstheoreme folgt daraus:

$$T = \begin{bmatrix} -c_3s_{12} & -s_3s_{12} & c_{12} & L_2c_{12} + L_1c_1 \\ c_3s_{12} & -s_3c_{12} & s_{12} & L_2s_{12} + L_1s_1 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (Gl. 18)$$

mit den Abkürzungen

$$s_{12} = \sin(q_1 + q_2)$$

$$c_{12} = \cos(q_1 + q_2)$$

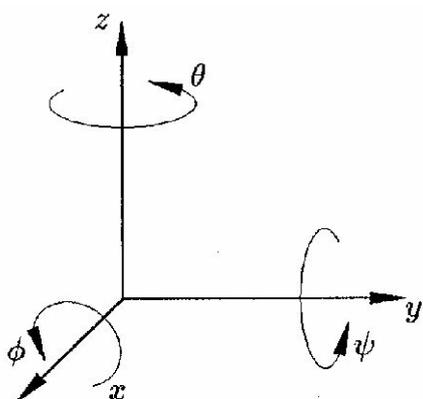


Abb. 27: Effektorsystem

Die Lage sowie auch die Orientierung des Endeffektorsystems lässt sich nun direkt aus der Matrix ablesen. Dabei gelten die in der Abb. 27 verdeutlichten Bezeichnungen. Die Lageelemente des Systems werden beschrieben durch:

$$x_0 = T_{E,(1,4)}^0$$

$$y_0 = T_{E,(2,4)}^0$$

$$z_0 = T_{E,(3,4)}^0 \quad (\text{Gl. 19})$$

Die Orientierung lässt sich berechnen durch:

$$\Psi_E = \arcsin(T_{E,(1,3)}^0) \quad \theta_E = \arctan\left(\frac{-T_{E,(1,2)}^0}{T_{E,(1,1)}^0}\right) \quad (\text{Gl. 20})$$

mit  $T_{E,(ij)}^0$  : i-tes Element der j-ten Spalte von  $T_E^0$ .

Zusammengefasst ergibt sich daraus für die direkte Kinematik die folgende Zusammenstellung:

$$x_0 = L_2 \cos(q_1 + q_2) + L_1 \cos(q_1)$$

$$y_0 = L_2 \sin(q_1 + q_2) + L_1 \sin(q_1)$$

$$z_0 = 0$$

$$\Psi_E = q_1 + q_2 + \frac{\pi}{2}$$

$$\theta_E = q_3 \quad (\text{Gl. 21})$$

Aus diesen das System beschreibenden Gleichungen lässt sich nun durch Einsetzen der aktuellen Gelenkpositionen sofort die Lage sowie auch die Orientierung des Endeffektors zum als ortsfest angenommenen Fahrzeug ermitteln.

Die inverse Kinematik berechnet die benötigten Gelenkstellungen der einzelnen Achsen, wenn eine bestimmte Stelle im Raum vom Endeffektor eingenommen werden soll. Sie ergibt sich aus:

$$q = f^{-1}(x) \quad (\text{Gl. 22})$$

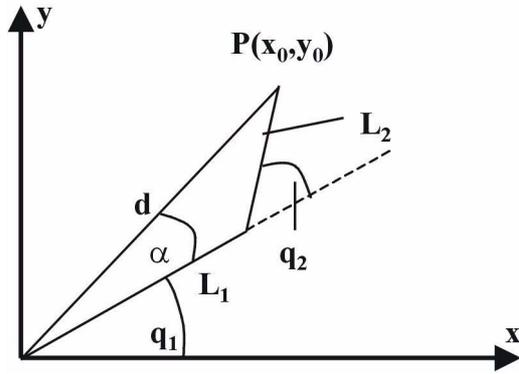


Abb. 28: Inverse Kinematik

Für die Koordinaten des Endeffektors im Arbeitsraum gilt:

$$x_0 = L_2 \cos(q_1 + q_2) + L_1 \cos(q_1)$$

$$y_0 = L_2 \sin(q_1 + q_2) + L_1 \sin(q_1)$$

(Gl. 23)

Durch Betrachtung der Abb. 28, die das System im 2-Dimensionalen Raum zeigt, muss weiterhin die Bedingung

$$\begin{aligned} d^2 &= x_0^2 + y_0^2 \\ &= [L_2 \cos(q_1 + q_2) + L_1 \cos(q_1)]^2 + [L_2 \sin(q_1 + q_2) + L_1 \sin(q_1)]^2 \end{aligned}$$

(Gl. 24)

gelten. Nach Umformung ergibt sich daraus direkt die Zustandsgleichung für das zweite Gelenk in Abhängigkeit der Endeffektorkoordinaten.

$$q_2 = \pm \arccos\left(\frac{x_0^2 + y_0^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

(Gl. 25)

Durch Einführen eines Hilfswinkels  $\alpha$  und mit der aus Abb. 28 gewonnenen Bedingung:

$$\tan(\alpha + q_1) = \frac{y_0}{x_0} \Leftrightarrow q_1 = \pm \arctan\left(\frac{y_0}{x_0}\right) - \alpha$$

(Gl. 26)

folgt daraus die Zustandsgleichung für den Winkel des ersten Gelenkes, nachdem man durch Anwendung des Sinussatzes den Hilfswinkel  $\alpha$  berechnet hat. Aus dem Sinussatz folgt:

$$\frac{\sin(\alpha)}{\sin(\Pi - q_2)} = \frac{L_2}{\sqrt{x_0^2 + y_0^2}}$$

(Gl. 27)

Substituiert man  $q_2$  durch die in (Gl. 25) gewonnene Beziehung ergibt sich:

$$\alpha = \pm \arcsin\left(\frac{\sqrt{(2L_1L_2)^2 - (x_0^2 + y_0^2 - L_1^2 - L_2^2)^2}}{4L_1^2(x_0^2 + y_0^2)}\right)$$

(Gl. 28)

Einsetzt in (Gl. 26) ergibt sich:

$$q_1 = \pm \arctan\left(\frac{y_0}{x_0}\right) \mp \arcsin\left(\frac{\sqrt{(2L_1L_2)^2 - (x_0^2 + y_0^2 - L_1^2 - L_2^2)^2}}{4L_1^2(x_0^2 + y_0^2)}\right)$$

(Gl. 29)

Für einen automatisierten Einsatz des Manipulators können die Gleichungen (Gl. 21) für die direkte sowie (Gl. 25) und (Gl. 29) für die inverse Kinematik genutzt werden.

### 3.5.4 Zweiachsige Linearführungsmaschine

Das Unterwasserfahrzeug lässt sich über die Propellerantriebe zentimetergenau positionieren, was für die Durchführung von Arbeiten, wie die optische Inspektion oder das Bergen von Kleinteilen, absolut ausreichend ist. Durch das Versorgungskabel, den Restauftrieb oder auch durch Strömungen wirken dabei permanent Störkräfte auf das Fahrzeug, die eine ständige Nachpositionierung erforderlich machen, wenn es auf einer bestimmten Position gehalten werden soll. Für die Führung von Sensoren, z. B. zur Untersuchung von Oberflächen mittels Ultraschall oder die Durchführung von Schneid-aufgaben, ist eine präzise Positionierung im Millimeter-Bereich erforderlich, die über eine Navigation mittels der Propellerantriebe nicht erreicht werden kann. Beim Vorgängerfahrzeug FAUST III wurde diese Problematik durch die fest integrierte Zweiachsen-Führungsmaschine gelöst. Um das Fahrzeug bzw. die Führungsmaschine zu fixieren, wird es mit den Propellerantrieben gegen die Oberfläche gedrückt. Dabei wird eine Anpresskraft im Bereich einiger Newton erzielt.

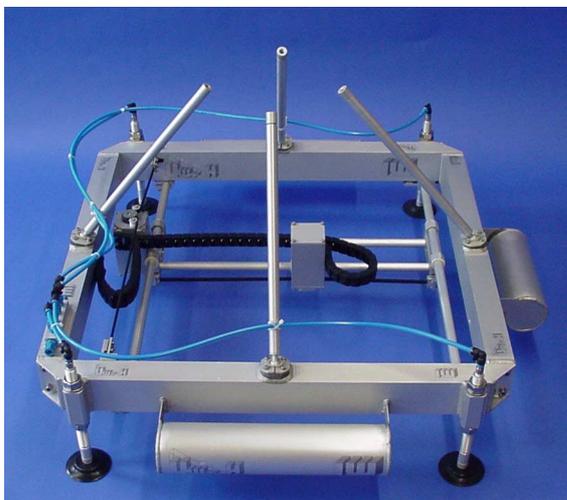


Abb. 29: Linearführungsmaschine

Um auch mit TRIBUN präzise Arbeiten durchführen zu können, wurde eine Zweiachsen-Linearführungsmaschine entwickelt (Abb. 29). Die Führungsmaschine ist als Rahmenkonstruktion ausgelegt und kann mittels vier Adaptionstangen mit dem ROV verbunden werden. Das Andrücken an eine Oberfläche kann dabei ebenfalls über die Propellerkräfte erfolgen. Zusätzlich wurde sie mit vier Vakuumsauggreifern ausgestattet, die es erlauben, die gesamte Einheit an ebenen oder leicht gewölbten Flächen zu fixieren, um auch Tätigkeiten ausführen zu können, die höhere Rückstellkräfte generieren. An einer Aufnahmevorrichtung können verschiedene Werkzeuge oder Sensoren

adaptiert werden.

Die Linearführungsmaschine verfügt über zwei Schrittmotoren, die über Zahnriemen und Umlenkvorrichtungen die Bewegung der Achsen erlauben, sowie eine Wasserpumpe zur Erzeugung des Unterdrucks in den Sauggreifern. Über ein Magnetventil können die Sauggreifer von der Unterdruckerzeugung getrennt werden. Dies bietet den Vorteil, dass die Pumpe nur in Betrieb genommen werden muss, falls der Unterdruck in den Greifern die zulässige Grenze unterschreitet. In einem wasserdichten Gehäuse werden die Steuerelektronik, die Sensorik zur Überwachung des Unterdrucks, das Ventil und die Pumpe integriert. Die Pumpe kann einen relativen Unterdruck  $p_{\text{rel}}$  von ca. 30.000 Pa erzeugen, wodurch sich eine Andruckkraft von

$$\begin{aligned} F_{A, \text{ges}} &= n \cdot A_G \cdot p_{\text{rel}} \\ &= 600\text{N} \end{aligned} \quad (\text{Gl. 30})$$

bei  $n=4$  eingesetzten baugleichen Sauggreifern mit einer wirksamen Anpressfläche von  $A_G=50\text{cm}^2$  ergibt. Werden größere Haltekräfte benötigt, so kann dies problemlos durch Austausch und Einsatz von Greifer mit größerer Fläche erreicht werden.

Die Führungsmaschine kann wahlweise manuell oder vollautomatisch betrieben werden. Über die integrierte Ablaufsteuerung der PC-Software können Auflaufprogramme im geschwindigkeitsgesteuerten Punkt-zu-Punkt-Betrieb realisiert und abgearbeitet werden.

### 3.5.5 Lage- und Tiefenerfassung

Grundlegende Voraussetzung für ein Monitoring sowie die Implementierung einer Fahrzeugführung und -regelung stellt die Positions-, Lage- und Orientierungserfassung dar. Gegenüber landgestützten Fahrzeugen und Robotern macht sich die fehlende Reibung im Unterwasserbereich sehr störend bemerkbar, da Objekte schon durch geringe Kräfte eine Positionsänderung erfahren und auf lange Zeiträume gesehen einer extremen Drift unterliegen können.

Positionssysteme lassen sich z. B. mittels Photogrammetrie, Laserstrahl (Theodoliten), Beschleunigungssensoren oder Ultraschall-Laufzeit-Verfahren realisieren [33/,/56/. Mit ihnen lassen sich alle 6 Freiheitsgrade erfassen. Alle Messverfahren haben aber den Nachteil, dass durch eine ungünstige Lage von Einbauten und Installationen, durch Verdeckung oder Reflexion in gewissen Bereichen, die Erfassung unmöglich gemacht wird.

Unterstützend könnte ein Kompass eingesetzt werden, wenn nur ein eingeschränkter Winkelbereich des Nick- und des Rollwinkels bzw. eine geringe Auslenkung aus der Nulllage (z. B. Schiffe) zu erwarten ist, also eine Vorzugslage existiert, was bei TRIBUN nicht zwingend gegeben sein muss. Gegen den Einsatz eines Kompasses gerade in technischen Anlagen spricht seine sehr hohe Fehleranfälligkeit gegen Störungen des Erdmagnetfeldes. Diese kann sowohl durch Einbauten mit permanentem Magnetfeld als auch durch das Feld stromdurchflossener elektrischer Leiter gegeben sein.

Inertialsysteme basieren auf Beschleunigungssensoren zur Erfassung aller translatorischen und rotatorischen Beschleunigungen mit nachgeschalteter Auswertung und Berechnung. Da es sich um ein relatives System handelt, bei dem aus der bekannten Position unter Verwendung der aktuellen Beschleunigungen die neuen Werte bestimmt werden, unterliegt es langfristig einer Drift, die entweder durch hochgenaue und damit teure Systeme oder eine regelmäßige Neuinitialisierung korrigiert werden muss. Hochgenaue Systeme, wie sie beispielsweise in der Luftfahrttechnik oder auch bei AUVs Verwendung finden, sind sehr teuer (> DM 30.000) so dass sie hier nicht zum Einsatz kommen.

Um dem Bediener trotzdem eine gewisse Unterstützung zukommen zu lassen, wurde innerhalb dieser Arbeit eine kostengünstige Alternative für die absolute Erfassung von einigen Freiheitsgraden entwickelt.

Hier bietet sich die Erfassung der z-Koordinate, also der Tauchtiefe an. Sie wird über einen Drucksensor, der den hydrostatischen Wasserdruck aufnimmt, bestimmt. Die Tiefe errechnet sich dann aus dem Messwert in Abhängigkeit der Dichte des umgebenden Wassers.

Die Beschleunigung eines UW-Fahrzeugs lässt sich gemäß (Gl. 31) berechnen /15/. Nachfolgend kann gezeigt werden, dass für den hier vorliegenden Einsatzfall dabei als äußere Kraft der Schwerkraftvektor von entscheidender Bedeutung ist.

$$\underline{m} \dot{\underline{a}} = \vec{F}_i + \vec{F}_G + \vec{F}_A + \vec{F}_H + \vec{F}_K + \sum_i \vec{F}_{T_i} \quad (\text{Gl. 31})$$

mit

- $\underline{m}$  : verallgemeinerte Massenmatrix
- $\dot{\underline{a}}$  : verallgemeinerter sechsdimensionaler Beschleunigungsvektor
- $\vec{F}_i$  : Trägheitskraft und -moment
- $\vec{F}_A$  : Auftriebskraft
- $\vec{F}_H$  : hydrodynamische Kräfte und Momente
- $\vec{F}_G$  : Gewichtskraft und -moment
- $\vec{F}_K$  : Kabelkraft und -moment
- $\vec{F}_{T_i}$  : Antriebskraft und -moment des i-ten Propellers

Zur Erfassung der auftretenden Gesamtbeschleunigung des Fahrzeugs sind drei Beschleunigungssensoren (dargestellt als Kästen) in einem orthogonalen System angeordnet und fest mit dem Fahrzeug verbunden (Abb. 30). Für den Fall, dass nur die Erdbeschleunigung auf die Anordnung wirkt, kann die Lage des Schwerkraftvektors  $\vec{g} = (g_x, g_y, g_z)$  und damit des Fahrzeugs im Sensorkoordinatensystem direkt aus den drei Sensormesswerten  $\vec{a}_m = (a_{mx}, a_{my}, a_{mz})$  gemäß

$$\vec{g} = \vec{a}_m \quad (\text{Gl. 32})$$

ermittelt werden.

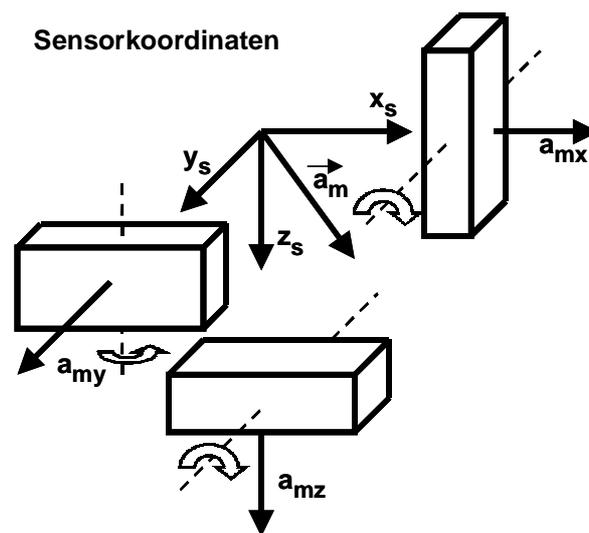


Abb. 30: Anordnung der Beschleunigungssensoren

Die kapazitiven Beschleunigungssensoren (Fa. VDO-Kienzle, Typ 2166.10) arbeiten in einem Bereich von  $\pm 2$  g und liefern eine der Beschleunigung proportionale Gleichspannung, zu deren Erfassung ein 12-Bit-AD-Wandler nachgeschaltet und zur Datenverarbeitung an den Scheckkartenrechner angebunden ist. Der Einbau des Sensormoduls in das Unterwasserfahrzeug erfolgt in der Form, dass die einzelnen Sensorachsen ( $x_s, y_s, z_s$ ) mit den Fahrzeugachsen ( $x, y, z$ ) gemäß Abb. 7 übereinstimmen. Die Lösung

von Gleichung (Gl. 31) stellt kein triviales Problem dar. Grundsätzlich existiert immer eine Überlagerung von translatorischer und rotatorischer Bewegung, so dass zusätzlich drei Winkelbeschleunigungssensoren benötigt würden, um die rotatorischen Anteile zu erfassen.

Die Gesamtbeschleunigung, die von den einzelnen Sensoren erfasst wird, lässt sich aus einer additiven Überlagerung von Schwerkraft und dynamischen Beschleunigungen beschreiben. Dabei müssen nicht nur die translatorischen Beschleunigungen auf die Translationssensoren berücksichtigt werden, sondern auch je nach Lage des Drehpunktes die rotatorischen, was zu komplexen Berechnungen führt. Nachfolgend können allerdings einige Abschätzungen getroffen werden, die den Berechnungsaufwand erheblich vereinfachen.

Das Unterwasserfahrzeug wird in der Art ausgetrimmt, dass es nur über einen minimalen Restauftrieb von einigen Newton verfügt, der es im Fehlerfall selbsttätig an die Oberfläche zurückbringt. Die Auftriebskraft wird dabei nahezu von der Gewichtskraft kompensiert. Auf Grund dieser Gegebenheit hat die Auftriebskraft nur einen sehr geringen Anteil auf die Gesamtbeschleunigung des Fahrzeuges und kann vernachlässigt werden.

Die Kabelkräfte sind ebenfalls sehr gering, da es nur über einen geringen Restauftrieb verfügt. Es wirken nur dynamische Kräfte auf das Kabel und damit auch auf das Fahrzeug. Größere Kräfte treten nur in Sonderfällen, z. B. bei Verhaken oder Verklemmen, auf und stellen nicht den Normalfall dar. Sie sind daher ebenfalls als sehr gering anzusehen.

Auf Grund fehlender oder nur minimaler Strömungen in kerntechnischen Anlagen treten hydrodynamische Kräfte und Momente im Wesentlichen nur durch die Eigenbewegung des Fahrzeuges auf. Da diese hydrodynamischen Anteile alle geschwindigkeitsabhängig sind, kann das Fahrzeug maximal beschleunigt werden, wenn die Geschwindigkeit Null ist. Je höher die Geschwindigkeit wird desto kleiner wird auch die erreichbare Beschleunigung. Die maximale Beschleunigung  $a_{\max}$  kann also bei  $v=0$  unter Vernachlässigung von Auftriebskraft, hydrodynamischer und Kabelkraft bestimmt werden zu:

$$\underline{\vec{m}}\vec{a} = \sum_i \vec{F}_{T_i} \quad (\text{Gl. 33})$$

Die von den Propellern maximal eingebrachten Kräfte betragen 6,5 N pro Antrieb. Unter der Voraussetzung, dass alle horizontalen und vertikalen Antriebe mit der gleichen Schubkraft arbeiten, lässt sich der prozentuale Betrag der Beschleunigung bezogen auf die Erdbeschleunigung ermitteln zu:

$$\begin{aligned} f_{\max} &= \frac{|\vec{a}|}{g} 100\% \\ &= \frac{\sqrt{(n \cdot F_{\text{Antrieb}})^2 + (k \cdot F_{\text{Antrieb}})^2}}{m_{\text{Fahrzeug}} + l \cdot m_{\text{Antrieb}}} \frac{100}{g} \% \\ &= \frac{F_{\text{Antrieb}} \sqrt{n^2 + k^2}}{m_{\text{Fahrzeug}} + l \cdot m_{\text{Antrieb}}} \frac{100}{g} \% \\ &= \lambda F_{\text{Antrieb}} \end{aligned} \quad (\text{Gl. 34})$$

Dabei gibt  $l$  die Anzahl der Antriebsmodule (mit  $l=2,4$ ),  $n$  die Anzahl der horizontalen Antriebe (mit  $n=2,4$  und  $n \leq l$ ) und  $k$  die Anzahl der vertikalen Antriebe (mit  $k=2,4$  und  $k \leq l$ ) an. Bei einer Fahrzeugmasse von  $m_{\text{Fahrzeug}}=25$  kg und der Masse eines Propellerdoppelantriebs von  $m_{\text{Antrieb}}=2$  kg ergeben sich die max. Fehler gemäß Tabelle 7.

Tabelle 7 : Fehlerbetrachtung Lagesensor

$l$	$n,k$	$\lambda$ [%/N]	$f_{\text{max}}$ [%]
2	2,0 oder 0,2	0,7	4,6
2	2,2	0,99	6,44
4	2,0 und 0,2	0,66	4,29
4	4,0 und 0,4	1,32	8,58
4	4,2 und 2,4	1,47	9,56
4	4,4	1,86	12,1

Auf Grund der geringen Beschleunigungen, die durch Antriebe, durch den Auftrieb oder Strömungen auf das Unterwasserfahrzeuges wirken, kann der dynamische Anteil jedoch für viele Betriebsfälle vernachlässigt werden; daher lassen sich die  $x,y,z$ -Komponenten des Schwerkraftvektors im Fahrzeug-Koordinatensystem direkt aus den Messwerten ablesen.

Typischerweise wird das Fahrzeug nicht immer mit allen Antrieben gleichzeitig und auch nicht mit dem maximal möglichen Schub bewegt. Fahrversuche haben ergeben, dass die Abweichung im Normalbetrieb auch bei vier Antrieben unter 5 % liegen. Größere Beschleunigungen können nur kurzzeitig, z. B. bei einer Kollision, auftreten und sind als impulsartig anzusehen. Solche „Ausreißer“ lassen sich softwaretechnisch mittels einer Plausibilitätskontrolle leicht eliminieren, eine zusätzlich nachgeschaltete Mittelwertbildung erlaubt Schwankungen zu unterdrücken.

Zur Ermittlung der Messgenauigkeit wird das Sensorsystem im stationären Betrieb vermessen. Da der Betrag des Beschleunigungsvektors als konstant anzusehen ist, wird er zu eins normiert. Damit ergeben sich die normierten Messwerte zu:

$$\begin{aligned} a_{mx0} &= a_{mx} / |\vec{g}| \\ a_{my0} &= a_{my} / |\vec{g}| \\ a_{mz0} &= a_{mz} / |\vec{g}| \end{aligned} \quad (\text{Gl. 35})$$

Die kartesischen Koordinaten werden in Kugelkoordinaten umgerechnet, da Winkelangaben zur messtechnischen Überprüfung besser geeignet sind. Die beiden Winkel  $\theta_{\text{am}}$  und  $\varphi_{\text{am}}$  lassen sich über die nachfolgenden Beziehungen ermitteln zu /44/:

$$\begin{aligned} a_{mx0} &= \sin(\theta_{\text{am}}) \\ a_{my0} &= \sin(\varphi_{\text{am}}) \cos(\theta_{\text{am}}) \\ a_{mz0} &= \cos(\varphi_{\text{am}}) \sin(\theta_{\text{am}}) \end{aligned} \quad (\text{Gl. 36})$$

Der Rollwinkel  $\alpha$  und der Neigungswinkel  $\beta$  ergeben sich zu:

$$\begin{aligned} \alpha &= \varphi_{\text{am}} \\ \beta &= 90^\circ - \theta_{\text{am}} \end{aligned} \quad (\text{Gl. 37})$$

Ein Vergleich der voreingestellten Winkel mit den Messwerten des Sensorsystem ergibt eine Messgenauigkeit von 0,5 Grad.

Mit dem hier vorgestellten Sensorsystem lassen sich damit die Tauchtiefe und die relative Lage des Schwerkraftvektors bestimmen. Drehungen um den Schwerkraftvektor sowie die beiden Koordinaten  $x$  und  $y$  im globalen KS sind damit nicht erfassbar. Die Messwerte lassen sich für diverse Aufgaben im Bereich Visualisierung und Steuerungstechnik einsetzen. Hierauf wird in Kap. 4.6 näher eingegangen.

### 3.5.6 Baukastensystem für Roboterarme

Für unterschiedliche Arbeiten im Unterwasserbereich wurde ein Roboterarm im Baukastenprinzip bestehend aus Gelenk- und Verbindungsmodulen entwickelt /56/. Auch hier setzt sich der modulare durch den Endanwender konfigurierbare Gedanke durch und bildet damit mit TRIBUN eine konzeptionelle Einheit.

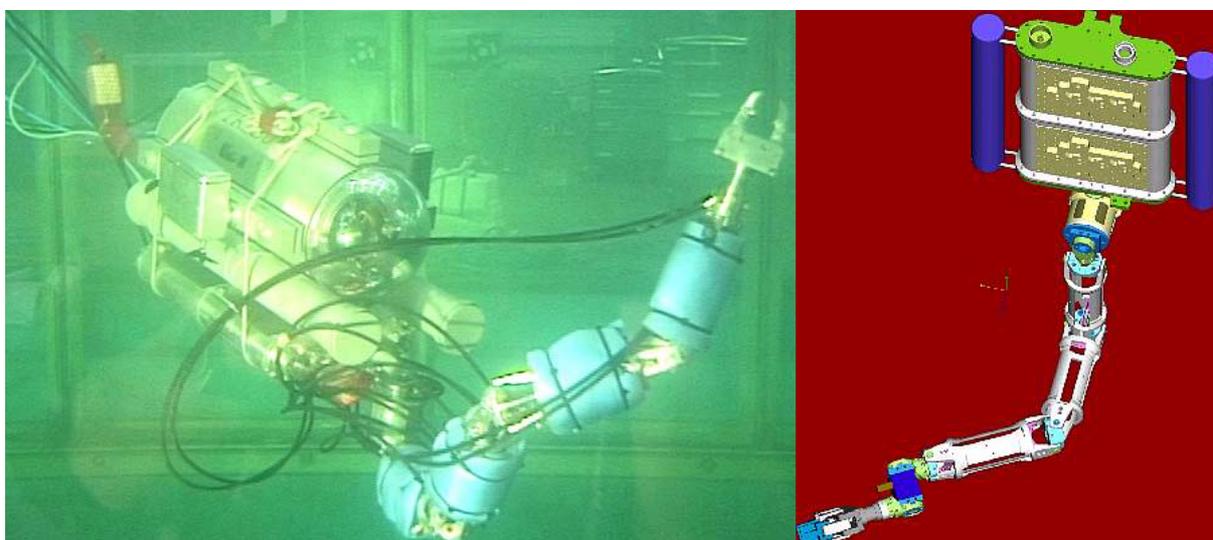


Abb. 31: TRIBUN mit Roboterarm in Kardan-Gelenkausführung

Die Module sind beliebig miteinander kombinierbar, so dass sich je nach Anwendung ein optimal geeignetes Handhabungsgerät konfigurieren lässt. Als angetriebene Module stehen hydraulische Drehgelenke, Kardangelenke sowie flexible Elemente zur Verfügung, die direkt oder über Verbindungselemente miteinander verbunden werden können. Für herkömmliche Anwendungen eignen sich die Drehgelenke, die über die Verbindungselemente zu bekannten Roboterkinematiken kombiniert werden können. In beengten Geometrien lässt sich ein Vielgelenkarm nutzen, der aus einer Folge von Dreh- und Kardangelenken besteht (Abb. 31 rechts).

Mit diesem Arm lassen sich Hindernisse umgreifen und Arbeiten in unzugänglichen Umgebungen durchführen. Zum Aufnehmen und Halten von Bauteilen wurden Greifer entwickelt, die nach unterschiedlichen kinematischen Prinzipien aufgebaut sind.

Die Steuerung aller mechanischen Komponenten erfolgt PC-basiert über das CAN-Bus-System ebenfalls über das TRIBUN-Steuerungssystem. Der Manipulator ist so ausgelegt, dass er sowohl an einer herkömmlichen Führungsmaschine als auch an TRIBUN adaptiert werden kann (Abb. 31 links) /57/.

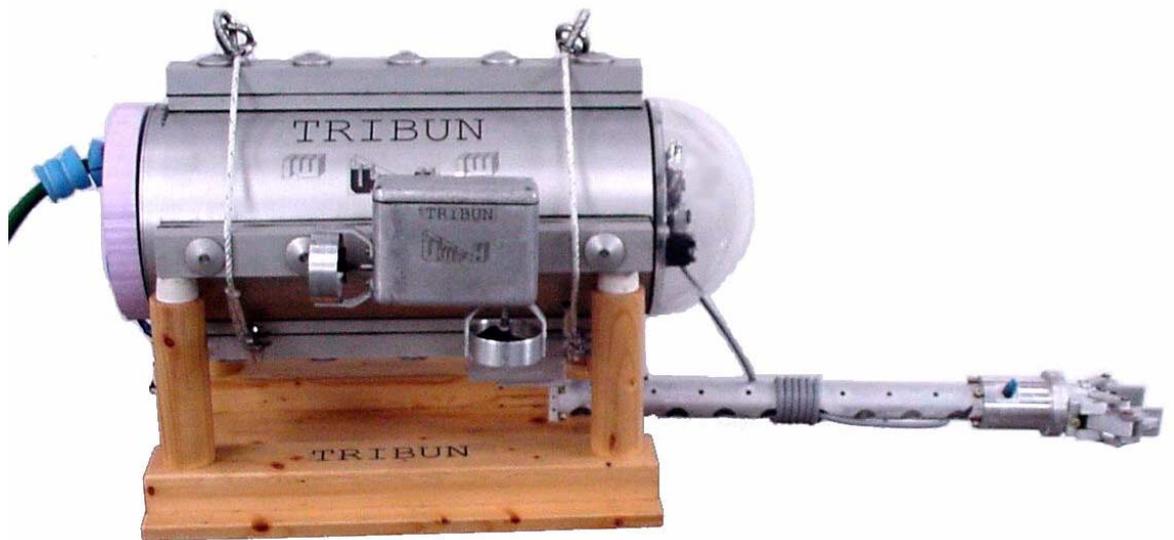


Abb. 32: TRIBUN mit Greifer

Aus demselben Baukastensystem stammt auch eine Stangen- und Greiferkombinationen (Abb. 32). Der elektrisch angetriebene Parallelgreifer mit einem Verfahrweg von 60 mm kann ebenfalls direkt über das TRIBUN-Stecksystem angebunden und versorgt werden.

### 3.6 Mensch-Maschine-Schnittstellen

Die wesentliche Aufgabe für eine bedienergestützte Steuerung ist die Erfassung der vom Bediener ausführbaren Aktionen und deren Umsetzung in verständliche Steuerbefehle für die Maschine. Dies erfolgt über Mensch-Maschine-Schnittstellen, die in der Regel kurz als MMI bezeichnet werden (engl. Man Machine Interface). Üblicherweise erfolgt die Steuerung oft manuell, also direkt mit den Händen. Neuere Entwicklungen basieren auf der Einbeziehung auch weiterer Benutzeraktionen wie der Erfassung der Kopf- oder der Augenbewegung, der Positionserfassung des Bediener oder auch der Spracherkennung. Dementsprechend wurde eine Vielzahl von MMIs entwickelt, die für diverse Steuerungsaufgaben eingesetzt werden können. Für das Unterwasserfahrzeug TRIBUN muss dabei sowohl eine Steuerung der Fahrzeugbewegung als auch der diversen Komponenten wie Manipulatoren, Führungsmaschine oder auch der Werkzeuge realisiert werden. Nachfolgend werden einige Eingabegeräte, die an einen PC angebunden werden können, benannt und deren Eignung für den hier vorliegenden Einsatzfall näher erläutert.

Computer-Tastaturen sind gut geeignet, um diverse Einzelaktionen direkt anzusteuern, aber zu umständlich für die Steuerung der Bewegung, da vom Bediener eine hohe Transferleistung notwendig ist, um Tasten mit Bewegungen in Verbindung zu setzen. Ein Gamepad ist in der Regel mit einer kleinen Tastatur vergleichbar und weist die gleichen Eigenschaften auf. Unterschiede bestehen in der Größe und Anordnung der Tasten. Inzwischen existieren zwar auch Pads mit eingebautem Joystick, wobei dieser meist aber als ein digitaler Joystick ausgeführt ist.

Für die Erfassung von zwei Freiheitsgraden stehen verschiedene Lösungen bereit. Die Computer-Maus erfasst eine Relativbewegung, wobei sich die Nullstellung nicht direkt aus der Mausposition ablesen lässt. Touchpads hingegen ermitteln eine Absolutposition und finden in der Regel bei tragbaren Computern Verwendung, wo ihr Vorteil der geringen Abmessungen genutzt wird. Auch hier besteht das Problem der genauen Nullpunktsfestlegung, so dass beide für Steuerungsaufgaben eher ungeeignet sind. Der einfache PC-Joystick ist relativ robust und kostengünstig in entsprechender Qualität verfügbar. Er erfasst ebenfalls eine Absolutposition und verfügt über eine definierte Nullstellung. Sein Einsatz ist gegenüber der Maus- oder Touchpad-Lösung vorzuziehen.

Seit einigen Jahren existieren sogenannte 3D-Mäuse. Im Gegensatz zur normalen Maus bewegt man bei ihnen keine Kugel in einem Gehäuse rollend über den Tisch, sondern die Kugel wird mit der Hand selbst gesteuert, wobei diese frei beweglich gelagert ist. Dadurch kann man die Kugel translatorisch in allen drei Richtungen verschieben, sowie rotatorisch um alle drei Achsen drehen.

Um Handbewegungen und -stellungen zu erfassen, wurden Datenhandschuhe (Data Gloves) entwickelt. Dazu sind z. B. Glasfasern auf einem normalen Handschuh aufgenäht. Eingestrahktes Licht in den Fasern wird am anderen Ende mittels Photodetektoren gemessen. Die Intensität des eingestrahkten Lichts ist dann ein Maß für die Stärke der Krümmung der Finger. Ein Data Glove ist als Steuereingabe für TRIBUN wegen der nicht logischen Zuordnung von Fingerbewegungen zu Fahrzeugbewegungen ungeeignet. Im Rahmen der Steuerung eines Manipulators/Greifers wäre der Einsatz eines Datenhandschuhs aber in Betracht zu ziehen.

Um die Stellung der Hand relativ zu einem festen Punkt zu ermitteln, wurden 3D-Positionssensoren entwickelt, die die Position der Hand im Raum bestimmen. So gibt es

neben den Ultraschallsensoren auch Magnetfeldsonden, Radarmessungen, Messungen mittels Infrarotkameras und Leuchtdioden. Für den Benutzer dieser Geräte stellt sich aber schnell das Problem der Ermüdung seiner Arme ein. Deswegen ist der Einsatz solcher Positionssensoren für die Steuerung eines ROVs nur bedingt geeignet.

Um die Kopfbewegung des Bedieners zu erfassen, stehen sog. Head-Tracker zur Verfügung. Sie basieren in der Regel auf Sensoren, die das Magnetfeld messen, welches den Bediener umgibt. Bei Drehung, Neigung oder Nicken des Kopfes des Bedieners verändern sich ebenfalls die von den Sensoren erfassten Anteile des Magnetfeldes. Die Magnetfeldänderungen lassen sich in Winkel bzw. Winkeländerungen umrechnen, die dann zu Steuerungszwecken genutzt werden können. Der Vorteil ist, dass die natürlichen Kopfbewegung genutzt wird und eine intuitive Steuerung möglich ist.

Die Rückwirkung auf den Bediener erfolgt hauptsächlich über Monitore aller Art wie PC-Monitore zur Darstellung der Bedienoberfläche oder Videomonitore zur Darstellung von Kamerabildern. In der letzten Zeit haben sich auch vom Bediener tragbare Monitore (Head-Mounted-Displays oder Videobrillen) durchgesetzt. Falls sie über zwei getrennte Kanäle zur Einspeisung der Bilder für rechtes und linkes Auge verfügen, so kann darüber auch ein stereoskopisches Sehen ermöglicht werden.

### 3.6.1 Schnittstellen zur Fahrzeug- und Werkzeugsteuerung

Die oben aufgeführten Eingabegeräte lassen sich weiterhin nach ihren speziellen Funktionen unterscheiden. Als wichtigstes Kriterium zur Unterscheidung dient dabei die Anzahl der gleichzeitig zu beeinflussenden Freiheitsgrade. Zwar lässt sich diese Zahl bei Verwendung von z. B. zwei Geräten verdoppeln, allerdings sind dann auch beide Hände des Bedieners für die Bedienung erforderlich. Somit kann der Benutzer nur schlecht weitere Steuerungskomponenten des ROVs bedienen. Damit ist auch eine Verwendung von mehreren Joysticks zur Fahrzeugsteuerung als ungeeignet anzusehen, wobei zusätzlich keine direkte Zuordnung von Joystickachsen und Bewegungsachsen ersichtlich ist.

Ein weiteres maßgebliches Kriterium ist die Präzision der Eingabegeräte. So bringt es keinen Vorteil, eine Vielzahl von Freiheitsgraden gleichzeitig manipulieren zu können, wenn dabei unerwünschte Eingaben durch fehlende Präzision erfolgen. Oftmals ist es jedoch hilfreich, wenn man nur eine Hand zum Steuern des Bootes benötigt, da mit der freien Hand dann z. B. Eingaben am Visualisierungsrechner vorgenommen werden können. Als geeignetes Eingabegerät, das es erlaubt, mit nur einer Hand alle sechs Freiheitsgrade gleichzeitig zu manipulieren, ist hier eine 3D-Maus anzusehen. Zur Auswahl kommen dabei folgende am Markt verfügbaren Geräte:

Tabelle 8 : Übersicht 3D-Mäuse

Bezeichnung	Hersteller	Preis
SpaceMaster	Gesellschaft für Anwender und Systemsoftware	DM 1700,-
SpaceBall	Spacotec IMC Corporation	\$ 850,-
SpaceMouse	LogiCad3D GmbH	DM 600,-

In Tabelle 8 wurde bewusst nur der Preis aufgeführt, da eine Unterscheidung durch technische Daten nur bedingt durchführbar ist /59/,/60/,/61/. Bei allen Systemen kommt

ein optisches Messverfahren zur Erfassung der sechs Freiheitsgrade zum Einsatz. Somit ergeben sich die Unterschiede hauptsächlich im Design der Maus und im Preis. Die SpaceMouse von LogiCad3D (entwickelt vom Deutschen Luft- und Raumfahrtzentrum, DLR) ist eindeutig am günstigsten und wurde deswegen ausgewählt.

Die messtechnische Erfassung der Verschiebung und Verdrehung wird mittels einer an Federn aufgehängten Kappe gemessen. Die Messung erfolgt durch ein berührungsloses und verschleißfreies, optoelektronisches Meßsystem. Somit genügt schon ein leichter Fingerdruck gegen die Sensorkappe, um eine verwertbare Auslenkung zu erhalten. Die sehr ergonomische Gestaltung der Maus ermöglicht eine bequeme und somit ermüdungsfreie Auflage des Handballens, während nur die Finger benötigt werden, um die Kappe auszulenken.

In Situationen, in denen ein Manövrieren auf engstem Raum oder mit höchster Genauigkeit erforderlich ist, ist es möglich, die SpaceMouse entsprechend zu konfigurieren. Dazu lassen sich die Translation sowie die Rotation einzeln deaktivieren bzw. aktivieren und ihre Empfindlichkeiten einstellen. Außerdem besitzt sie einen sogenannten 'dominant mode' bei dem nur die größte Auslenkung der Sensorkappe übermittelt wird.

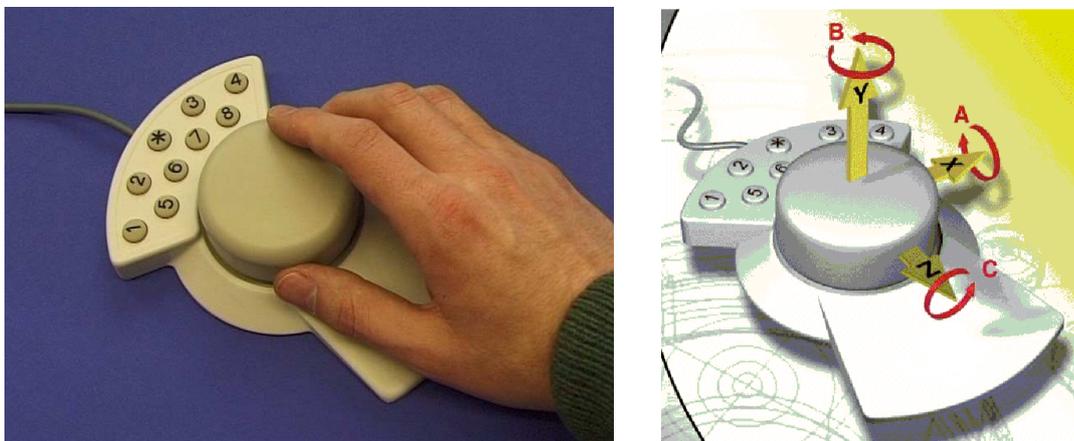


Abb. 33: Eingabegerät Spacemouse (l.) und erfasste Freiheitsgrade/Achsen (r.)

Die Kommunikation der SpaceMouse erfolgt über eine serielle Schnittstelle vom Typ RS232. Aufgrund der Konstruktion der Maus ist eine zusätzliche Spannungsversorgung nicht notwendig, da sie sich aus der Schnittstelle über das DTR-Signal mit Strom versorgt. Die Bewegungsinformationen der sechs Freiheitsgrade der SpaceMouse für Translation und Rotation können somit direkt vom Leitreechner abgefragt und in Abhängigkeit von der aktuellen Antriebskonfiguration in Steuerinformationen für die Antriebe umgerechnet werden. Die Vorgabe der Drehzahlen erfolgt dabei proportional der Auslenkungen.

Neben der SpaceMouse wird auch die Anbindung eines Joysticks für einfachere Steuerungsaufgaben realisiert. Ein analoger PC-Joystick verfügt über eine Zweiachsensteuerung. Dabei werden mittels eines Griffstückes zwei Dreh- oder Schiebewiderstände verstellt. Seine Vorteile liegen in der einfachen Bedienung und seiner Robustheit. Weiterhin ist die Joystickschnittstelle durch den PC-Bereich genormt. Dadurch ist es dem Bediener möglich, einen beliebigen am Markt verfügbaren analogen Joystick zu wählen. Einsatzgebiete sind hierbei Bereiche, wo speziell zwei Achsen gleichzeitig gesteuert werden müssen, wie z. B. bei der Führungsmaschine.

### 3.6.2 Schnittstellen des Kamera- und Videosystems

Für die dreidimensionale Aufbereitung der Videodaten dient ein sogenanntes Head-Mounted-Display (HMD). Um den Bediener bei Steuerungsaufgaben zu entlasten, erfolgt die Kamerasteuerung nicht über Joysticks oder Taster, sondern über ein Trackingmodul, das die Kopfbewegung des Bedieners erfasst und diese Positionsdaten an die Kameras weiterleitet. Dem Bediener wird damit eine Rundumsicht im Rahmen des Bewegungsbereichs der Kameras geboten.

Die ergonomischen und optischen Merkmale eines Head-Mounted-Displays sind bei der Auswahl von entscheidender Bedeutung [62],[63],[64],[65],[66],[67],[68]. Als Bestandteil des Steuersystems der Handhabungsfahrzeuge muss damit gerechnet werden, dass das Display schnell an mehrere Bediener angepasst werden muss. Durch die Operationszeiten des Unterwasserroboters bis zu mehreren Stunden ist ein nicht angenehm zu tragendes Head-Mounted-Display praktisch wertlos. Um Beeinträchtigungen im späteren Betrieb zu vermeiden, wurde bei der Auswahl des Displays auf folgende Kriterien geachtet:

Abstand der Linsen vom Auge:

Ist der Abstand zwischen der ersten Linse vor dem Auge kleiner als ca. 25 mm, ist es für Brillenträger generell nicht möglich, das HMD zu benutzen.

Gewicht des gesamten HMDs:

Soll das HMD über einen längeren Zeitraum getragen werden, so ist das Gesamtgewicht unbedingt so gering wie möglich zu halten. Problematisch bei HMDs ist auch der durch die Linsen im Sichtfeld weit nach vorne verlagerte Schwerpunkt. Viele Systeme versuchen diesen durch ein Gegengewicht am Hinterkopf in die Kopfmitte zu verschieben, wodurch das Gesamtgewicht zunimmt.

Belüftung:

Einige HMDs sind als eine Art Helm ausgeführt. Die Folge nach längerem Tragen ist Schweißbildung, die im Extremfall die Linsen beschlagen lassen kann. Es sollte deshalb auf möglichst wenig Kontaktpunkte zum Kopf und gute Belüftung geachtet werden.

Justierbarkeit:

Für einen schnellen Bedienerwechsel müssen leicht zu bedienende Einstellmöglichkeiten für verschiedene Kopfformen vorgesehen sein. Um eine einwandfreie Darstellung zu gewährleisten, sollten außerdem gut zugängliche Regler für den Pupillenabstand, Helligkeit und Kontrast vorhanden sein.

Darstellungsqualität:

Für entspanntes Sehen ist es notwendig, auf eine hochwertige Optik zu achten. Schlecht justierte Linsensysteme erzeugen Verzerrungen und der Benutzer neigt dazu, schnell zu ermüden. Die Darstellungsqualität hängt außerdem mit der Auflösung und dem maximal möglichen Blickwinkel zusammen. Ein großer Blickwinkel bei geringer Auflösung lässt die Bilder unscharf erscheinen.

Ein weiterer wichtiger Aspekt für die Auswahl des HMD ist das Eingangssignal. Fast ausschließlich alle HMDs der unteren Preisskala bieten als Signaleingang einen dem VGA-Standard entsprechenden Eingang. Dieses VGA-Signal wird üblicherweise von

PC-Grafikkarten erzeugt. Um also die Kamerabilder, deren Signal als herkömmliche FBAS-Modulation vorliegt, auf ein VGA-Signal mit einer anderen Wiederholfrequenz und Auflösung umzusetzen, müssten die beiden Kamerabilder in den PC-Speicher digitalisiert werden. Im nächsten Schritt werden die beiden Bilder der unterschiedlichen Kameraansichten zu einem Bild zusammengefügt. Die für HMDs übliche Regelung sieht vor, alle ungeraden Bildzeilen auf das linke und alle geraden auf das rechte Auge abzubilden. Das so im PC-Speicher aufgebaute Bild kann dann über die PC-Grafikkarte in das HMD eingespielt werden. Der Nachteil einer solchen Lösung ist offensichtlich, da zu den für das Kamerasystem notwendigen Komponenten weitere für zwei Digitalisierer und eine leistungsfähige Software hinzu kommen, die immerhin die Bilddatenrate von mindestens

$$\begin{aligned}
 f &= n_{\text{Kamera}} \cdot f_{\text{Bild}} \cdot P_{\text{horizontal}} \cdot P_{\text{vertikal}} \cdot n_{\text{Farbe}} \\
 &= 2 \cdot 50\text{Hz} \cdot 300 \cdot 256 \cdot 8\text{Bit} \\
 &= 7,32 \frac{\text{MByte}}{\text{s}} \qquad \qquad \qquad (\text{Gl. 38})
 \end{aligned}$$

verarbeiten muss (typische Auflösung für HMDs: 300\*256 Bildpunkte; Farbtiefe: 8 Bit; Bildrate: 50 Bilder/Sekunde; zwei Kameras). Die Rechenleistung des PCs wird bei einer solchen Lösung im Wesentlichen für die Bildaufbereitung verbraucht.

Eine Auswahl einiger zum Zeitpunkt dieser Arbeit auf dem Markt erhältlichen HMDs ist in Tabelle 9 wiedergegeben.

Tabelle 9 : Technische Daten und Preise einiger verfügbarer HMDs (Stand 1998)

Firma	Produkt	Technik	Blickfeld (hori.*vert.)	Farbpunkte (hori.*vert.)	Gewicht	Preis (US-\$)
Virtual Reality	Model 133	2*CRT	116° * 58°	2560 * 1024	1130 g	ca. \$65000
Leep Systems	CyberFace 5	4*LCD	140° * 110°	1170 * 234	keine Angabe	ca. \$45000
Fakespace	Boom 3C	2*CRT	140° * 90°	1280 * 1024	Gewicht vollständig kompensiert	ca. \$75000
Virtual Research	V8	2*CRT	48° * 36°	640 * 480	821 g	ca. \$11400
Kaiser Electro-Optics	ProView60	2*LCD	48° * 36°	640 * 480	900 g	\$11595
Liquid Image	X3	2*LCD	84° * 65°	640 * 480	keine Angabe	\$7000
Virtuality	Visette	2*LCD	40° * 30°	263 * 225	550 g	\$2000
General Reality	CE-200	2*LCD	22.5° * 16.8°	263 * 230	400 g	\$2500
Forte VR	VFX 1	2*LCD	53° * 35°	263 * 230	680 g	\$1000
Victormaxx	Cybermaxx	2*LCD	56° * 42°	263 * 230	400 g	\$800

Auffallend dabei ist die geringe Auswahl in der unteren Preisklasse (<\$ 5000). Dennoch muss auf ergonomische und optische Qualität Wert gelegt werden. Das HMD der Firma General Reality erfüllt alle für den Tragekomfort ausschlaggebenden Punkte. Es hebt sich außerdem durch wahlweise Ausstattung des Eingangs mit FBAS-PAL oder VGA-Standard von seinen Konkurrenten ab. Durch die Ausstattung mit FBAS-Eingängen kann auf die für andere HMDs benötigten Bilddigitalisierer und den PC verzichtet werden, so dass der Preis von \$ 2500 gegenüber den günstigeren HMDs gerechtfertigt ist.

Die Firma General Reality bietet passend zu dem HMD auch Tracking-Sensoren an. Diese erfassen wahlweise drei oder sechs Freiheitsgrade der Kopfposition und geben die Messdaten über eine aus dem Computerbereich bekannte serielle Schnittstelle aus.

Die Anforderungen des Kamerasystems an den Tracking-Sensor sind eine schnelle Positionserfassung, wobei die Positionsangabe von drei Freiheitsgraden alle für die Kameras wichtigen Informationen beinhaltet. Der Sensor arbeitet „quellenlos“, also ohne Montage von Licht- oder Magnetfeldemittern, um den Einsatz des Handhabungsfahrzeugs an häufig wechselnden Einsatzorten nicht zu verkomplizieren. Der Tracking-Sensor liefert drei Winkelangaben für die Drehung, das Nicken sowie das Rollen des Kopfes des Bedieners. Letzter Wert findet auf Grund des mechanischen Aufbaus keine Verwendung.

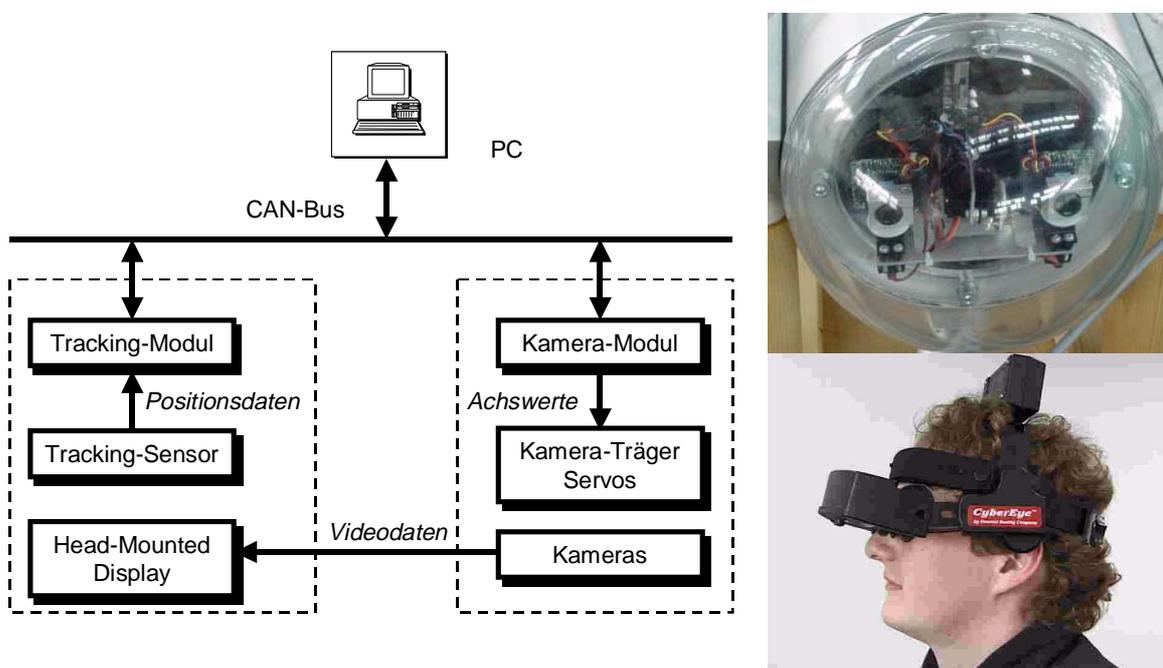


Abb. 34: Signalfluss des Kamerasystems (l.) und Stereokamera und Bediener mit HMD (r.)

Für beide Module, Kamera- und Trackingmodul, wird ein Scheckkartenrechner mit entsprechender Zusatzhardware zur Ansteuerung der Aktoren und Sensoren eingesetzt. Wie in Abb. 34 wiedergegeben ist, steuert das Tracking-Modul im Normalbetrieb das Kamerasystem direkt an, wenn es am Bus erkannt wird. Zu diesem Zweck wird an dem HMD ein Sensor befestigt und mit dem Trackingmodul verbunden. Der PC übernimmt dabei nur die Visualisierung der Moduldaten und wird nur optional zur Bedienung der Kamera verwendet, wenn das Trackermodul vorher deaktiviert wurde. Das Videosystem stellt damit eine vom PC-unabhängige Einheit dar.

Seit 1999 ist eine Videobrille der Firma Olympus mit der Bezeichnung *EyeTrek* auf dem Markt /69/. Entwickelt wurde sie als tragbarer, mobiler Bildschirm für den Home-Entertainment Bereich z. B. zum Fernsehen in Zusammenspiel mit einer Funkvideo-Übertragung oder auch zum Ansehen von Filmen mit einem tragbaren DVD-Player. Sie verfügt allerdings nur über einen Videoeingang, dessen Signale auf beide Displays übertragen werden und ist daher nicht stereofähig. Vorteil ist ihr geringes Gewicht (85 g), ein deutlich bessere Auflösung von jeweils 180.000 Bildpunkten pro Display und die offene Bauform, die es gestattet, dass der Bediener unter der Brille durchschauen kann. Dadurch ist es möglich, sowohl die Videobilder der Kamera als auch weitere Informationen z. B. von anderen Monitoren (Steuerung/3D-Visualisierung) gleichzeitig zu betrachten. Der Head-Tracker lässt sich ohne Probleme auch auf dieser Brille montieren und zur Steuerung einsetzen.

## 3.7 PC-Software für Steuerung und Bedienung

### 3.7.1 Softwarearchitektur

Die hohe Modularität des Systems erfordert eine gute Anpassungsfähigkeit der Leit-rechner-Software an die momentane Systemkonfiguration. Diese wird durch eine objekt-orientierte Programmierung und Realisierung einer leicht bedienbaren Benutzeroberfläche unter Windows NT gewährleistet. Die Steuerung des Unterwasser-Handhabungssystems kann dabei entweder über einen PC oder ein Laptop erfolgen. Über die grafische Bedienoberfläche sind alle notwendigen Funktionen für die Inbetriebnahme, Steuerung und Wartung der einzelnen Module, die Konfiguration des Gesamtsystems, die Visualisierung der Systemdaten sowie die Anbindung an die Simulation und die virtuelle Arbeitsumgebung zugänglich.

Wie auf Abb. 6 zu erkennen, gibt es eine Vielzahl von Sensor- und Aktorknoten im Handhabungssystem, die über den Feldbus vom Leit-rechner geeignet angesteuert werden müssen. Für das zuständige Steuerprogramm entstand die Forderung nach einem einheitlichen Steuerungskonzept, mit dessen Hilfe eine einfache Anpassung der Benutzeroberfläche auf die unterschiedlichen Konfigurationen des Systems ermöglicht wird.

Die gängigsten Programmiersprachen, die momentan im Computerbereich angewendet werden, lassen sich fünf hauptsächlichen Programmierstilen zuordnen:

- Prozedurorientiert: Abbildung in Algorithmen
- Objektorientiert: Zerlegung in Klassen und Objekte
- Logikorientiert: Angabe des Ziels (oft in Prädikantenkalkül ausgedrückt)
- Regelorientiert: Angabe von *If-Then*-Regeln
- Bedingungsorientiert: Darstellung durch Invariantenbeziehungen

Es lässt sich nicht der beste Programmierstil angeben, da die Eignung auch stark von der zu erstellenden Applikation abhängt. So werden z. B. Wissensdatenbanken meist regelorientiert, rechenintensive Algorithmen hingegen oft prozedurorientiert implementiert. Trotzdem hat sich in den letzten Jahren abgezeichnet, dass der objektorientierte Stil ein breites Applikationsspektrum mit guten Problemlösungen abdeckt /70/,/71/.

Nachfolgend wird ein objektorientiertes Konzept vorgestellt, das die Erstellung von Bedienoberflächen für den Leit-rechner eines kabelgeführten Handhabungssystems vereinfacht und damit ein allgemeines Steuerungskonzept anbietet. Die praktische Reali-

sierung erfolgt in der Programmiersprache C++ und wird mit dem Borland-Compiler 5.0 durchgeführt, wobei als grafische Benutzeroberfläche MS-Windows NT 4.0 dient.

### 3.7.2 Objektorientierte Zerlegung des Gesamtsystems

Den ersten Schritt hin zu einer objektorientierten Lösung eines zu implementierenden Systems bildet immer die objektorientierte Zerlegung. Die nachfolgenden Abschnitte beschäftigen sich mit dieser grundlegenden Aufgabe und stellen die wesentlichen Hauptfunktionen der in dieser Arbeit erzeugten Klassen und Objekte dar. Hierbei wird besonders auf deren Kooperation zur Bereitstellung der Systemfunktion eingegangen.

Vor der objektorientierten Zerlegung ist eine eingehende Beschäftigung mit dem physikalischen Gesamtsystem notwendig, welches in diesem Fall das Handhabungssystem darstellt. Das zu erstellende Steuerprogramm wird stark von der konkreten Systemausprägung bestimmt.

Das kabelgeführte Handhabungssystem verfügt über Aktoren und Sensoren, die alle über den gemeinsamen CAN-Bus mit der CAN-Schnittstelle im PC verbunden sind. Der Benutzer kommuniziert mit einer grafischen Bedienoberfläche, wobei er vor allem mit Fensterobjekten und darin angeordneten Bedienelementen arbeitet. Als Basis dient das Steuerprogramm im PC, welches auf der einen Seite die Umsetzung der Benutzereingaben in CAN-Befehle für das Handhabungsfahrzeug übernimmt. Andererseits muss es aber auch die über den CAN-Bus empfangenen Nachrichten zwecks Visualisierung und Auswertung geeignet verarbeiten.

Dieser erste Systemüberblick zeigt, dass man es in diesem Fall mit einer komplexen Analyseaufgabe zu tun hat, da es viele voneinander abhängige Komponenten im System gibt. So sind die Aktoren und Sensoren jeweils physikalischen Netzknoten mit eigener CAN-Adresse zugeordnet, so dass für eine Kommunikation diese funktionelle Verbindung bekannt sein und im System geeignet modelliert werden muss. Auch existieren Zuordnungen von Fensterelementen zu den Aktoren/Sensoren, da diese ja die Zustände im Handhabungssystem beeinflussen bzw. visualisieren sollen. Nebenbei spielt auch die Art der CAN-Schnittstelle im PC eine Rolle, da die konkrete Übertragung auf speziell dafür angepassten Bibliotheksfunktionen basiert.

Man kann aber schon an dieser Stelle das vom objektorientierten System zu lösende Hauptproblem erkennen. So wird nachfolgend in erster Linie eine Strategie entwickelt, die es ermöglicht, die funktionalen Zusammenhänge im System angemessen zu modellieren. Diese Modellierung muss bei Benutzereingaben auf der grafischen Oberfläche die Generierung einer eindeutigen CAN-Nachricht für den zugeordneten Aktor/Sensor gewährleisten. Andererseits müssen die CAN-Nachrichten vom Handhabungsfahrzeug an alle Bedienelemente weitergeleitet werden, die ein Interesse daran angemeldet haben. So wird das Rückgrat des zu erstellenden Steuersystems vom Mechanismus zur geeigneten Nachrichtenweiterleitung gebildet, wobei der Weg der Nachrichten zwischen Bedienoberfläche und Handhabungsfahrzeug von der jeweiligen aktuellen Betriebskonfiguration abhängt.

### 3.7.3 Identifizierung der Klassen und Objekte

Der erste Schritt der objektorientierten Zerlegung wird durch die Identifizierung der am System beteiligten Objekte gebildet. Hierbei kann auf die Ergebnisse aus der Betrachtung

tung des physikalischen Systems zurückgegriffen werden, in dem den dort erkennbaren Entitäten konkrete Objekte zugeordnet werden. Bei dem hier betrachteten System lassen sich folgende Elemente auf den ersten Blick als Objekte identifizieren:

- Netzwerkobjekte: PCCAN-Karte, CAN-Dongle
- Physikalische Knotenobjekte: verschiedene Scheckkarten-Knoten
- Dialogobjekte: Dialogfenster
- Oberflächenelemente: Button, Slider, Texte, Checkboxes, Radiobutton

Diesen direkt aus dem physikalischen System ableitbaren Objekten stehen bei genauere Betrachtung auch einige andere abstraktere gegenüber, die ebenfalls zur Erfüllung der Systemfunktion beitragen:

*CAN\_FRAME* stellt ein CAN-Paket dar und enthält die entsprechenden Datenelemente (Identifizier, RTR-Bit, Länge und CAN-Nutzdaten); ermöglicht die Handhabung dieser Daten als Einheit (anstelle von Bytekolonnen)

*ZYKLUSZEIT* stellt Funktionen bereit, um die möglichen Netzwerkschnittstellen im PC zyklisch abfragen zu können. (Unter Windows werden die Hardwareempfangsinterrupts nicht weitergereicht, so dass die Schnittstelle via Pollingverfahren schnell abgefragt werden muss, um Datenverluste zu vermeiden)

*SystemStatus* stellt das eigentliche Kernobjekt der Anwendung dar; in ihm werden nach Scannen einer Konfigurationsbeschreibung alle nötigen Objekte angelegt (den physikalischen Knoten im Unterwasserfahrzeug entsprechend) und mit den Bedienelementen auf der Benutzeroberfläche verbunden - es ist verantwortlich für den geordneten Informationsaustausch zwischen GUI-Oberfläche und den Knoten des Handhabungsfahrsystems; im System existiert nur ein solches Objekt, welches mit seinem Zustand den gesamten Systemstatus repräsentiert.

Man erkennt hier, dass sich einige Objekte sinnvollerweise von Basisklassen ableiten lassen. So können die PCCAN-Karte und das CAN-Dongle von einer gemeinsamen Netzwerkklasse, die verschiedenen Versionen der Scheckkarte von einer Klasse für physikalische Knotenobjekte und die unterschiedlichen Oberflächenelemente von einer weiteren GUI-Klasse abgeleitet werden. Somit ergeben sich schon auf dieser hohen Abstraktionsebene Möglichkeiten zum Aufbau einer Klassenhierarchie.

### 3.7.4 Identifizierung der Semantik der Klassen und Objekte

Nach der Bezeichnung der Objekte mit Ansätzen für eine erste Zusammenfassung in Klassen werden nun ihre unterschiedlichen Aufgaben und Beziehungen untereinander beschrieben. Teilweise wurden schon im vorigen Abschnitt Objektfunktionalitäten aufgezeigt, wobei diese hier nun systematisch für alle vorhandenen Objekte aufgeführt werden.

Netzwerkobjekte sind zuständig für das Ansprechen der konkreten CAN-Schnittstellenkarte im PC und die Bereitstellung von Funktionen zur Initialisierung der Hardware. Weiterhin enthalten sie die grundlegenden Übertragungsbefehle (Senden/Empfangen) und dienen der Einstellung der Baudrate und der I/O-Adresse.

Die Knotenobjekte bilden die physikalischen Knoten des Handhabungssystems ab und legen deren Verhalten im System fest. Sie erhalten die eingehenden CAN-Pakete, die

ihrem Knotenidentifizier entsprechen und sind für die Benachrichtigung der zugeordneten Ausgabeelemente auf der Oberfläche zuständig. Weiterhin werden Sendeanforderungen von Bedienelementen von ihnen bearbeitet und sie speichern den momentanen Betriebszustand des physikalischen Knotens.

Die *CAN\_Frames* bilden die Repräsentation von CAN-Datenpaketen und weisen alle dort gespeicherten Informationsfelder auf (ID, Länge, RTR-Bit, CAN-Daten). Sie ermöglichen eine einheitliche Behandlung von CAN-Paketen im Gesamtsystem.

Die Dialogobjekte dienen als Aufnahmebehälter für die Oberflächenelemente und werden zur Anzeige des Dialogfensters auf der Oberfläche benötigt.

Die Oberflächenelemente verarbeiten die Benutzereingaben auf der Oberfläche mit in ihnen definierten Funktionen und senden die von diesen zusammengesetzten CAN-Nachrichten an den zugehörigen physikalischen Knoten. Ausgabeelemente visualisieren die empfangenen Datenpakete entsprechend des in ihnen festgelegten Verhaltens.

*Zykluszeit* bietet einen Mechanismus an, Funktionen im Millisekundenbereich aufzurufen.

*Systemstatus* erstellt aufgrund einer Konfigurationsbeschreibung die Verbindungen der physikalischen Knoten mit den Oberflächenelementen. Es initialisiert das Netzwerk, stellt das zyklische Aufrufen der Empfangsfunktion sicher, erhält vom Netzwerkobjekt die eingehenden CAN-Meldungen aus dem System und verteilt diese auf die dem Identifizier entsprechenden Knoten. Es gibt eingehende Sendewünsche der physikalischen Knoten an das Netzwerk weiter und hat im System dazu als einziges Objekt direkten Zugriff auf diese Klasse.

### 3.7.5 Systemdesign

Mit den herausgearbeiteten Anforderungen lässt sich ein vorläufiges Systemdesign erstellen (Abb. 35). Hierbei sind die Objekte und Klassen dargestellt, die über Linien miteinander in Beziehung stehen.

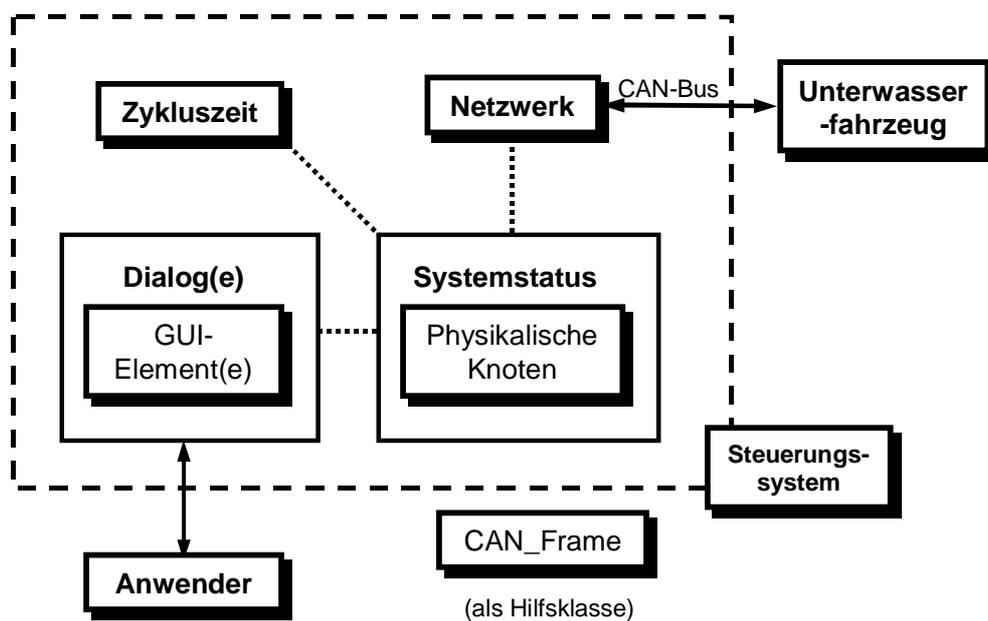


Abb. 35: Systemdesign auf hoher Abstraktionsebene

Man kann aus der Darstellung ablesen, dass nur das Objekt *SystemStatus* über *Netzwerk* eine Verbindung zum CAN-Bus hat. Die Verbindung zu *Zykluszeit* erklärt sich dadurch, dass für die Polling-Abfrage der Schnittstellenkarte im PC eine Funktion in *SystemStatus* zyklisch aufgerufen werden muss. Der Anwender hat über einen Dialog Zugriff auf den *SystemStatus*, der verschiedene GUI-Elemente enthalten kann. Da aber auch mehrere Dialoge gleichzeitig am Bildschirm geöffnet sein können, können im System verschiedene Dialoge gleichzeitig existieren. Dieses gilt natürlich auch für die GUI-Elemente, von denen in einem Dialog mehrere angezeigt werden können. Im *SystemStatus* können ebenfalls mehrere physikalische Knoten registriert werden, da im Handhabungsfahrzeug i. A. mehr als nur ein physikalischer Konten installiert werden kann. So bildet diese Systemstruktur den Ausgangspunkt für die weitere Verfeinerung des Objektmodells, welches dann durch nähere Klassenspezifikationen die Grundlage der Implementierung ist.

### 3.7.6 Klassenimplementierungen der CAN-Kommunikation

In den folgenden Abschnitten werden die unterschiedlichen Systemfunktionalitäten dargestellt, wobei besonders auf die Kooperation der Klassen untereinander eingegangen wird. Ziel ist es, eine klare Vorstellung von den Mechanismen zu vermitteln, die in ihrer Gesamtheit die Systemfunktion verwirklichen.

Da das Handhabungssystem über den CAN-Bus gesteuert wird, ist eine Klasse zur Kapselung des Frameformates sinnvoll, die für eine einheitliche Behandlung der CAN-Daten im Gesamtsystem sorgt. So sind in der Klasse *CAN\_Frame* Datenfelder für den Identifier, die Datenbytes sowie das RTR-Bit definiert und als geschützt (protected) deklariert, da der Zugriff auf sie nur über Schnittstellenfunktionen der Klasse ermöglicht werden soll.

Die Netzwerkklassse verfügt für eine Datenübertragung über verschiedene Grundfunktionen zur Initialisierung der CAN-Schnittstelle, sowie zum Senden und Empfangen von Daten. Für den Zugriff auf den CAN-Bus stehen mit der PC-CAN-Karte und dem PCAN-Dongle zwei unterschiedliche Realisierungen zur Verfügung, wobei jede über spezielle, vom Hersteller mitgelieferte Bibliotheksfunktionen die hardwaremäßige Umsetzung der CAN-Kommunikation realisiert.

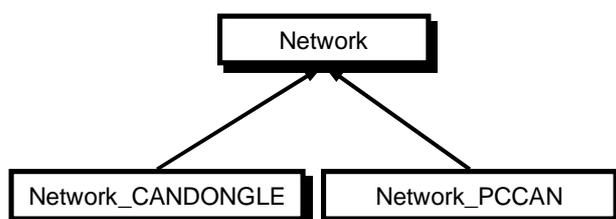


Abb. 36: Netzwerkklassen

Um dem Anwender eine möglichst transparente Nutzung der Netzwerkklassse unabhängig von der jeweiligen Hardware zu ermöglichen, bietet sich eine Klassenhierarchie an (Abb. 36). Hierbei stellt *Network* die abstrakte Basisklasse für *Network\_PCCAN* und *Network\_CANDONGLE* dar. Von ihr können keine Objekte gebildet werden,

sie legt aber mit den in den Unterklassen zu definierenden Funktionen die allgemeine Schnittstelle fest. So werden die oben beschriebenen Funktionen konkret in der jeweiligen Unterklasse implementiert, die Anwendung kann aber allgemein mit *Network*-Objekten arbeiten. Somit wird durch diese Anordnung der Vorteil des Polymorphismus sinnvoll ausgenutzt.

Ein weiteres zu lösendes Problem stellt der Aufruf der Empfangsfunktion in der *Network*-Klasse dar. Die physikalischen Schnittstellenrealisierungen sind auf eine rechtzeitige Abnahme der empfangenen CAN-Pakete angewiesen. Ist die Applikation zu langsam, so gehen durch den vollen Empfangspuffer des CAN-Bausteins die neu ankommenden Daten unwiderruflich verloren. Der Hardware-Interrupt der Karten bzw. des Dongles kann nur durch Implementierung eines virtuellen Gerätetreibers ausgenutzt werden. Durch die mitgelieferten Treiber der Hersteller wird sichergestellt, dass die eingehenden und zu sendenden CAN-Daten nicht verloren gehen. Sie verfügen dazu über Puffer, die in beiden Richtungen Daten zwischenspeichern. Die Datenpuffer müssen von der angebundnen Applikation rechtzeitig ausgelesen werden, bevor sie überlaufen.

Unter Windows kann eine zyklische Abfrage durch das Abfangen der WM\_TIMER-Nachricht realisiert werden. Diese Nachricht wird mit einer minimalen zeitlichen Verzögerung von ca. 10 ms (Windows NT 3.5 und später) bzw. mit ca. 55 ms (Windows 95/98) bereit gestellt. Dieser Mechanismus ist damit für das Auslesen der Nachrichtenpuffer durchaus ausreichend, da der PC im Wesentlichen nur darstellende und anzeigende Funktionen bzw. die Übermittlung von Bedienbefehlen übernimmt und keine „harte“ Echtzeitfähigkeit aufweisen muss. Alle relevanten Echtzeitabläufe werden auf Mikrocontroller-Ebene realisiert, so dass die geringe Wiederholrate des Timers und geringe zeitliche Verzögerungen keinen störenden Einfluss auf das System haben. Spezielle Echtzeitaufsätze sind daher nicht erforderlich, da der PC im wesentlichen nur Sollgrößen an die Module weiterleitet.

### 3.7.7 Realisierung der Systemklassen und Einbindung der GUI-Elemente

Ein Kernproblem im Steuerungssystem ist die Herstellung der Verbindungen der physikalischen Knoten (in *SystemStatus* abgespeichert) mit den GUI-Oberflächenelementen. Diese sind notwendig, um den Nachrichtenfluss im System zu ermöglichen. So sieht sich das System Dialogklassen gegenüber, die in separaten Quelltextdateien implementiert sind. Sie erzeugen dort die zugeordneten Oberflächenelemente (wie z. B. Buttons, Slider, Gauges usw.), wobei die Erzeugung direkt im Quelltext oder mit Hilfe einer zugeordneten Ressourcen-Datei geschieht.

Das Steuerungssystem muss dabei wissen, welche Benutzerdialoge verwendet werden und welche Bedienelemente in ihnen gespeichert sind. Letztendlich muss das System von der textuellen Beschreibung der Dialoge mit den darin enthaltenen Elementen ("Dialog x enthält Element y") zu einem Zeiger auf das im Dialogmodul erzeugte Element kommen, um damit arbeiten zu können. Zusätzlich muss für die Dialogelemente die Schnittstelle festgelegt werden, mit denen CAN-Nachrichten empfangen und gesendet werden können, um eine benutzerdefinierte Behandlung zu realisieren.

Im Folgenden werden die wesentlichen Lösungen dieser Probleme aufgezeigt.

Der Borland Compiler C++ 5.0 kapselt die Windows-API (Application Programmers Interface) in einer eigenen Klassenhierarchie, der OWL (Objekt Windows Library) /73/. Diese soll auf der einen Seite die Applikationserstellung vereinfachen, andererseits die einfache Portierung für andere Betriebssystemumgebungen ermöglichen.

Sie kapselt die unter Windows verfügbaren GUI-Bedienelemente in speziellen Klassen, die z. B. in Dialogen verwendet werden können. Im Steuerungssystem sollen diese Elemente ebenfalls verwendet werden können, wobei sie aber um spezielle Funktionen

(z. B. zur Nachrichtenverarbeitung) ergänzt werden. Daher werden von den OWL-Elementen spezielle Ableitungen gebildet, um diese dann entsprechend erweitern zu können. Damit bieten die Elemente in der Anwendung immer noch die bekannten OWL-Funktionen, der Applikationsprogrammierer braucht sich nicht auf völlig neue umzustellen. Nur die neu hinzugekommenen Funktionen des Steuerungssystems muss er neu erlernen.

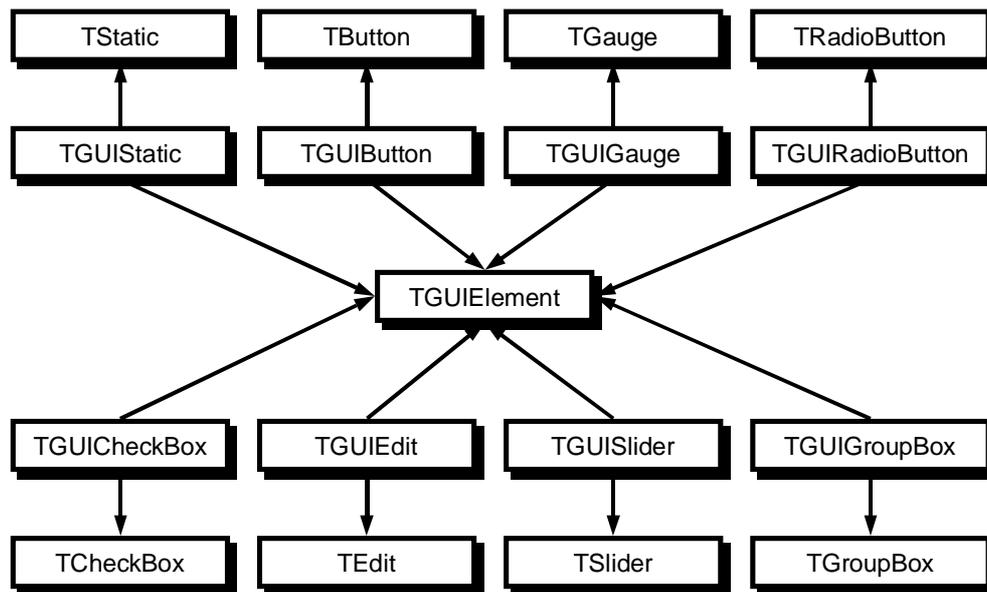


Abb. 37: Klassenhierarchie der TGUI-Elemente

Abb. 37 stellt die Klassenhierarchie der Elemente dar. Man erkennt, dass die Bedienelemente nicht nur von den OWL-Klassen erben, sondern auch eine gemeinsame Basis-Klasse *TGUIElement* besitzen. Dieses ermöglicht die einheitliche Behandlung der Steuerelemente in der Klasse *SystemStatus*. Die für das Steuerungssystem wichtigen Funktionen sind in der abstrakten Basisklasse *TGUIElement* vereint. Die Sende- und Empfangsfunktionen sind dabei rein virtuell und müssen von den abgeleiteten Klassen redefiniert werden. Diese Funktion ist in allen *TGUIxxx*-Klassen identisch implementiert.

Dabei kann ein Bedienelement mit der implementierten Sende-Methode Nachrichten nur dann an einen physikalischen Knoten senden, wenn dieser vorher registriert wurde. Das Statussystem führt diese Registrierung beim Erstellen der Systemkonfiguration durch, wobei alle als Eingabelemente in der textuellen Konfigurationsdatei bezeichneten Bedienelemente dadurch den zum Senden benötigten Zeiger auf den zugeordneten physikalischen Knoten erhalten. Diese können dann die von ihnen zusammengestellten CAN-Pakete an den zugeordneten Knoten senden.

*TGUIDialog* ist eine von der OWL-Klasse *TDialog* abgeleitete Klasse des Steuerungssystems. Sie kann in der gleichen Art wie diese Basisklasse zur Erstellung von Windowsapplikationen verwendet werden. Zusätzlich werden in ihr aber auch Verwaltungsinformationen gespeichert, die eine eindeutige Identifikation der im Dialog vorhandenen *TGUIElemente* ermöglicht. Diese werden benötigt, um später von der textuellen Beschreibung der im Dialog vorhandenen Bedienelemente zu den entsprechenden Zeigern zu kommen.

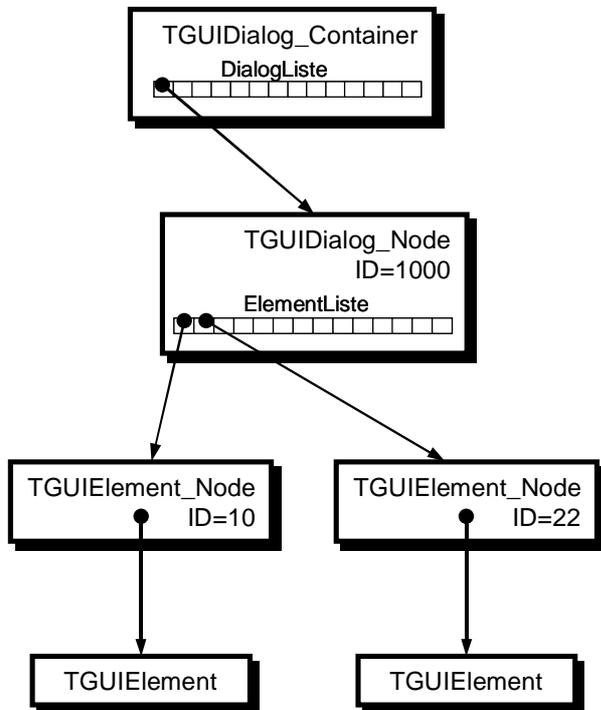


Abb. 38: Aufbau der Dialogstruktur

Um die Verwaltungsinformation abzuspeichern, existiert die Hilfsklasse *TGUIElement\_Node*, die als einziges Datenelement einen Zeiger auf ein *TGUI-Element* und die dazugehörige Integer-ID besitzt. *TGUIDialog* speichert die registrierten Elemente mittels einer durch die Borland-Containerklasse implementierten Liste, die als Knoten *TGUIElement\_Nodes* enthält (Abb. 38). Mit diesem Mechanismus können somit die Zeiger auf die registrierten *TGUIElemente* beschafft werden.

Die Zuordnung der Dialogbezeichner (Ressourcen-ID's), die in der Konfigurationsdatei textuell eingelesen werden, zu den wirklich vom Benutzer in den Objektdateien bereitgestellten *TGUIDialogen* erfolgt über die Klasse *TGUIDialog\_Container*, die alle im Steuerungssystem anzusprechenden *TGUIDialoge* mit ihren zugehörigen Ressourcen-

ID's verbindet.

In Abb. 39 ist ebenfalls die Datenverbindungen wiedergeben, die in *TGUIDialog\_Container* nach dem Anmelden der *TGUIDialoge* besteht.

Für die Registrierung aller in der Applikation verwendeten *TGUIDialoge* ist das Hauptprogramm verantwortlich. Deshalb müssen dort alle *TGUIDialoge*, deren Verwendung geplant ist, erzeugt und registriert werden.

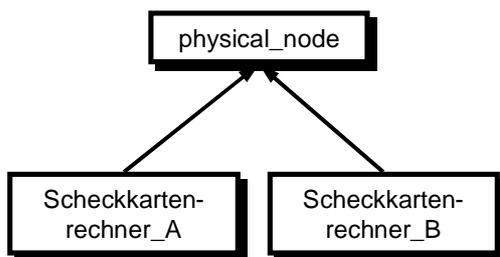


Abb. 39: Klasse für physikalische Knoten

Ähnlich zur Klasse *Network* erfolgt auch die Behandlung der unterschiedlichen physikalischen Knoten des Handhabungssystems. Momentan werden diese durch die beiden Versionen der Scheckkartenrechner realisiert, für die eigene Klassen im Steuerungssystem vorgesehen sind. Beide haben eine abstrakte Basisklasse namens *physical\_node* (Abb. 39), welche die gemeinsame Schnittstelle zu allen physikalischen Knoten festlegt. Hierbei

legt *physical\_node* die grundlegende Funktionalität der Sende- und Empfangsfunktionen rein virtuell fest, die von der ererbenden Klasse definiert werden müssen. Die Empfangsfunktion wird für die unterschiedlichen Knotenarten von der Empfangsroutine in *SystemStatus* aufgerufen. In dieser muss der physikalische Knoten (*physical\_node*) entscheiden, was mit der Nachricht geschehen soll, was immer von der konkreten Knotenart abhängt.

Die Sendefunktion wird hingegen direkt von den *TGUIElementen* aufgerufen, die von *SystemStatus* Scanvorgang als Eingabeelemente registriert wurden. Sie senden das von ihnen erzeugte CAN-Paket an den ihnen physikalisch zugeordneten Knoten.

Die Klasse *Scheckkartenrechner\_A* und *Scheckkarterechner\_B* implementieren ebenfalls die von *physical\_node* rein virtuell definierten Sende- und Empfangsfunktionen.

Die Klasse *SystemStatus* ist die Hauptklasse des Steuerungssystems, die alle nötigen Objektverbindungen gemäß der Konfigurationsbeschreibung vornimmt. Sie verwendet die Klasse *TGUIDialog\_Container* zur Verwaltung der Verknüpfungen zu den *TGUIDialogen* und den darin enthaltenen *TGUIElementen*. Die polling-Funktion wird vom implementierten Mechanismus zyklisch aufgerufen, worüber die Schnittstellentreiber zyklisch abgefragt werden.

Die Scanmethode ist zugleich die wichtigste Funktion in *SystemStatus*. Sie realisiert das Auslesen der Konfigurationsdatei und die Erstellung aller sich dadurch ergebenden Sende- und Empfangsverknüpfungen. Zusätzlich wird auch die interne Empfangsliste angelegt. Diese enthält Knoten der Hilfsklasse *Node*, die jeweils nur einen Zeiger auf einen physikalischen Knoten und die dazugehörige ID speichert. Wenn eine CAN-Nachricht von der Klasse *Netzwerk* an *SystemStatus* übergeben wird, so kann mittels dieser Liste der zugehörige physikalische Knoten mit der entsprechenden Empfangs-ID ermittelt werden, dem das Paket dann übergeben wird.

### 3.7.8 Oberflächenelemente der PC-Bedienoberfläche

Während die vorherigen Abschnitte auf den prinzipiellen Aufbau des PC-Programms eingegangen sind, wird nachfolgend das Programm zur Steuerung von TRIBUN gemäß

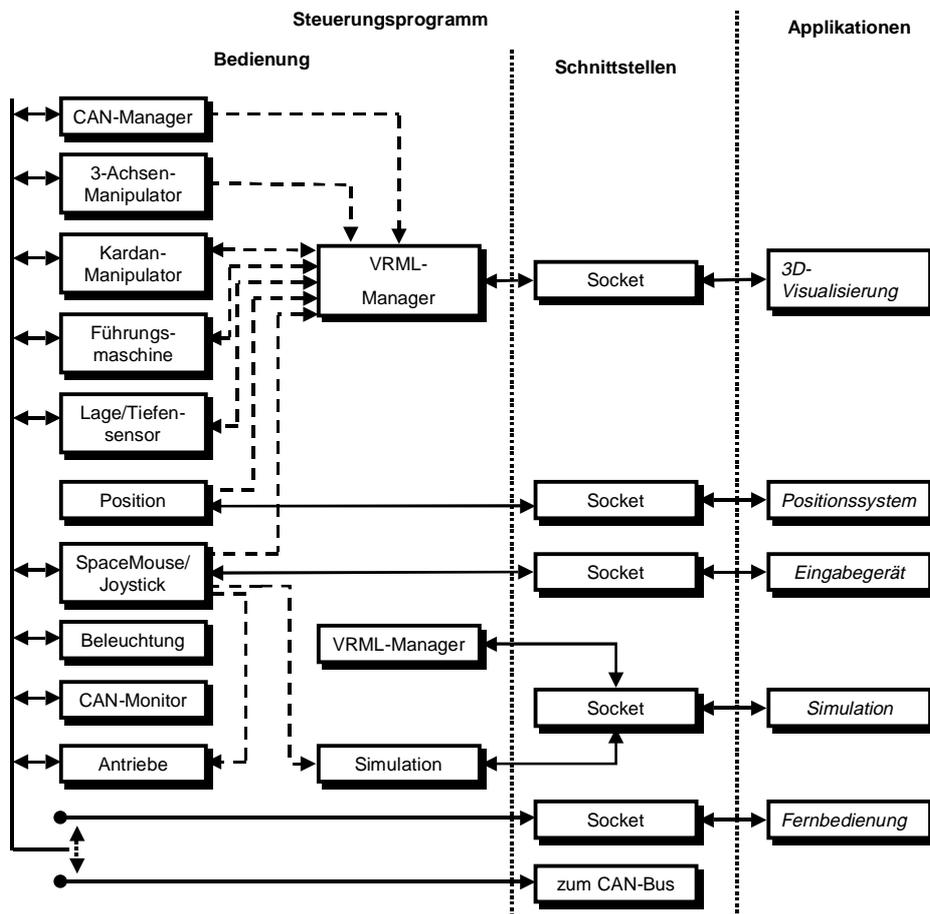


Abb. 40: Übersicht der Dialogboxen und der angebotenen Applikationen

der vorhandenen Komponenten sowie der gewünschten Funktionalität vorgestellt. Die Bedienoberfläche des Hauptprogramms wird als sog. MDI-Anwendung (Multiple Document Interface) ausgelegt. Dabei werden in einem Programmrahmen gleichzeitig mehrere Dialogboxen dargestellt, die für die Bedienung bereitstehen.

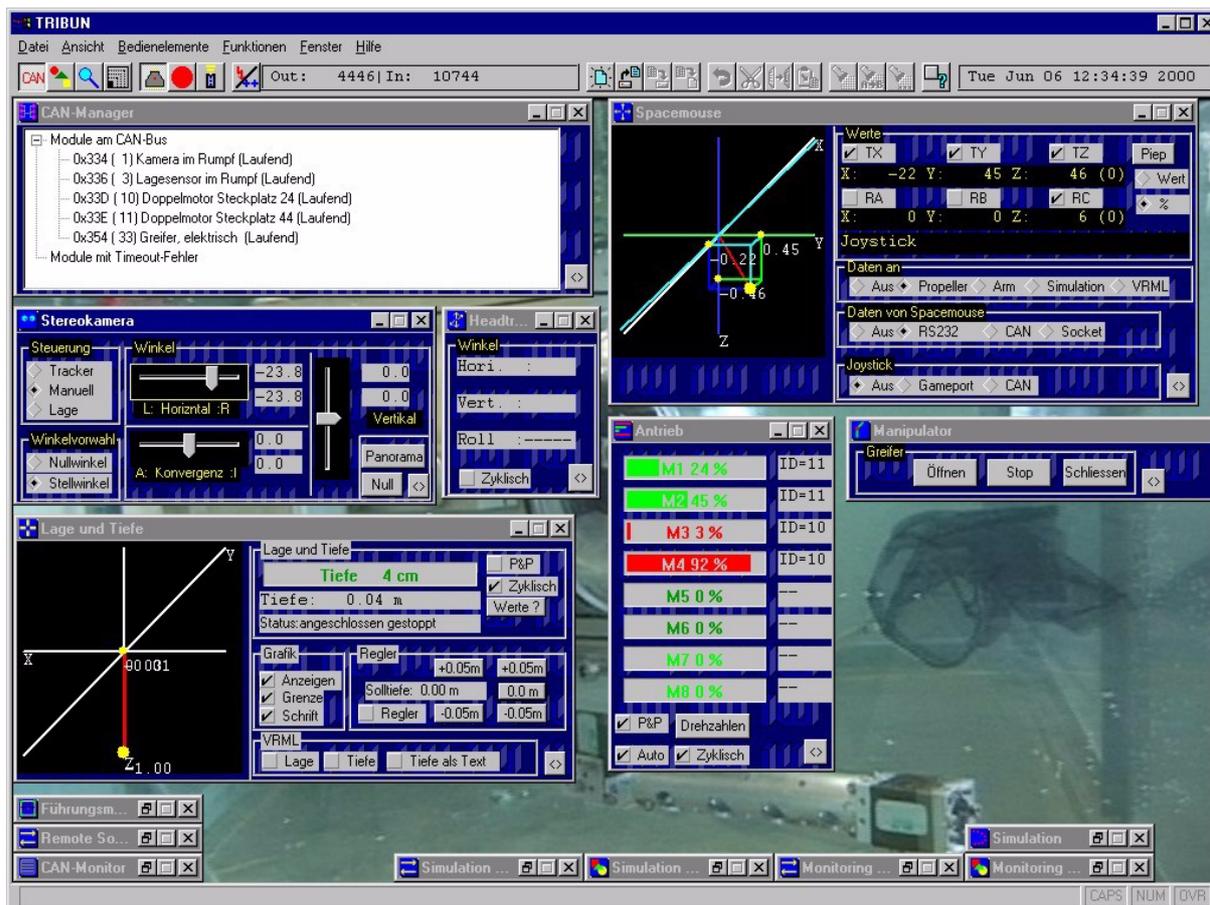


Abb. 41: Dialogboxen für CAN-Komponenten

Die erstellten Dialogboxen (Abb. 40) lassen sich in Boxen mit und ohne CAN-Funktionalität aufteilen. Grundsätzlich verfügt jedes existierende Modul über eine eigene Dialogbox (*TGUIDialog*), mit der es visualisiert und bedient werden kann. Implementiert wurden hier Dialoge für das Stereokamerasystem, den Headtracker, den Lagesensor, den Manipulator und Greifer, die Linearführungsmaschine sowie die Eingabemedien wie SpaceMouse oder Joysticks. Die Ausnahme bilden die Dialogboxen für das Propeller-Antriebssystem sowie die Beleuchtung, in der die vorhandenen Antriebs- bzw. Beleuchtungsmodule aus Gründen der Funktionalität gruppiert werden. Durch die implementierte Plug&Play-Funktionalität wird dabei, gemäß der von Bediener gewählten Einstellungen, automatisch bei Erkennung des Vorhandenseins eines Moduls die gewünschten Betriebsdaten oder ggf. auch Messdaten abgefragt und angezeigt.

Übergeordnet stehen zwei Dialogboxen für die Analyse und die Entwicklung bereit. Dabei handelt es sich um den CAN-Manager, der als Anzeige aller angeschlossenen CAN-Komponenten sowie deren aktuellen Betriebszustand fungiert sowie den CAN-Monitor. Über ihn können alle angeschlossenen Komponenten analysiert und in Betrieb genommen werden. Dies sind z. B. Funktionen zum Download der Software in CAN-Module oder Funktionen für das Monitoring der CAN-Kommunikation und der Speicherbereiche. Für die 3D-Visualisierung (Kap.3.10) steht zur Verwaltung der Fahrzeugkonfi-

guration, der Arbeitsumgebung sowie der Sensorenkonfiguration ein Dialogobjekt bereit, das die Erstellung von beliebigen Arbeitsumgebungen gestattet, über Funktionen zum Speichern und Laden der Konfigurationen verfügt und die Erstellung einfacher Animationen erlaubt.

Für die Netzwerkkommunikation stehen Dialogelemente mit Sokketanbindung zur Verfügung. Über sie kann beim Netzwerkbetrieb von TRIBUN, z. B. die Anbindung von externer Sensorik erfolgen, die 3D-Visualisierung angebunden werden oder auch eine Fernbedienung des Fahrzeug über ein lokales Netzwerk erfolgen (Kap.3.8). Weiterhin gestatten sie die Unterstützung bei Verbundbetrieb, wobei mehrere Bediener an unterschiedlichen Rechner eine Bedienung durchführen können.

Die einzelnen Dialogboxen sind in logische Gruppen für verschiedene Betriebsarten wie Analyse und Inbetriebnahme, Normalbetrieb oder auch Kommunikationsanalyse unterteilt. Gemäß der Auswahl des Bedieners werden alle zugehörigen Dialogboxen einer Gruppe angezeigt.

Zur Anbindung, Steuerung und Kontrolle der angeschlossenen MMIs dient dabei eine weitere Dialogbox.

### 3.7.9 Handbediengerät

Gerade wenn Arbeiten rein manuell ausgeführt werden sollen und auf die 3D-Visualisierung verzichtet werden kann, dient der PC im wesentlichen nur der Anbindung der Eingabegeräte und der Ansteuerung der Motoren oder auch eines Greifers. Da ein PC eine höhere Anfälligkeit gegen Beschädigung (Festplatte) aufweist, wurde speziell eine Bedienkonsole entwickelt, über die das Fahrzeug alternativ gesteuert werden kann.



Abb. 42: Handbediengerät mit Joysticks und SpaceMouse

Diese verfügt über eine serielle Schnittstelle zur Anbindung der SpaceMouse und zwei analoge Joystickports (PC-Anschluss kompatibel) für den Anschluss von maximal vier 2-Achsen-Joysticks. Die Analog-Joysticks können zur Bedienung der Werkzeuge, z. B. von Greifern oder der Zweiachsenführungsmaschine verwendet werden. Zusätzlich verfügt es über einige LEDs zur Statusanzeige sowie noch frei verfügbare Schaltfunktionen. Alle Steuersignale

der angeschlossenen Eingabegeräte werden von einem Mikrocontroller erfasst, verarbeitet und über den CAN-Bus an die entsprechenden Module weitergeleitet. Die Konsole kann durch deaktivieren der Steuerungsfunktionen aber auch nur zur reinen Anbindung der Eingabegeräte an den CAN-Bus bzw. an den PC dienen.

## 3.8 Netzwerk-Funktionalität

### 3.8.1 Schnittstellen und Buskoppler für den Netzwerkbetrieb

Die Vernetzung von Rechnern ist heute Stand der Technik. Die Möglichkeiten, die sich dadurch für die Handhabungstechnik bieten, sind vielfältig und reichen von der Verteilung von Steuerung und Visualisierung auf verschiedene Rechner bis hin zum kooperativem Arbeiten und Bedienen im Verbund. Die entwickelte Software zur Steuerung von TRIBUN unterstützt Fernbedienfunktionen, die eine Trennung von Leitstand und Leit-rechner ermöglichen und somit eine Bedienung auch über ein lokales Netzwerk erfolgen kann.

Die Trennung von Leit-rechner und Leitstand macht dabei einige Erweiterungen im Steuerungsprogramm notwendig. Idealerweise wird für den Netzwerkbetrieb keine neue Bediensoftware eingesetzt, sondern der Bediener kann mit der ihm vertrauten Bedienoberfläche arbeiten. Damit das Steuerungsprogramm über das LAN kommunizieren kann, wird eine zusätzliche Socketverbindung bereitgestellt. Die CAN-Daten, die sonst direkt auf den CAN-Bus gelegt werden, werden mittels aufgesetzter Protokolle zum Rechner am Leitstand übertragen. In umgekehrter Weise müssen die vom CAN-Bus kommenden Daten an das Bedienprogramm weitergeleitet werden. Zusätzlich müssen alle Statusinformation wie z. B. Fehlermeldungen ebenfalls übertragen werden. Weiterhin besteht die Notwendigkeit auch die Videodaten zu übertragen, um das Fahrzeug manuell führen zu können.

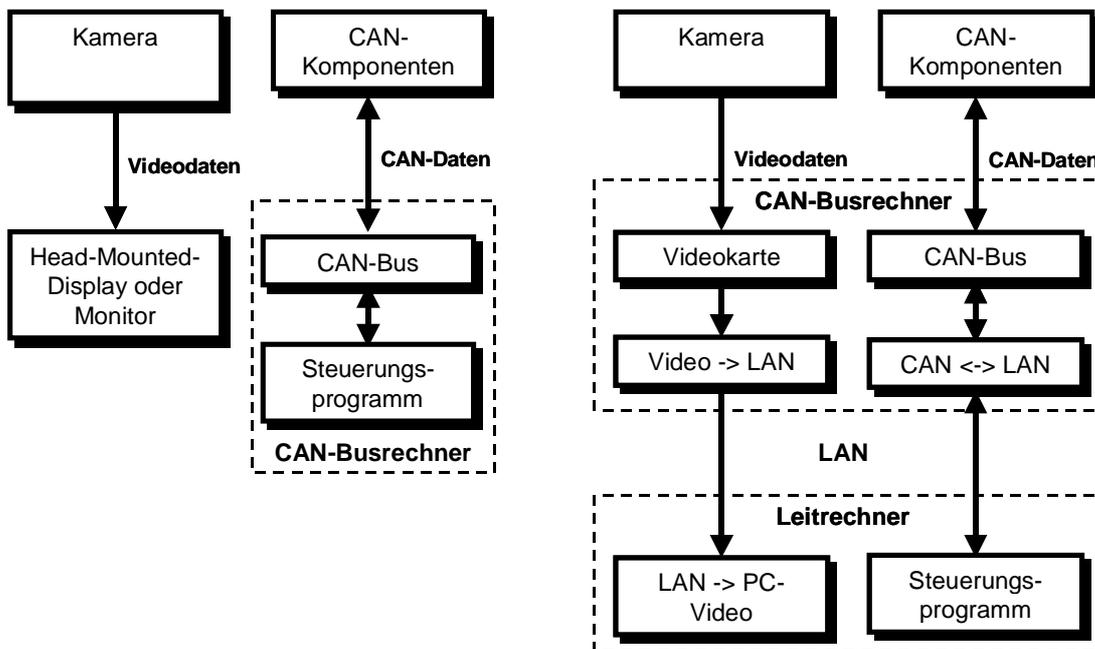


Abb. 43: Leitstandvarianten im Normalbetrieb (l.) und Netzwerk-Betrieb (r.)

In Abb. 43 ist das Prinzip der implementierten Datenübertragung und der Kopplungsmechanismen für den Normalbetrieb und den LAN-Betrieb dargestellt.

Die Aufgabe des Buskopplers auf dem Leit-rechner übernimmt ein speziell für diese Aufgabe im Rahmen der Arbeit entwickeltes Programm, das im Folgenden als *CANSock* bezeichnet wird. Das aufgesetzte Protokoll gestattet es CAN-Daten zum CAN-Netz bzw.

zur angebundenen Applikation zu übertragen. Weiterhin können Zusatzinformationen wie Fehlermeldungen abgefragt oder Voreinstellungen getätigt werden.

Das Programm *CANSock* ist so ausgelegt, dass entweder alle eingehenden CAN-Daten übertragen werden oder, um die Netzbelastung gering zu halten, zusätzliche Filterfunktionen konfiguriert und aktiviert werden können. Dies ermöglicht es, nur spezielle Nachrichten zu übertragen, da z. B. die Kommunikation zwischen zwei Teilnehmern uninteressant sein kann, spezielle Messdaten der CAN-Module aber genutzt werden sollen.

Als weitere Applikation wurde *COMSock* entwickelt, die ebenfalls eine Socket-basierte Schnittstelle zu externer Hardware bereitstellt. *COMSock* ist analog zu *CANSock* ausgelegt, nur dass eine Socket-Schnittstelle zu einem PC mit serieller RS-232-Schnittstelle (COM-Schnittstelle) bereitgestellt wird. Da eine Vielzahl am Markt erhältlicher Sensoren/Aktoren über diese Schnittstelle verfügen, ist eine große Einsatzbandbreite gegeben.

Die beiden Programme sind auch über gängige Netzwerk-Tools bedienbar, ohne dass eigene Applikationen programmiert werden müssen. Über ein Terminalprogramm (z. B. Telnet) können damit Steuer- und Abfragefunktionen ausgeführt werden. Mit Hilfe eines Web-Browsers können ebenfalls Hardware-Zugriffe und Abfragen realisiert werden. Damit ist es auch möglich, Module über das Netzwerk oder auch über das Internet zu parametrisieren. Die benötigte Funktionalität wird in eine Webseite z. B. über Formulare, Buttons und JavaScript-Funktionen integriert, die dann ein Versenden der Daten die Applikation sowie eine Auswertung erlauben /97/,/98/.

Weitere Einzelheiten sowie die konkrete Einsatzszenarien der Programme werden in Kap.4.5 erläutert.

### **3.8.2 Videodatenübertragung für den Netzbetrieb**

Wenn das Fahrzeug über das LAN fernbedient werden soll, muss zumindest ein Videobild der Bordkamera oder auch einer externen Kamera, die das Fahrzeug im Arbeitsraum zeigt, ebenfalls über das Netzwerk übertragen werden. Dafür muss der PC mit einer zusätzlichen Videoeinsteckkarte ausgestattet werden. Soll nur ein externes Bild übertragen werden, genügt auch eine einfache WebCam, die z. B. an die Drucker-schnittstelle oder über eine USB-Schnittstelle angebunden wird.

Für die Bildübertragung stehen mittlerweile eine Vielzahl von Applikationen bereit, so dass auf eine kosten- und zeitintensive Eigenentwicklung verzichtet werden kann. Obwohl die Programme im Wesentlichen für die Internet-Bildtelefonie entwickelt wurden, lassen sie sich auch hervorragend für die Beobachtung von Handhabungsaufgaben einsetzen. Die Programme schalten sich auf die verfügbaren Videoquellen des PCs auf und führen eine Komprimierung der Videodaten zwecks Reduzierung der Übertragungsbandbreite durch. Danach werden die Bilder übertragen und über dieselbe Software dem Bediener zur Verfügung gestellt. Eine Bewertung der verwendeten Software erfolgt in Kap.4.5.

### **3.8.3 Anbindung an die Simulation und das Positionssystem**

Die Grundlage für die Simulation bilden neben der notwendigen Software, die das Bewegungsverhalten auf der Basis der mathematischen Modelle simuliert, vor allem die

Bereitstellung der Parameter zur Beschreibung des Bewegungsverhaltens. Diese Parameter sind u. a. die Geräte- und Modulmassen, die Massenträgheitsmomente, die Strömungswiderstände sowie die durch die Propeller aufgebrachtten Kräfte. Ziel ist es, das Bewegungsverhalten des Tauchroboters in den unterschiedlichsten Konfigurationen simulieren zu können, um damit optimierte Regler für Lage und Position zu entwerfen sowie ein Bedienertraining durchführen zu können. Die Simulation von modularen Unterwasserfahrzeugen wurde bereits von Reinhardt vorgestellt /15/. An dieser Stelle soll daher nur auf die Anbindung eingegangen werden.

Das Steuerungsprogramm von TRIBUN wurde mit einer Socket-Anbindung an die Simulation ausgestattet. Die Steuerdaten der SpaceMouse können damit direkt als Eingabegrößen für die Simulation dienen. In umgekehrter Datenflussrichtung werden die Positions- und Lagedaten zurückgeliefert und an die 3D-Visualisierung weitergeleitet.

Die Grundlage bildet die Arbeit /56/, auf deren Basis momentan eine Weiterentwicklung erfolgt. Die Positions- und Lageinformation werden direkt an die 3D-Visualisierung zu Monitoringzwecken weitergeleitet oder können den übergeordneten Reglern zugeführt werden. Die Anbindung des Positionssystems erfolgt ebenfalls über eine Socketverbindung.

### 3.9 Sicherheitsaspekte

Ein wichtiger Punkt bei der Auslegung des Gesamtsystems ist eine hohe Funktionssicherheit und die gesicherte Rückholbarkeit des Fahrzeugs während eines Einsatzes. Dabei kommen sowohl aktive als auch passive, konstruktionsbedingte sowie steuerungstechnische Maßnahmen zum Einsatz.

Als passive Sicherheitsmaßnahme verfügt das Fahrzeug über einen geringen Restauftrieb, der bei Ausfall der Steuerung dafür sorgt, dass das Fahrzeug zur Oberfläche zurückkehrt, von wo eine weitere Bergung deutlich erleichtert wird. Das Fahrzeug ist konstruktionsbedingt durch einfaches Abwaschen der Oberfläche leicht zu dekontaminieren. Während des Betriebes kommt kein Wasser in den Rumpf, da auf den Einsatz z. B. von Tauchtanks und Pumpen verzichtet wurde.

Eine Ausnahme bildet hierbei die 2-Achsen-Führungsmaschine, die mit einer Pumpe zur Erzeugung des Unterdrucks in den Sauggreifern ausgestattet ist und nach einem Einsatz ggf. dekontaminiert werden muss.

Als aktive Maßnahme kann das Fahrzeug in Abhängigkeit von der Tauchtiefe mit einem definierten Überdruck beaufschlagt werden, um das Eindringen von Wasser zu verhindern. Die Konstruktion des Rumpfs sowie der Multifunktionsleisten ist dabei so ausgelegt, dass sich die Druckluft sowohl im gesamten Druckkörper, in jedem Steckplatz und über den Steckplatz auch in allen ankoppelten Modulen verteilen kann.

Grundsätzlich gibt es keine Alternative zu einer Bergeleine, da immer eine Restfehlerwahrscheinlichkeit eines totalen Systemversagens besteht. Das Versorgungskabel wird für hohe mechanische Belastungen ausgelegt, um bei einem Systemausfall die Bergung des Fahrzeugs zu gewährleisten.

Alle Module von TRIBUN verfügen über Überwachungsfunktionen der angeschlossenen Aktorik und Sensorik und melden Ausfälle. Bei Ausfall der steuernden Einheit erfolgt zeitgesteuert die Ausführung von Notfallmaßnahmen gemäß Tabelle 10.

Tabelle 10 : Sicherheitsfunktionen der Module

Modul/Einheit	Sicherheitsfunktion
Trackermodul	Der Ausfall des Headtracker oder der Kameraeinheit führt dazu, dass keine weiteren Steuerbefehle an die Kameraeinheit gesendet werden.
Kamerasteuerung	Empfängt die Kameraeinheit keine Steuerbefehle, fährt sie die Achsen in eine voreingestellte Nullposition, um eine akzeptable Sicht zu garantieren. Der Ausfall einer Kamera führt zwar zum Verlust der Stereofähigkeit, gestattet aber trotzdem mit der intakten Kamera das Fahrzeug zu navigieren.
Lagesensor	Der Ausfall des Lage- bzw. Tiefensensors führt zur automatischen Abschaltung der übergeordneten Lage- oder Tiefenregelung.
Propellereinheit	Erhält eine Motoreinheit während einer voreingestellten Zeitspanne keine Steuerbefehle, werden die vorhandenen Motoren gestoppt, um eine unkontrollierte Bewegung des Fahrzeugs zu verhindern.
Steuernde Einheit (PC oder Handbediengerät)	Detektiert die Steuerung einen Ausfall eines Motor wird die Ansteuerung des Moduls nach erneuter Prüfung unterbunden.
Führungsmaschine	Bei Ausfall der steuernden Einheit werden ebenfalls nach einer voreingestellten Zeit alle Achsbewegungen gestoppt. Sind die Sauggreifer auf „Halten“ geschaltet wird die Haltefunktion auch weiterhin geregelt und überwacht. In allen übrigen Betriebszuständen werden die Pumpe zur Unterdruckerzeugung abgeschaltet und der Unterdruckkreislauf geschlossen.

Auf die kostenintensive Entwicklung strahlenharter Elektronikkomponenten wird generell zugunsten leicht austauschbarer konventioneller Baugruppen mit verkürzter Lebensdauer verzichtet.

Das verwendete CAN-Bussystem weist standardmäßig durch die implementierte Fehlererkennung und Fehlerkorrektur eine hohe Störsicherheit auf.

### 3.10 System für die 3D-Visualisierung unter VRML 97

Die Durchführung von komplexen Handhabungsaufgaben stellt hohe Anforderungen an den Bediener, insbesondere wenn die Arbeit rein manuell ausgeführt wird und das System über viele Freiheitsgrade verfügt. Videobilder, die das Fahrzeug im Arbeitsraum zeigen, liegen im Allgemeinen nicht oder nur mit Einschränkungen vor, so dass sich der Bediener oft nur allein mit Hilfe der Bordkamera orientieren muss. Das Unterwasserfahrzeug TRIBUN wird daher im Rahmen dieser Arbeit mit einer 3D-Visualisierung ausgestattet. Sie soll den Bediener während der Durchführung der Arbeiten unterstützen und weiterhin im Zusammenspiel mit einer Simulation des Bewegungsverhaltens des Fahrzeuges im Vorfeld ermöglichen ein Bedienertraining durchführen zu können. Der modulare Charakter des Systems muss dabei auch von der Visualisierungssoftware unterstützt werden.

Grundsätzlich können für die Visualisierung entweder am Markt verfügbare Programme oder Eigenentwicklungen verwendet werden. Auf Grund des hohen Verbreitungsgrades von PCs werden Möglichkeiten für einen kostengünstigen Einsatz gerade auf PC-Basis untersucht.

#### 3.10.1 Vorbetrachtungen zur 3D-Visualisierung

Ein breites Anwendungsgebiet der 3D-Visualisierung ist neben der 3D-Konstruktion die Robotersimulation, die als modernes Hilfsmittel eingesetzt wird, um die Planung und die Programmierung von Roboterzellen rechnergestützt ausführen zu können. Am Markt sind mittlerweile verschiedene DMP-Systeme (Digital manufacturing and production) wie ROBCAD oder ROBSIM verfügbar. Die Programmierung, der Test und die Optimierung erfolgen zunächst offline unter Berücksichtigung von Kollisionsbetrachtungen und Kollisionsvermeidung. Die erstellten Programme können danach direkt in die Maschinensteuerung geladen und ausgeführt werden. Obwohl ganze Produktionszellen modelliert und in Realzeit simuliert und animiert werden können und eine Schnittstelle für Up- und Download für die Roboterprogramme existiert, sind sie für einen Monitoring-Betrieb oft nicht geeignet, da den Programmen eine äquivalente Schnittstelle zum Auslesen z. B. der Achsstellungen oder weiterer Maschinendaten fehlt /74/,/75/,/76/,/77/.

Speziell für die Planung von automatisiert durchzuführenden Unterwasserhandhabungsaufgaben wurde bei der GKSS-Forschungszentrum Geesthacht GmbH ein Softwaresystem mit der Bezeichnung PUSSY für die Offline-Programmierung und grafische Handhabungssimulation entwickelt und erprobt. Es gestattet verschiedenartige Roboterkinematiken oder auch Unterwasserträgerfahrzeuge zu modellieren und deren Einsatz in Echtzeit zu simulieren. Damit können im Vorfeld eines geplanten Einsatzes Szenarien durchgespielt und optimiert werden. Ein Schwerpunkt des Programms ist die Kollisionserkennung und -vermeidung. Da es ebenfalls über keine Programmierschnittstelle verfügt, ist es allerdings für einen Monitoringbetrieb ungeeignet /78/.

Für die Visualisierung von Handhabungssystemen wird seit 1986 am Forschungszentrum Karlsruhe das Programm KISMET (**K**inematic **S**imulation, **M**onitoring and **O**ff-Line **P**rogramming **E**nvironments for **T**elerobotics) entwickelt /79/. KISMET ist ein 3D-Simulationsprogramm für die effektive Planung, Simulation, Programmierung und Monitoring von Fernhandhabungsaufgaben sowie Robotersimulationen. Die leistungsfähige grafische Visualisierung findet dabei sowohl bei medizinischen, ingenieurwissenschaftlichen

und bei produktionstechnischen Aufgaben Verwendung. Das Programm gestattet die Modellierung komplexer Roboterkinematiken und Arbeitsräume sowie die Animation von Objekten durch die Erstellung von Ablaufprogrammen. Zur Animation können die Positions- und Achsdaten über eine unidirektionale Socketschnittstelle zugeführt werden, die speziell für den Monitoring-Einsatz bereit steht. Ein Vorteil von KISMET ist die Stereofähigkeit der Software, d. h. dass dem Bediener durch das Tragen einer Shutter-Brille ein räumlicher Eindruck vermittelt werden kann. Nachteilig ist, dass das Programm zu Beginn der Arbeit ausschließlich auf Rechnern der Firma Silicon Graphics (SGI) lauffähig war und erst seit Dezember 1999 auch für Windows NT verfügbar ist. Zusätzlich sind relativ hohe Lizenzgebühren zu entrichten.

OpenGL ist ein weit verbreiteter Grafikstandard, der von der Firma SGI entwickelt wurde und über Bibliotheken an eigene Programme angebunden werden kann /80/. Die so erstellten Programme zeichnen sich durch hohe Ausführungsgeschwindigkeiten sowie individuelle Gestaltungsmöglichkeiten aus. Nachteilig ist der extrem hohe Programmieraufwand einer solchen Eigenentwicklung und das Fehlen einer standardisierten Beschreibungssprache für Geometrien. Auf den Einsatz einer solchen Lösung wurde daher verzichtet.

### 3.10.2 Grundlagen und Einsatzgebiete von VRML 97

Eine weitere Möglichkeit stellt die Kombination aus Kauflösung und Eigenentwicklung dar. Im Laufe der letzten Jahre wurde durch die schnelle Entwicklung des Internets sowie von Internet-basierter Technologie interessante Möglichkeiten geschaffen, um virtuelle Welten zu beschreiben und darzustellen. Der VRML-Standard (Virtual Reality Modelling Language) ist eine dieser Entwicklungen, der mittlerweile einen hohen Verbreitungsgrad bei großer Einsatzbandbreite aufweist und dem Anwender zusätzlich kostenlos zur Verfügung steht /81/.

Die Sprache VRML dient zur Beschreibung dreidimensionaler Welten. Die Visualisierung dieser Welten erfolgt durch einen sogenannten VRML-Browser, der die Beschreibungen interpretiert und die darzustellende Welt aus der Sicht des Betrachters berechnet. VRML-Browser existieren sowohl in Form eigenständiger Programme als auch als Plugin zu einem WWW-Browser. Als Plugin bezeichnet man ein Hilfsprogramm, das ein Programm (in diesem Fall den WWW-Browser) ergänzt, indem es über eine definierte Schnittstelle neue Funktionen zur Verfügung stellt. In diesem Falle werden die VRML-Welten innerhalb einer HTML-Seite dargestellt.

Die Sprache VRML wurde in der Version 1.0 im Jahre 1995 erstmals spezifiziert, mit dem Zweck, eine Beschreibungssprache für dreidimensionale Inhalte im WWW bereitzustellen /82/. Sie wurde von der textbasierten Beschreibungssprache des Open Inventor abgeleitet und dem Verwendungszweck angepasst /83/. Die Version 1.0 umfasste nur die Beschreibung statischer virtueller Welten. Darauf basierend wurde die Version 2.0 mit dem Ziel entwickelt, dynamische Welten darzustellen, die mit dem Betrachter interagieren können. Zusätzlich wurde die Möglichkeit vorgesehen, multimediale Inhalte wie Geräusche oder Videos in die Welt einzubinden. Zur Standardisierung der Sprache VRML durch die ISO wurde die Spezifikation redaktionell überarbeitet, jedoch ohne funktionelle Unterschiede unter dem Namen VRML 97 veröffentlicht /84/. Seit dieser Standardisierung wurde noch eine externe Schnittstelle zu Programmen in

der Sprache Java entwickelt, die als External Authoring Interface (EAI) bekannt ist /85/,/86/.

Obwohl VRML noch eine recht junge Sprache ist, hat sie sich mittlerweile in vielen Bereichen etabliert.

So beschreibt Beier eine VRML-basierte Darstellung bei der Konstruktion und Simulation von Schiffen, Einsätze im Bereich des Designs eines Flughafenterminals und die Visualisierung von Robotern im Produktionsbereich der Automobiltechnik /87/. In /88/ wird ein VR-Schulungssystem zur Durchführung von Wartungs- und Inspektionsarbeiten an Leistungstransformatoren vorgestellt, das ebenfalls den VRML-Browser zur Darstellung der Modelle für Arbeitsumgebung und Geräten verwendet.

Unbemannte Luftfahrzeuge werden häufig von einer Leitstelle am Boden gesteuert. Dabei tritt das Problem auf, dass der Bediener weder dasselbe Gefühl wie ein Pilot an Bord eines Luftfahrzeuges hat noch dass er über einen direkten Blickkontakt verfügt. Abhilfe schafft hier ein Simulationssystem, das in Echtzeit ein Bedienertraining für die Fernbedienung gestattet /89/.

In /90/ wird die Visualisierung von mobilen Robotern zu Überwachungs- und Testzwecken beschrieben. Dabei kann der Roboter über eine Java-Applikation ebenfalls über das Netz gesteuert werden. Ein ähnlich arbeitendes System beschreiben auch Colon und Baudoin /91/. Dabei werden 3D-Modelle eingesetzt, um auf eine optische Kontrolle mittels einer Videoübertragung über das Internet verzichten zu können, wenn die Qualität der Bilder oder die Geschwindigkeit der Datenübertragung unzureichend ist.

Zur Überwachung von Produktionsprozessen zeigen Lehmann und Richter auf, wie prozessfähige VR-Modelle für die Anlagenplanung und die Darstellung von logistischen Prozessabläufen eingesetzt werden können /92/.

Ein weites Einsatzgebiet sind Darstellungen im Bereich der Architektur. So wird in /93/ eine Kombination aus VRML-Browser und Modellierungsprogramm vorgestellt, die die Konstruktion von virtuellen Gebäuden aus Einzelkomponenten erlaubt, die in VRML-Datenbanken abgelegt sind. Der Anwender kann sich innerhalb der Gebäude bewegen und kann interaktive Objekte näher untersuchen.

VRML weist allerdings auch einige Schwachstellen auf, wie z. B. das Fehlen einer Objekt-Objekt-Kollisionserkennung oder fehlende Programmierrichtlinien bei der Namensvergabe für Prototypen und Knotennamen, wodurch eine konfliktfreie Kombination von unabhängig voneinander erzeugten Modellen in der Regel nicht möglich ist. Auch eine interaktive Veränderung von Szenen, die über das reine Bewegen von Objekten hinaus geht, ist nur eingeschränkt oder mit großem Aufwand realisierbar. Trotzdem ist der Standard für die Darstellung von statischen und einfachen dynamischen Szenen hinreichend mächtig /94/.

Alle Lösungen haben gemeinsam, dass, wenn es über die reine Darstellung der Objekte hinausgeht, eine Datenverbindung zum VRML-Browser mittels des EAI und einer Java-Netzwerkverbindung (Applet mit Socketanbindung) realisiert werden muss, um von externen Applikationen mit den virtuellen Welt interagieren zu können /95/. Nachfolgend wird die im Rahmen dieser Arbeit VRML-basierte Visualisierung für den Einsatz speziell im Bereich der Handhabungstechnik vorgestellt.

Die gesamte Funktionalität von VRML ist relativ komplex und soll hier nicht näher behandelt werden. Für weitergehende Informationen sei auf die Spezifikation oder wei-

terführende Literatur verwiesen /81/,/82/,/84/,/85/. In den folgenden Abschnitten sollen nur auf einige grundlegende Konzepte der Sprache VRML eingegangen werden. Der Schwerpunkt liegt hierbei auf den zum Verständnis der Arbeit notwendigen Aspekten.

Eine VRML-Welt wird durch ihren Szenengraphen in Form einer hierarchische Baumstruktur beschrieben. Der Graph besteht aus sogenannten Knoten. In der Spezifikation sind eine Vielzahl von Knotentypen festgelegt, die jeweils ein Objekt der realen Welt abstrahieren (z. B. Form oder Oberfläche eines Gegenstandes) oder zum strukturellen Aufbau der virtuellen Welt dienen (z. B. Knoten zur Gruppierung verschiedener Objekte). Ein Knoten enthält Felder zur Beschreibung seines Zustandes und sogenannte Events (Ereignisse) als Schnittstelle zwischen dem Knoten und seiner Umgebung.

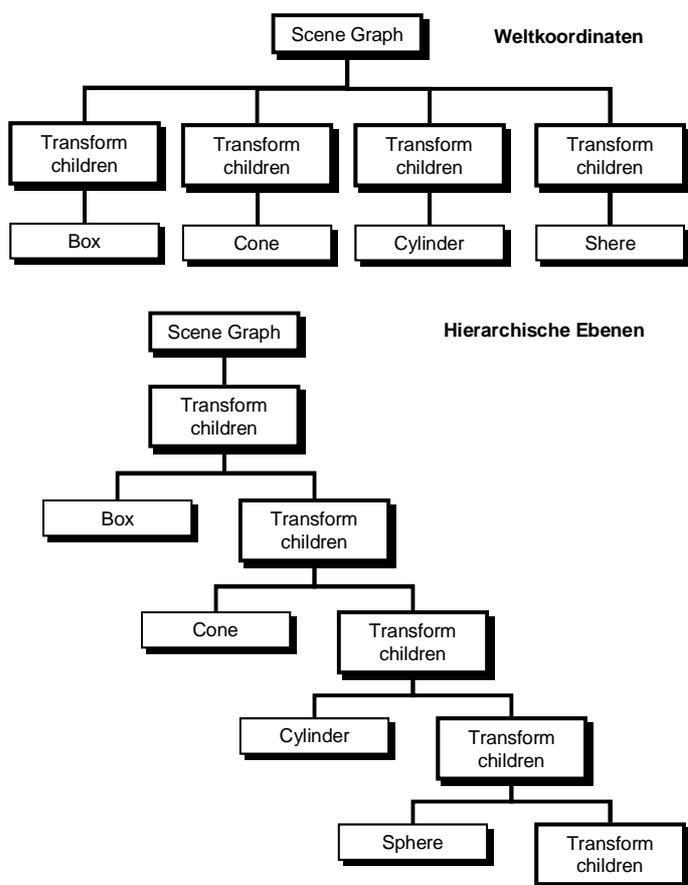


Abb. 44: Scenegrphen in globalen Koordinaten (o.) und hierarchischen Ebenen (u.)

Um Objekte zu modellieren, werden Gruppen aus graphischen Primitiven wie Quader, Kugel, Kegel, Zylinder oder auch Drahtgitterelementen zusammengesetzt.

Die Aufgabe der Echtzeit-Visualisierung ist es, einzelne Geometriegruppen innerhalb des Szenegraphen kontinuierlich in der virtuellen Umgebung gemäß den Positions- und Lageinformationen zu aktualisieren.

Dabei wird bei vielen vorgestellten Lösungen zwecks Reduzierung des Programmieraufwandes häufig ein Szene-Graph modelliert, der in der obersten Ebene alle benötigten geometrischen Objekte (Arbeitsumgebung, Fahrzeuge, etc.), die animiert werden sollen, enthält /15/. Der Nachteil ist, dass aktuelle Änderungen z. B. in der Fahrzeugkonfiguration nur ungenügend berücksichtigt werden, da sie entweder von Anfang an in der Beschreibung vorliegen oder zur Laufzeit generiert werden müssen.

Natürlich ist es möglich, komplexe Objekte aus einzelnen Teilobjekten zusammenzusetzen, aber eine Bewegung der Einzelobjekte erfolgt dabei immer zeitlich getrennt. Dies kann je nach Rechenaufwand zu extrem störenden optischen Effekten führen, da sich das komplexe Objekt während der Bewegung aufzulösen scheint. Sind Translationen dabei vom Rechenaufwand noch relativ gering, so muss bei der Rotation eines zusammengesetzten Objektes um eine Achse jedes Einzelobjekt in Lage und Position neu berechnet werden. Typischerweise verwendet man aber für zusammengesetzte Objekte eine hierarchische Struktur (Abb. 44 unten). Dies hat den Vorteil, dass z. B. bei einer

translatorischen Bewegung des obersten Knotens ebenfalls alle untergeordneten Knoten verschoben werden.

Setzt man die virtuelle Welt dynamisch aus hierarchisch aufgebauten Teilwelten zusammen, so ergibt sich jedoch die Problematik, dass man nachträglich nach dem Laden des Objektes nicht beliebig auf die untergeordneten Knoten zugreifen kann. Bei unveränderlichen Geometrien, wie der Arbeitsumgebung, ist dies nicht weiter von Bedeutung, stellt aber z. B. beim Hinzufügen von Funktionsmodulen an den Grundkörper von TRIBUN, ein Problem dar. Daher werden weitere Mechanismen vom VRML für die dynamische Modellierung von TRIBUN genutzt, wodurch allerdings eine teilweise Spezialisierung auf dieses Handhabungssystem der Software auftritt. Die Visualisierungssoftware setzt sich daher aus einem allgemeinen Bereich und einer speziellen Handhabungssystem spezifischen Lösung zusammen.

### 3.10.3 Programmier- und Kommunikationsschnittstellen

Das ExternalAuthoringInterface (EAI) ist eine Schnittstelle zur Anbindung externer Software wie z. B. JavaApplets oder auch Javascript-Funktionen an den VRML-Browser, bzw. der durch ihn dargestellten VRML-Welt. Über sie erhalten Anwender die Möglichkeit, die Funktionalität des VRML-Browser nach eigenen Wünschen zu erweitern.

Zur Nutzung des EAI muss das JavaApplet zusammen mit der VRML-Welt in eine HTML-Seite eingebunden werden. Abb. 45 zeigt die dazu notwendige minimale HTML-Seite. Die VRML-Welt wird über das HTML-Tag `EMBED` eingebunden und das JavaApplet direkt über die Tag `APPLET` eingefügt. Beim JavaApplet wird über die Option `MAYSCRIPT` der browserinterne Zugriff des JavaApplets auf den VRML-Browser freigeschaltet.

```
<HTML>
  <BODY>
    <EMBED SRC = "Startwelt.wrl">
    <APPLET CODE = "TRIBUNVisu.class" MAYSCRIPT>
  </APPLET>
  </BODY>
</HTML>
```

Abb. 45: Listing einer HTML-Seite zur Anbindung der Visualisierung

Das Gesamtsystem kann grob durch drei Komponenten dargestellt werden (Abb. 46). Es besteht aus dem Unterwasserfahrzeug mit seinen Sensoren und Aktoren, dem Steuerrechner, der über einen CANBus mit dem Unterwasserfahrzeug verbunden ist und dem Visualisierungsrechner, der über ein TCP/IP-Netzwerk mit dem Steuerrechner verbunden ist. Der Steuerrechner kann dabei auch identisch mit dem Visualisierungsrechner sein.

Die Komponenten des Steuerrechners sind in Abb. 46 ersichtlich. Das eigentliche Steuerprogramm kommuniziert über die CANBus-Hardwareanbindung des PCs mit dem Unterwasserfahrzeug, zeigt dem Bediener Informationen auf dem Bildschirm an und wertet seine Eingaben aus. Zusätzlich ist auf dem Steuerrechner noch ein http-Server aktiv. Dieser übermittelt, wenn er von einem WWW-Browser auf dem Visualisierungsrechner (oder lokal) angesprochen wird, eine HTML-Seite nebst einer VRML-Welt und einem JavaApplet. Nach erfolgter Übertragung baut das JavaApplet eine Verbindung auf und startet die Visualisierung. Auf dem Visualisierungsrechner muss ein WWW-Browser installiert sein, der über eine Java Virtual Machine sowie ein VRMLPlu-

gin verfügt. Zum Start der Visualisierung wird über den WWW-Browser eine Verbindung mit dem Steuerrechner aufgenommen. Von diesem wird dann eine VRML-Welt und ein JavaApplet übertragen, das im folgenden als *TRIBUNVisu* bezeichnet wird.

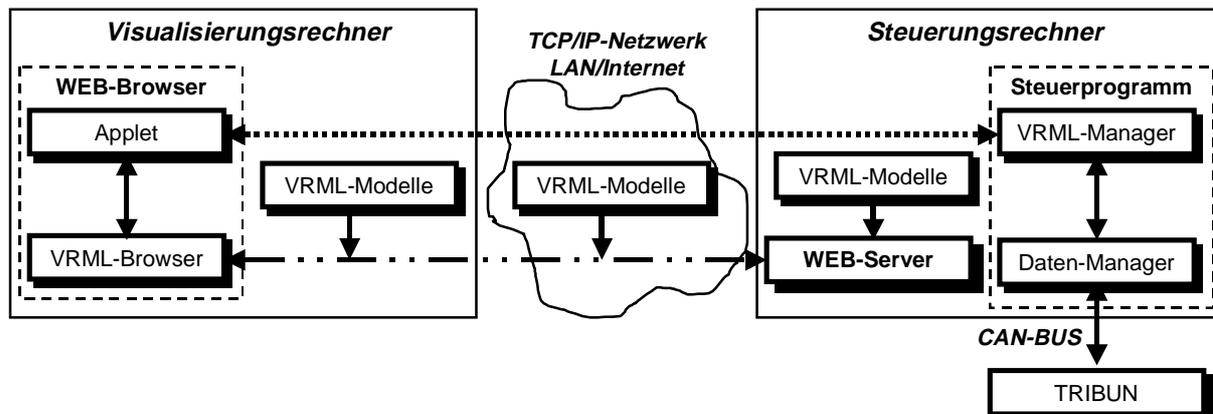


Abb. 46: Anbindung der Visualisierung

Der genaue Ablauf des Verbindungsaufbaus sowie der Visualisierung ist in Abb. 47 dargestellt und gliedert sich in vier Schritte. Nach dem Start der oben genannten Software gibt der Bediener die Internetadresse (URL /104/) des Visualisierungsrechner ein und wählt eine der möglichen Visualisierungsvarianten aus. Als dritter Schritt wird nun die VRML-Startwelt und das JavaApplet *TRIBUNVisu* heruntergeladen. Der WWW-Browser stellt die zunächst noch leere VRML-Welt dar und startet das JavaApplet. Dieses baut zur weiteren Kommunikation mit dem Steuerrechner automatisch eine socketbasierte Verbindung zwischen Applet und dem Datenserver des Steuerprogramms auf. Als Netzwerkprotokoll wird hier das TCP-Protokoll verwendet /99/. Über diese Verbindung erfolgt die Interprozesskommunikation zwischen Visualisierungs- und Steuerprogramm. Zur Kommunikation dient ein ASCII-textbasiertes zeilenorientiertes Protokoll (Kommandozeile). Ein Befehl setzt sich dabei aus dem Befehlsnamen sowie einer Anzahl befehlspezifischer Parameter zusammen. Der Vorteil dieses Protokolltyps ist, dass zum Funktionstest der Schnittstelle z. B. ein Terminalprogramm oder auch ein Webbrowser verwendet werden kann.

Nach erfolgreicher Verbindungsaufnahme wird der Steuerrechner eine virtuelle Welt samt Fahrzeugen übermitteln, die dann im VRML-Fenster dargestellt wird. Zusätzlich werden für jedes sich in der Welt befindliche Fahrzeug einige Informationen im Textbereich der HTML-Seite angezeigt. Trotz des komplexen Ablaufs und der Beteiligung einer hohen Zahl an Programmen läuft der gesamte Vorgang bis auf den ersten Verbindungsaufbau und die Auswahl voll automatisch ab und beschränkt sich für den Anwender auf den Aufruf einer Webseite.

Damit sind die Startphase und der Verbindungsabbau abgeschlossen und es kann mit dem vierten Schritt, der eigentlichen Visualisierung, begonnen werden. Jede Änderung, wie z. B. die Bewegung eines Fahrzeuges wird vom Steuerrechner an den Visualisierungsrechner übermittelt. Diese Informationen werden sofort in der VRML-Ansicht aktualisiert.

Der Datenserver ist nicht wie Webserver oder Applet als eigenständiges Programm anzusehen, sondern stellt einen Teil der Steuerungssoftware dar (siehe auch Kap. 3.7.8). Ausgehend von den eingehenden CAN-Meldungen der angeschlossenen Kom-

ponenten erfolgt eine Aufbereitung der CAN-Nachrichten im Sinne der Bestimmung der vorliegenden Systemkonfiguration, der Betriebszustände sowie der Messwerte. Gemäß der Benutzervoreinstellungen erfolgt dann eine Weiterleitung der Daten an die Visualisierung.

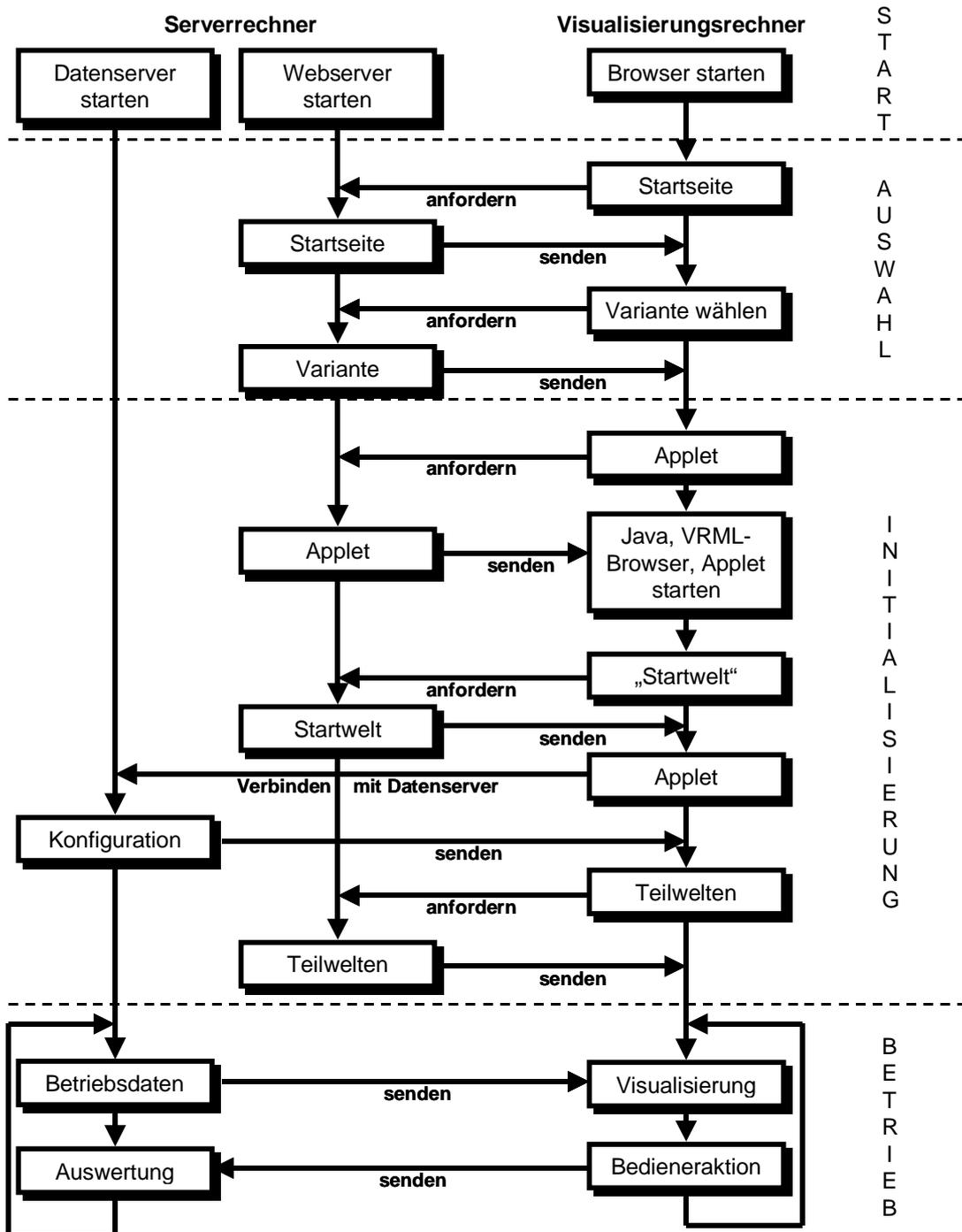


Abb. 47: Verbindungsaufbau und Start der Visualisierung

Der Betrachter kann jetzt die Bewegung des Fahrzeugs beobachten und dabei seinen Standort in der virtuellen Welt frei wählen, indem er mit den vom VRML-Browser bereitgestellten Mechanismen navigiert. Weiterhin hat der Betrachter die Möglichkeit, vorgegebene Kamerapositionen aus einer Liste auszuwählen. Diese können sich an beliebigen Positionen innerhalb der virtuellen Welt befinden oder auch mit bestimmten

Objekten assoziiert sein. Bei Wahl einer objektbezogenen Position wird die Kameraposition, wenn sich das Objekt bewegt, ebenfalls mitbewegt. Dadurch hat man beispielsweise die Möglichkeit, dass der Betrachter die Welt durch die im Fahrzeug befindliche virtuelle Kamera beobachten kann und somit aus der gleichen Perspektive sieht wie der Bediener, der sich am realen Videomonitor befindet.

Für die entwickelte Software wurde die Sprache Java mit dem Sprachumfang der Version 1.1 verwendet. Zur Kompilierung des Quellcodes wurde das Java Development Kit (JDK) der Firma Sun Microsystems in der Version 1.1.7 verwendet [102],[103]. Zudem besteht die Möglichkeit, Java-Programme auch mittels der Borland Entwicklungsumgebung zu erstellen. Dazu muss das Java-Add-On sowie die gewünschte Version des JDKs installiert werden. Gegenüber am Markt erhältlichen Entwicklungsumgebungen, die in den meisten Fällen eine veraltete Implementierung der Sprache Java verwenden, ist somit eine Aktualisierung der Java-Version durch Überschreiben des JDKs problemlos möglich. Zudem ist das JDK der Firma Sun kostenlos erhältlich.

Das Java Development Kit in der Sprachversion 1.2, auch als „Java 2“ bezeichnet, ist vollständig abwärtskompatibel zu Java 1.1. Ein mit dem JDK 1.2 kompiliertes Programm kann auch in einer Virtual Machine (Java-Interpreter) der Version 1.1 arbeiten, sofern nur Funktionen der Version 1.1 verwendet werden. Ein Test der Software ergab, dass der erzeugte Bytecode problemlos mit einer Implementierung der JVM (Java Virtual Machine) in der Sprachversion 1.1 arbeitet.

#### 3.10.4 Datenverarbeitung des Steuerrechners

In diesem Abschnitt sollen kurz einige Aspekte zur Berechnung von inkrementellen Positions- und Lageveränderungen erläutert werden. Da diese Berechnungen oft relativ komplex sind, werden diese möglichst auf dem Steuerrechner in schnellem Maschinencode berechnet, statt auf dem Visualisierungsrechner in langsameren, da interpretiertem, Java-Bytecode.

Zur Visualisierung des Unterwasserfahrzeugs auf dem Visualisierungsrechner muss der Steuerrechner die jeweils aktuellen Positions- und Lage-Daten übermitteln. Dabei werden grundsätzlich absolute Werte verwendet. Die Positionsdaten entsprechen also dem aktuellen Standort des Fahrzeugs. Die Lagedaten werden gemäß der VRML-Konvention durch eine Drehachse (Einheitsvektor) und einen Drehwinkel bezogen auf die Grundstellung (Drehwinkel =  $0^\circ$ ) angegeben. In der hier vorliegenden Implementierung ist das Fahrzeug derart ausgerichtet, dass die Längsachse entlang der x-Achse verläuft, die Hochachse entlang der y-Achse und die Querachse entlang der z-Achse.

Steht im Steuerprogramm nur eine Datenquelle zur Verfügung, die inkrementelle Daten liefert (z. B. durch Auswertung von Positionsdaten, die aus Beschleunigungssensoren gewonnen wurden), müssen die absoluten Positions- und Lagedaten im Programm des Steuerrechners zwischengespeichert werden. Derart gewonnene Daten beziehen sich meistens auf das lokale Koordinatensystem des Fahrzeugs, zum Beispiel entspricht eine Vorwärtsbewegung einer Bewegung in Richtung der Längsachse des Fahrzeugs.

Da die übertragenen Koordinaten sich immer auf das globale Koordinatensystem (KS) ( $x_P, y_P, z_P$ ) beziehen, muss vor der Aktualisierung der Positionsdaten erst eine Umrechnung der Bewegungsdaten vom lokalen Koordinatensystem des Fahrzeugs in das globale Koordinatensystem vorgenommen werden. Dazu behilft man sich mit den bekannten Lagedaten des Fahrzeugs im globalen KS, die durch die Drehach-

se  $\vec{a} = (a, b, c)$  und Drehwinkel  $\alpha$  gegeben sind /84/. Zur Umrechnung der Koordinatensysteme verwendet man eine Drehmatrix  $M$ . Diese ist für den gegebenen Fall gegeben durch:

$$M = \begin{bmatrix} a^2 - a^2 \cos(\alpha) + \cos(\alpha) & ab - ab \cos(\alpha) - c \sin(\alpha) & ac - ac \cos(\alpha) + b \sin(\alpha) \\ ab - ab \cos(\alpha) + c \sin(\alpha) & b^2 - b^2 \cos(\alpha) + \cos(\alpha) & bc - bc \cos(\alpha) - a \cdot \sin(\alpha) \\ ac - ac \cos(\alpha) - b \sin(\alpha) & bc - bc \cos(\alpha) + a \cdot \sin(\alpha) & c^2 - c^2 \cos(\alpha) + \cos(\alpha) \end{bmatrix} \quad (Gl. 39)$$

Mit  $\vec{x}_{\text{Neu}} = M \vec{x}_{\text{Alt}}$  ergibt sich für die aktuelle Längsachse (1,0,0) des Fahrzeugs im globalen KS zu:

$$\vec{x}_{\text{Laengs}} = M \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a^2 - a^2 \cos(\alpha) + \cos(\alpha) \\ ab - ab \cos(\alpha) + c \sin(\alpha) \\ ac - ac \cos(\alpha) - b \sin(\alpha) \end{pmatrix} \quad (Gl. 40)$$

Entsprechend ergibt sich die Hochachse (0,1,0) zu:

$$\vec{x}_{\text{Hoch}} = M \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} ab - ab \cos(\alpha) - c \sin(\alpha) \\ b^2 - b^2 \cos(\alpha) + \cos(\alpha) \\ bc - bc \cos(\alpha) + a \sin(\alpha) \end{pmatrix} \quad (Gl. 41)$$

und die Querachse (0,0,1) zu:

$$\vec{x}_{\text{Quer}} = M \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} ac - ac \cos(\alpha) + b \sin(\alpha) \\ bc - bc \cos(\alpha) - a \cdot \sin(\alpha) \\ c^2 - c^2 \cos(\alpha) + \cos(\alpha) \end{pmatrix} \quad (Gl. 42)$$

Sei  $(\delta_x, \delta_y, \delta_z)$  die Verschiebung des Fahrzeugs entlang seiner Längs-, Hoch- und Querachse seit der letzten Aktualisierung der globalen Positionsdaten  $(x_P, y_P, z_P)$ , so lassen sich die neuen Positionsdaten  $(x_{P\text{Neu}}, y_{P\text{Neu}}, z_{P\text{Neu}})$  berechnen zu:

$$\begin{aligned} x_{P\text{Neu}} &= x_P + (a^2 - a^2 \cdot \cos(\alpha) + \cos(\alpha))\delta_x \\ &\quad + (ab - ab \cos(\alpha) - c \sin(\alpha))\delta_y \\ &\quad + (ac - ac \cos(\alpha) + b \sin(\alpha))\delta_z \\ y_{P\text{Neu}} &= y_P + (ab - ab \cdot \cos(\alpha) + c \sin(\alpha))\delta_x \\ &\quad + (b^2 - b^2 \cos(\alpha) + \cos(\alpha))\delta_y \\ &\quad + (bc - bc \cos(\alpha) - a \sin(\alpha))\delta_z \\ z_{P\text{Neu}} &= z_P + (ac - ac \cdot \cos(\alpha) - b \sin(\alpha))\delta_x \\ &\quad + (bc - bc \cos(\alpha) + a \sin(\alpha))\delta_y \\ &\quad + (c^2 - c^2 \cos(\alpha) + \cos(\alpha))\delta_z \end{aligned} \quad (Gl. 43)$$

Zur Berechnung der Lagedaten ist ein höherer Aufwand notwendig als zur Berechnung der Positionsdaten, da Drehungen im dreidimensionalen Raum nicht kommutativ sind. Daher verwendet die Sprache VRML zur eindeutigen Darstellung von Drehungen folgendes Schema:

Die Drehung eines Objekts bzw. einer Gruppe von Objekten wird durch eine Drehachse und einen Drehwinkel modelliert, die auf die Lage des Objekts in seiner Grundstellung (Drehwinkel =  $0^\circ$  bei beliebiger Drehachse) bezogen sind.

Ein Algorithmus zur Berechnung der neuen Drehachse und des neuen Drehwinkels aus der alten Achse und Winkel sowie einer zusätzlichen Drehung ist unter /101/ zu finden. Dabei werden die Daten erst in eine sogenannte quaternionForm umgerechnet, verknüpft (Multiplikation von Quarternionen anstelle von Drehmatrix-Multiplikation) und wieder zurückkonvertiert. In der quaternionForm können beliebig viele inkrementelle Drehungen hintereinander durchgeführt werden bis zurückkonvertiert wird.

1. Schritt: Umwandlung der Drehachse und des Drehwinkels in quaternionForm (beschreibt Lage eines Objekt im verwendetem globalen KS), die Drehachse  $\vec{a} = (a, b, c)$  muss hierbei ein Einheitsvektor sein.

$$\begin{aligned} q_1[0] &= a \cdot \sin(\alpha/2) \\ q_1[1] &= b \cdot \sin(\alpha/2) \\ q_1[2] &= c \cdot \sin(\alpha/2) \\ q_1[3] &= \cos(\alpha/2) \end{aligned} \quad (Gl. 44)$$

2. Schritt: Umwandlung der inkrementellen Drehachse  $\vec{\delta}_a = (\delta_a, \delta_b, \delta_c)$  und des Drehwinkels  $\delta_\alpha$  in quaternionForm ( $q_2[0-3]$ ) entsprechend der Gleichungen in Schritt 1.

3. Schritt: Multiplikation der Quarternionen

$$\begin{aligned} q_3[0] &= q_2[3]q_1[0] + q_2[0]q_1[3] + q_2[1]q_1[2] + q_2[2]q_1[1] \\ q_3[1] &= q_2[3]q_1[1] + q_2[1]q_1[3] + q_2[2]q_1[0] + q_2[0]q_1[2] \\ q_3[2] &= q_2[3]q_1[2] + q_2[2]q_1[3] + q_2[0]q_1[1] + q_2[1]q_1[0] \\ q_3[3] &= q_2[3]q_1[3] + q_2[0]q_1[0] + q_2[1]q_1[1] + q_2[2]q_1[2] \end{aligned} \quad (Gl. 45)$$

4. Schritt: Sollen weitere inkrementelle Drehungen angewendet werden, so ist  $\vec{q}_3$  gleich  $\vec{q}_1$  zu setzen und ab Schritt 2 zu wiederholen.

5. Schritt: Konvertierung des Ergebnisses aus der quaternionForm nach Drehachse und Drehwinkel gemäß:

$$\alpha = \arccos(q_3[3]) \cdot 2 \quad x = \frac{q_3[0]}{\sin(\alpha/2)} \quad y = \frac{q_3[1]}{\sin(\alpha/2)} \quad z = \frac{q_3[2]}{\sin(\alpha/2)} \quad (Gl. 46)$$

Für den Sonderfall  $\alpha = 0$  liegt keine Drehung des Objekts vor, wodurch die Achse frei wählbar (z. B.: (0,0,1) SFRotation-Initialisierungswert) ist.

Wird das Fahrzeug nur um eine Achse gedreht, so muss, wie bei der Positionsberechnung, die aktuelle Drehachse bestimmt werden. Damit kann der obige Algorithmus verwendet werden. Dreht sich das Fahrzeug jedoch kontinuierlich gleichzeitig um mehr als eine seiner Achsen, z. B. um Längs- und Hochachse, so muss berücksichtigt werden, dass sich die Drehachsen im globalen Koordinatensystem ständig ändern. Somit ist es

erforderlich, diese Drehung durch Aufsummieren infinitesimal kleiner Teildrehungen zu modellieren.

### 3.10.5 Funktionalität von TRIBUNVisu

Die erstellte Schnittstelle bietet vielfältige Möglichkeiten für die Visualisierung von Handhabungsgeräten. Grundsätzlich kann die Konfiguration von Handhabungssystem und Arbeitsumgebung vollständig über die Socket-Schnittstelle während der Laufzeit vorgegeben und beliebig animiert werden.

Es gibt dabei keine Unterscheidung zwischen Modellierung und Visualisierungsphase, alle Aktionen können vollständig über die Socket-Schnittstelle während der Laufzeit im Gegensatz zu gängigen Programmen jederzeit ausgeführt werden.

- Die Arbeitsumgebung kann durch Hinzufügen und Entfernen beliebiger Teilwelten sowie durch Vorgabe von Lage, Position und Größe modelliert werden. Dabei besteht die Option jedes einzeln eingefügte Objekt zu animieren.
- Weiterhin besteht die Möglichkeit, Roboterkinematiken zu modellieren. Die bidirektionale Schnittstelle gestattet dabei sowohl die Vorgabe der Achsstellungen als auch die Bewegung der Achsen in der virtuellen Welt durch den Bediener und die Weiterleitung der Positionsinformation zum realen Gerät.
- Zur weiteren Interaktion mit dem Bediener steht ein frei konfigurierbares Bedienpanel bereit, das über Schalter, Schieberegler und Textfelder zur Einblendung von wichtigen Systemdaten verfügt.
- Gerätespezifisch lässt sich TRIBUN mit beliebigen Modulen konfigurieren und bewegen.
- Grundsätzlich können mehrere Unterwasserfahrzeuge und Manipulatoren gleichzeitig verwaltet und ihr Zusammenwirken bei der Durchführung komplexer Arbeiten ebenfalls dargestellt werden.

Die Visualisierung kann durch die Socket-Schnittstelle auch leicht an andere Applikationen angebunden und über die Kommandofunktionen gesteuert werden.

### 3.10.6 Anwendungsbeispiele der 3D-Visualisierung

In Abb. 48 ist eine typische Arbeitsumgebung für TRIBUN, bestehend aus Wasserbecken und Leitstand, wiedergegeben. Im Wasserbecken befindet sich das Fahrzeug ausgestattet mit Propellerantrieben sowie der Zweiachsen-Linearführungsmaschine. Das virtuelle Bedienpanel verfügt auf der linken Seite über drei Ein/Aus-Button und auf der rechten Seite über zwei Schieberegler. Im unteren Bereich stehen zwei Textzeilen zur Einblendung der Systeminformationen bereit. Für die Modellierung von Roboterkinematiken gelten dabei folgende Regeln:

1. Es können kinematische Ketten aus einer Folge von Translations- und Rotationsachsen gebildet werden. Die Nummerierung der Achsen erfolgt von eins bis n, wobei Achse n zur Aufnahme des Werkzeugs dient. Der Grundkörper (Sockel) erhält die Nummer null.
2. Die Translation erfolgt bei Translationsachsen entlang der x-Achse des Bauteils, die

Rotation erfolgt bei Rotationsachsen entsprechend um die x-Achse des Bauteils. Die x-Achse muss in der VRML-Beschreibung so gelegt werden, dass sie mit der Bewegungsachse übereinstimmt.

- Die relativen Übergänge der KS von Achse  $i$  auf Achse  $(i+1)$  werden gemäß der VRML-Konvention durch eine Translation durch  $d_x$ ,  $d_y$ , und  $d_z$  um die x-, y-, und z-Achse und eine Drehachse mit Drehwinkel durch  $a_x$ ,  $a_y$ ,  $a_z$  und  $\alpha$  (bezogen auf das  $i$ -te KS) beschrieben.

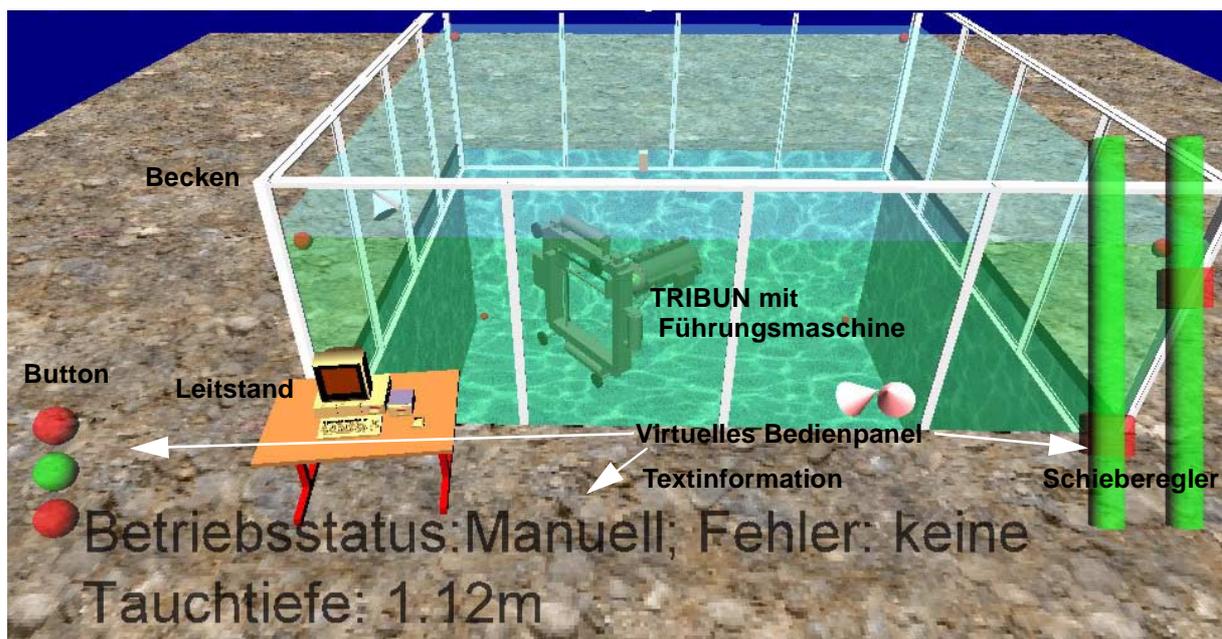


Abb. 48: Beispiel für virtuelle Arbeitsumgebung

Abb. 49 zeigt beispielhaft, wie Roboterkinematiken aus einzelnen Komponenten zusammengesetzt werden können. Die Zweiachsen-Linearführungsmaschine besteht aus den Modulen Rahmen, Achse 1 und Achse 2. Diese werden in VRML modelliert und mittels der implementierten Kommandos zu einer Einheit zusammengesetzt. Danach können die beiden Achsen über die Schnittstelle animiert werden.

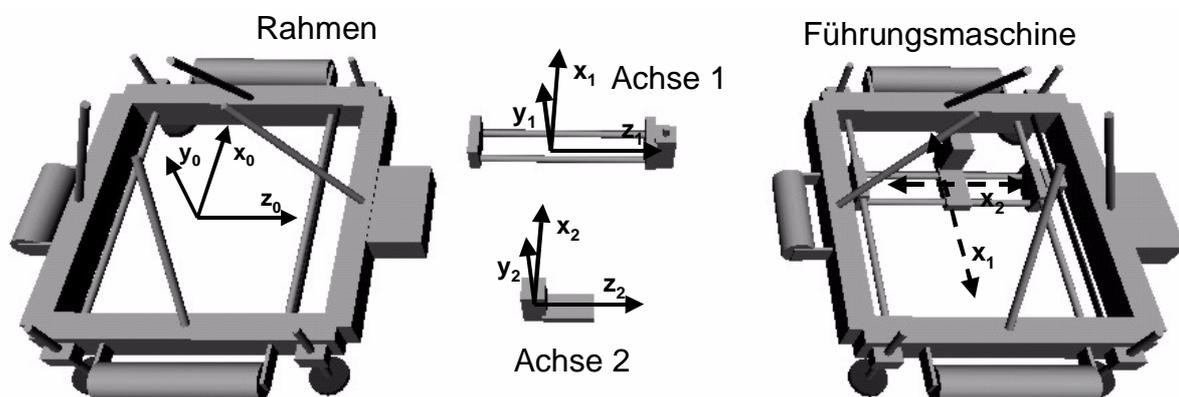


Abb. 49: Zusammensetzen der Führungsmaschine aus Einzelkomponenten

Tabelle 11 : Parameter für die Zweiachsen-Führungsmaschine

Nr.	Art	VRML-Name	$d_x$ [m]	$d_y$ [m]	$d_z$ [m]	$a_x$ [m]	$a_y$ [m]	$a_z$ [m]	$\alpha$ [rad]	$a_{min}$	$a_{max}$
0	x	rahmen.wrl	0.45	-0.5	0	0	0	1	1.57	0	0
1	T	achse_1.wrl	-0.1	0	0	0	0	1	0	0	0.5
2	T	achse_2.wrl	0	0	-0.17	0	1	0	-1.57	0	0.5

Achsart: x=Grundkörper, T=Translationsachse, R=Rotationsachse  
 Min.=Achsuntergrenze; Max.=Achsobergrenze in Grad bzw. Meter

Für die 0-te Achse (Rahmen) sind die Verschiebungen und die Drehung nur für die Verbindung mit TRIBUN notwendig. Die Achse 1 muss gegenüber dem Rahmen in negativer x-Richtung (Offset) verschoben werden. Achse 2 benötigt neben der Translation zusätzlich eine Drehung um -90 Grad um ihre  $y_2$ -Achse, damit die Achsbewegungsrichtung ( $x_2$ -Achse) mit der gewünschten Bewegungsrichtung übereinstimmt.

Für die Verwaltung der Szenen steht im TRIBUN-Steuerungsprogramm ein VRML-Manager bereit (Abb. 50). Dieser gestattet über diverse Bedienelemente Objekte in die virtuelle Welt einzufügen und zu manipulieren. Dabei können komplette Szenen auch abgespeichert und zu einem späteren Zeitpunkt wieder geladen werden. Ebenso können über eine einfache Ablaufsteuerung Animationen erstellt werden.

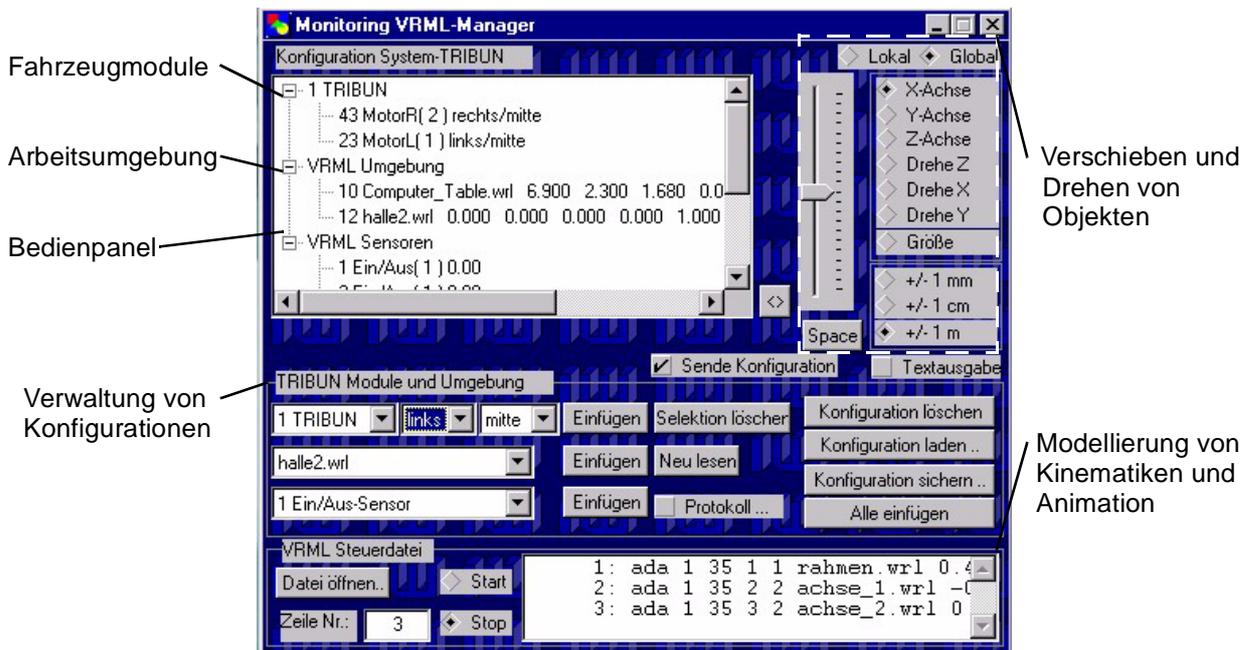


Abb. 50: VRML-Manager zur Verwaltung der virtuellen Szene

## 4 Durchgeführte Tests und Ergebnisse

### 4.1 Testumgebung

Zur Inbetriebnahme des Systems und zur Durchführung von Fahrversuchen wurde in der Versuchshalle des Unterwassertechnikums Hannover ein Leitstand aufgebaut (Abb. 51). Er besteht in seiner jetzigen Konfiguration aus der zentralen Energieversorgung, zwei Farbfernsehern zur Wiedergabe der Videobilder und dem HMD auf dem die Bilder alternativ dargestellt werden können. Die Steuerung des Fahrzeugs erfolgt über einen PC, auf dem die in Kap. 3.7 beschriebene Steuerungssoftware läuft. Alle Eingabegeräte, wie Joystick oder SpaceMouse, können dabei direkt an den Steuerungs-PC angebunden werden. Der zweite PC dient im Wesentlichen zur Darstellung der 3D-Visualisierung und der Anbindung einer WebCam.



Abb. 51: Leitstand und TRIBUN am Testbecken

Als Testumgebung dient dabei ein Wasserbecken, das die Abmessungen 5\*5\*4 m (Länge, Breite, Tiefe) hat. Das Becken ist dabei in der Höhe in zwei Bereiche unterteilt, wobei der obere Teil aus einer Stahlrahmenkonstruktion (Höhe ca. 1,6 m) und der untere Teil aus einer Grube besteht. In die Rahmenkonstruktion sind Plexiglasscheiben eingesetzt, so dass das Becken von allen Seiten eingesehen werden kann. Vor dem Becken befindet sich der Leitstand.

### 4.2 Inbetriebnahme der Module und Konfiguration des Systems

Durch die standardisierten Multifunktionsleisten können die extern am Rumpf angebrachten Module innerhalb weniger Minuten adaptiert oder ausgetauscht werden. Schwerere Komponenten wie Manipulatoren werden zur Entlastung der Steckverbin-

dung zusätzlich mit U-Klammern an den Leisten fixiert. Zum Öffnen des Fahrzeugs wird der Schnellverschluss des Deckels entriegelt und der Rumpf mit einem leichten Überdruck beaufschlagt und der Deckel entfernt. Die internen Komponenten lassen sich dann leicht aus den Führungsschienen entnehmen. Schraubverbindungen müssen dabei nur bei den U-Klammern und der Stereokamera gelöst werden. Durch die einfache Montagetechnik kann sichergestellt werden, dass alle Module in weniger als fünf Minuten ab- bzw. ausbaubar sind.

Grundsätzlich bleibt somit festzustellen, dass durch die Bereitstellung der elektromechanischen Kupplungen der Multifunktionsleisten eine einfache Konfigurierbarkeit erreicht wird. Als Nachteil ist hier jedoch zu nennen, dass die fehlende Auftriebsneutralität der einzelnen Komponenten ein Austrimmen des Fahrzeuges notwendig macht. Konstruktionsbedingt kann diese nicht erreicht werden, da die Abmessungen der Module ein sinnvolles Maß überschreiten würden. Um diesen Vorgang für den späteren Einsatz zu beschleunigen, können für bereits getestete Konfigurationen auch die notwendigen Maßnahmen zum Austrimmen angegeben werden.

Für die schnelle steuerungstechnische Konfiguration ist das hier vorgestellte CAN-Busbasierte Plug&Play-Konzept ein Gewinn, da alle Komponenten einzeln im Labor ausgiebig getestet werden können, bevor sie in das Gesamtsystem integriert werden. In der hier implementierten Form dient der Bus allen Knoten in gleicher Weise zum Transport der Nachrichten. Ob die angeschlossenen Komponenten miteinander interagieren, hängt dabei von der implementierten übergeordneten Leitsoftware ab. Ein weiterer wesentlicher Vorteil des hier entwickelten Modulkonzeptes ist die Austauschbarkeit der Module zwischen einzelnen Handhabungssystemen. Insbesondere im Bereich der Experimentalsysteme und deren Erprobung kann hier ein deutlicher Zeitgewinn verzeichnet werden.

Die Möglichkeit, die Software in die einzelnen Scheckkartenrechner per Download zu übertragen, bietet in der Entwicklung und Optimierungsphase ein sehr bequemes Werkzeug, da auf einen zeitraubenden Ein- und Ausbau, ein Öffnen der Gehäuse sowie das Brennen einer neuen EPROM-Version verzichtet werden kann.

Der CAN-Bus wird mit einer Datenrate von 100 kBaud betrieben, wobei die maximale Buslänge über 500 m liegen darf. Die typische Busauslastung liegt bei Parallelbetrieb aller momentan verfügbarer Komponenten zwischen 5 -15 % der maximal möglichen, sodass ausreichende Kapazitäten für weitere Module bereitstehen. Der CAN-Bus erweist sich als sehr geeignetes Bussystem auch für die Handhabungstechnik im Bereich kleinbauender Systeme. Die Kosten, die sich durch die zusätzliche Elektronik im Vergleich zu einer Einrechner-Lösung ergeben, werden durch die Vorteile im Bereich Vernetzung, Inbetriebnahme und Wartung mehr als wett gemacht. Als CAN-Busschnittstelle für den PC hat sich ein sogenanntes CAN-Dongle als besonders praktisch herausgestellt. Dieses Dongle wird auf die parallele Schnittstelle (Drucker-Port) des Rechners gesteckt und über den Tastaturanschluss mit der benötigten elektrischen Energie versorgt. Gegenüber den ebenfalls getesteten PC-Einsteckkarten hat das Dongle den Vorteil, dass es in extrem kurzer Rüstzeit an jeden Rechner, insbesondere auch an Laptops, adaptiert werden kann.

## 4.3 Fahrverhalten und Bedienung

### 4.3.1 Bedienung und Bewertung der MMIs

Die Steuerung der Fahrzeugbewegung erfolgt über die Spacemouse, die an den PC über die serielle Schnittstelle (RS232) angebunden wird. Die Steuerungssoftware rechnet die Steuersignale der SpaceMouse in Abhängigkeit der verfügbaren Antriebe und deren Kraftwirkungsrichtungen in Stellbefehle für die Propeller um. Dabei werden die Soll Drehzahlen direkt proportional der Auslenkung an die Propellerantriebe weitergeleitet, wenn der entsprechende Antrieb eine Bewegung in der oder um die entsprechende Fahrzeugdrehachse (Abb. 18) bewirkt. Das Fahrzeug lässt sich damit so führen, als würde man es direkt anfassen und mit der Hand verschieben und drehen. Es lässt sich manuell im Zentimeterbereich mit nur einer Hand positionieren, ist extrem wendig und kann auch in engen Räumen gut manövriert werden. Durch die intuitive Bedienbarkeit verringert sich die Zeit für ein Bedienertraining deutlich. Über die Taster der SpaceMouse sind Aktionen wie Öffnen und Schließen eines Greifers oder Ein- und Ausschalten der Beleuchtungsmodule realisiert. Für viele Anwendungsfälle ist damit nur eine Hand zur Bedienung erforderlich.

Ein weiterer Vorteil der Steuerungssoftware ist es, dass die einzelnen Freiheitsgrade der SpaceMouse softwaremäßig abgeschaltet werden können, so dass der Bediener in gewissen Situationen zusätzlich entlastet wird. Der Betriebsmodus "Stärkste Auslenkung" bei translatorischen oder rotatorischen Bewegungen ermöglicht ein definiertes Verfahren in nur einem der gewünschten Bewegungsarten (translatorisch oder rotatorisch).

Das Head-Mounted-Display ist immer dann von Vorteil, wenn es auf eine gute räumliche Orientierung ankommt, z. B. beim Greifen von Gegenständen oder dem Abschätzen von Entfernungen. Die automatische Steuerung der Kameras mittels des Headtrackers liefert ein gutes Ergebnis. Es ist möglich, das HMD mehrere Stunden ohne Probleme zu tragen. Obwohl das HMD eine relativ geringe Auflösung aufweist und die Mechanik der Kamera einfach aufgebaut ist, kann ein ausreichend guter Stereoeindruck erzielt werden. Nachteilig wirkt sich die geschlossene Ausführung des Systems aus, da der Bediener keinen Blick auf die Bedienoberfläche oder die 3D-Visualisierung werfen kann. Abhilfe schafft hier z. B. die Videobrille von Olympus. Sie gestattet bei deutlich besserer Darstellungsqualität des Bildes unter dem Gestell durchzuschauen, wobei allerdings die Stereofähigkeit verloren geht.

Welche Videobrille man wählt, hängt dabei stark von dem Einsatzszenario ab. Für eine optische Inspektion kann beispielsweise in gewissen Bereichen auch ganz auf das Schwenken der Kamera verzichtet und nur feste Einstellungen gewählt werden, ohne dass eine zu große Benachteiligung auftritt.

Durch die in dieser Arbeit vorgestellte streng objektorientierte Implementation der Klassen und Methoden wird sichergestellt, dass die Software leicht erweiterbar und wartungsfreundlich ist. Eine Änderung in der Hardware-Anbindung ist somit ebenso problemlos realisierbar wie die leichte Integration zusätzlicher Funktionsmodule. Trotz der erheblichen Komplexität der verschiedenen Softwaremodule kann auch hier eine Langzeitsystemstabilität nachgewiesen werden. Die Software kann damit auch als Basis für zukünftige Handhabungssysteme dienen, wobei große Teile wieder verwendet

werden können. Das PC-Steuerungs- und Bedienprogramm arbeitet für alle getesteten Szenarien einwandfrei.

### 4.3.2 Rumpf mit Antriebs- und Greifarmvariationen

Typische Aufgaben von ROVs sind die optische Inspektion sowie das Manipulieren von Kleinteilen. Unter einer großen Anzahl von denkbaren Fahrzeugkonfigurationen sind für die Durchführung dieser Arbeiten zwei bzw. vier Propellerdoppelantriebe und zusätzlich ein Greifer notwendig. Die Propellerantriebe werden seitlich (symmetrisch links und rechts) am Fahrzeugrumpf und der Greifer am Kiel des Fahrzeugs adaptiert. Dabei verschiebt sich durch den Greifer der Schwerpunkt nach unten, wodurch gleichzeitig eine passive Zwangstrimmung und Stabilisierung erreicht wird. Die Funktionen für Öffnen und Schließen des Greifers werden dabei auf die freien Tasten der SpaceMouse gelegt, so dass für die gesamte Bedienung nur eine Hand benötigt wird. Tests haben gezeigt, dass sich das Fahrzeug zuverlässig und präzise mit der SpaceMouse führen lässt.

Der Versorgungskabelstrang wirkt sich durch die gute Trimmung kaum störend auf das Fahrverhalten aus. Innerhalb des Wasserbeckens kann das 50 m lange Kabel problemlos vom Fahrzeug mitgezogen werden. Durch den leicht positiven Auftrieb treibt das Kabel immer an die Oberfläche zurück und ist bei Manövern nicht im Weg, da es sich dadurch oberhalb des Fahrzeugs befindet.

### 4.3.3 TRIBUN mit Führungsmaschine

Abb. 52 zeigt die an TRIBUN adaptierte Führungsmaschine. Die Adaption erfolgt mittels der vier Stangen sowie der Halterungsklemmen an den Multifunktionsleisten. Trotz der erheblichen Größe des Gesamtsystems lässt sich die Einheit mit nur zwei Antrieben ausreichend gut und genau manövrieren. Die integrierte Haltefunktion mittels der Sauggreifer gestattet eine sichere Ankopplung an die Oberfläche. Die voll automatische Ablaufsteuerung gestattet eine Durchführung von einfachen Arbeitsgängen wie z. B. das Abtasten der Oberfläche.



Abb. 52: TRIBUN mit Führungsmaschine in Montageposition (l.) und angesaugt an Beckenwand (r.)

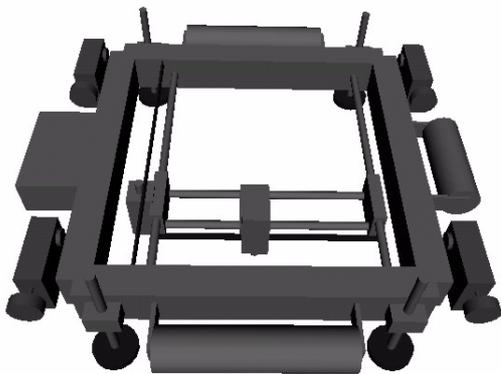


Abb. 53: Erweiterung der Führungsmaschine mit vier Antrieben

Das modulare Steuerungskonzept gestattet einfache Erweiterungen der Führungsmaschine. So kann z. B. die Funktionalität der Führungsmaschine durch das Anbringen von Multifunktionsleisten erweitert werden. Damit ließen sich auch vier Propellertriebe integrieren und die Führungsmaschine als ROV aufrüsten (Abb. 53). Die Steuerung kann direkt über TRIBUN Steuerungssoftware erfolgen, wenn eine zu TRIBUN äquivalente Steckplatzkennung gewählt wird.

#### 4.4 Test der 3D-Visualisierung

Die Visualisierung benötigt einen WWW-Browser auf dem Visualisierungsrechner, um die Software auszuführen. Zum Zeitpunkt dieser Arbeit existieren der *Internet Explorer* der Firma Microsoft und der *Communicator* der Firma Netscape als einzige WWW-Browser, die die Voraussetzungen für diese Aufgabe erfüllen /105/,/106/. Diese sind zum einen die Implementierung eines Java-Interpreters (Java Virtual Machine) und zum anderen die Möglichkeit, einen VRML-Browser als Plugin einzubinden. Beide WWW-Browser und die getesteten VRML-Browser sind kostenlos erhältlich. Getestet wurde die Software unter Windows 95, 98 sowie NT, wobei als Webbrowser *Netscape Communicator* und der *Internet-Explorer* zum Einsatz kommt. Als VRML-Browser wurde sowohl der *Cosmoplayer* der Firma Cosmo-Software, der *Cortona VRML Client* von Parallel-Graphics sowie *Blaxxun Contact* von Blaxxun eingesetzt /107/, /108/,/109/. Unter UNIX (SGI) wurde als Web-Browser *Netscape* und als VRML-Browser der *Cosmoplayer* getestet.

Die erstellte Schnittstelle arbeitet mit den drei getesteten VRML-Browsern fehlerfrei zusammen. Geringfügige Unterschiede gibt es in der Darstellung und der Bedienung. Welcher Browser vorzugsweise verwendet, wird hängt dabei weitestgehend von den Präferenzen des Bedieners ab. Durch die Möglichkeit, verschiedene VRML-Browser einzusetzen, wird eine Herstellerunabhängigkeit erreicht, so dass bei fehlenden Updates eines Produktes auf einen anderen Hersteller ausgewichen werden kann. Der Vorteil dieser Art der Visualisierung liegt dabei auch in der Plattformunabhängigkeit, d. h. sie kann unter verschiedenen Betriebssystemen eingesetzt werden.

Weiterhin wird ein http-Server auf dem Steuerrechner benötigt, um dem Visualisierungsrechner eine Verbindung mit dem Steuerrechner zu ermöglichen. Er ist für die Übertragung der HTML-Seite, des Java-Applets sowie der VRML-Welten zuständig. Die Wahl fiel auf das Programm *TinyWeb*, da es einerseits den gestellten Anforderungen genügt und andererseits sehr einfach zu konfigurieren ist /110/. Darüber hinaus ist es ebenfalls kostenlos erhältlich.

Die 3D-Visualisierungssoftware gestattet eine sehr gute Unterstützung des Bedieners bei der Durchführung der Arbeiten. Die Software hat dabei nicht nur darstellenden Charakter, sondern kann auch interaktiv für Steuerungszwecke genutzt werden, wobei

sowohl das virtuelle Bedienpanel als auch modellierte Roboterkinematiken zum Einsatz kommen. Dabei werden die durch den Bediener ausgeführten Aktionen in der virtuellen Welt an das TRIBUN-Steuerprogramm weitergeleitet. Beide Möglichkeiten wurden exemplarisch getestet, um die 2-Achsen-Führungsmaschine zu bedienen. Zum einen können über zwei Schieberegler des Bedienpanels direkt die Achspositionen vorgegeben werden, zum anderen wird durch Verschieben der Achsen in der virtuellen Umgebung die Positionssteuerung ausgelöst. Gerade bei der zweiten Möglichkeit wird auch hier eine intuitive Bedienbarkeit erreicht.

Die Bildaktualisierungsrate (Framerate) und damit auch die Qualität der Visualisierung hängt grundsätzlich stark von der Komplexität der verwendeten VRML-Modelle ab. Mittlerweile verwenden viele CAD-Programme VRML-Konverter, die automatisch aus den Konstruktionsdaten VRML-Modelle erzeugen, die dann sofort für die Modellierung verwendet werden können. Um eine fließende und kontinuierlich Darstellung der virtuellen Szene zu gewährleisten, erfolgt die Aktualisierung der Fahrzeugposition und -lage im Normalbetrieb 10-mal pro Sekunde. Dabei beträgt die Datenrate auf dem Socket max. 1 kByte/s. Bei komplexen Modellen kann es allerdings zu zeitlichen Verzögerungen kommen, da die Berechnungszeit höher ist als der zeitliche Abstand zwischen den ein-treffenden Positions- und Lageinformationen. Diese Verzögerungseffekte können bewirken, dass die Echtzeitfähigkeit der Software verloren geht. Abhilfe schafft hier eine Abfrage der aktuellen Framerate sowie die Anpassung der Datenaktualisierungsrate an die aktuelle maximal mögliche. Die Komplexität der Modelle hängt dabei stark von der Qualität und den Einstellungsmöglichkeiten des Konverters ab, die leider oft noch als unzureichend anzusehen sind. Oft können Objekte auch alternativ hinreichend gut aus geometrischen Primitiven modelliert werden, die geringere Berechnungszeiten benötigen.

Die hier vorgestellte Software zur Visualisierung bietet überall dort ein gutes Einsatzpotenzial, wo auf eine teure Spezialsoftware verzichtet werden kann. Große Teile können ohne Änderung sofort im Bereich der Handhabungstechnik eingesetzt werden. Dazu zählen die Modellierung von Roboterkinematiken, der Arbeitsumgebung sowie des virtuellen Bedienpanels. Gerätespezifische Erweiterungen, wie sie durch die VRML-Struktur notwendig sind, müssen allerdings durch eine Änderung der Software realisiert werden.

Die Einsatzmöglichkeiten reichen dabei von dem hier vorgestellten Monitoring bis zur Verwendung für die Simulation. Auch die Abfrage z. B. von Fertigungszellen ist denkbar, wobei dem Anwender statt der klassischen Bedien- und Anzeigeoberfläche dreidimensionale Darstellungsmöglichkeiten gegeben werden, die eine verbesserte Orientierung erlauben. Die Bedienung und Abfrage kann dabei mittels eines normalen Webbrowsers ohne komplizierte Nachinstallation von Software erfolgen.

Nachteilig ist allerdings der erhöhte Speicheraufwand, da immer ein Web-Browser aktiv ist und die langsamere Ausführungszeit gegenüber Standard-Anwendungen, die durch den Java-Interpreter bedingt ist. Bei Bedarf kann der Vorteil der Netzwerkfähigkeit der Software genutzt werden, die eine Verteilung von Visualisierung und Simulation auf verschiedene Rechner zulässt. Insbesondere kann so das rechenzeitintensive Monitoring auf leistungsfähige Rechner verlagert werden. Weiterhin können die Modelle auf allen verfügbaren Rechnern im gesamten Netzwerk (oder Internet) verteilt werden.

Die bisher gewonnenen Ergebnisse zeigen, dass die Visualisierung und das Monitoring mit VRML eine kostengünstige Unterstützung bei der Durchführung von Handhabungsaufgaben gerade für den PC-Bereich bietet.

## 4.5 Fernhantierung im Netzwerkbetrieb

### 4.5.1 Fernhantierung über LAN und Modem

Für die Übertragung der Videobilder wurden exemplarisch verschiedene Programme aus dem Bereich der Internet-Bildtelefonie wie *Webphone 3.0* der NetSpeak Corporation und *IVisit* von Eyematic Interfaces getestet /111/,/112/. Beide Programme weisen jedoch den Nachteil auf, dass das Bild nicht skalierbar ist. Gegenüber sonst meist ähnlicher Funktionalität weist *Internet Phone 5* der Firma Vocaltec diesen Nachteil nicht auf und kommt daher hier zum Einsatz /113/. Eine besonders praktische Lösung stellt *Web-Video* der Universität Ulm dar, da die Bilder direkt in einem Internet-Browser innerhalb einer Webseite als JPEG-Stream angezeigt werden können /114/. Die Installation einer speziellen Videosoftware auf dem Leitrechner entfällt dabei. Dabei werden je nach verwendeter Hardware (PC-Typ und Rechenleistung) Bildraten zwischen drei und maximal fünf Frames/Sekunde erzielt. Die Videodaten werden mit ca. 50 kByte/s pro Bildkanal bereitgestellt und übertragen, wobei das LAN eine Bandbreite 10 MBit/s aufweist.

Die Anzahl der CAN-Komponenten beträgt für die getesteten Szenarien zwischen fünf und zehn, wobei eine durchschnittliche Buslast von 10 % bei 100 kBaud Datenübertragungsrate auf dem CAN-Bus vorliegt. Gegenüber dem direkten CAN-Buszugriff müssen die zu übertragenden CAN-Daten mittels der aufgesetzten Protokolle gewandelt, übertragen und auf dem CAN-Busrechner wieder zurückgewandelt werden, bevor sie auf den CAN-Bus gelegt werden können. Dabei entstehen auf dem Socket Datenraten von 2-6 kByte/s, wobei es zu Verzögerungszeiten von etwa 100-200 ms durch die Wandlung und Weiterleitung kommt. Diese Verzögerung beeinträchtigt die manuelle Steuerung des Fahrzeugs nur minimal, so dass eine gute Handhabung des Fahrzeugs sicher gestellt wird. Grundsätzlich läuft die Bedienoberfläche bei Netzwerkbetrieb etwas langsamer, da bei der Datenanforderung auf die Quittung bzw. das Eintreffen der Antwort gewartet wird.

Generell kann es jedoch zu Problemen kommen, da die Möglichkeit besteht, dass ein Netzwerk, je nach Auslastungsgrad, nicht zu jedem Zeitpunkt die geforderte Bandbreite liefern kann. Dies ist kein Fehler der Steuerungssoftware, sondern stellt eine typische Eigenschaft des Netzes dar. Umgangen werden kann dieses Problem nur durch radikale Lösungen, wobei alle übrigen Rechner des Netzwerks vom Netz getrennt, ein Zugriff verhindert oder ein abgetrenntes Netz verwendet wird.

Die implementierte Software stellt damit sicher, dass der Bediener unabhängig vom Standort des Rechners im Netzwerk das System mit ein und derselben Bedienoberfläche handhaben kann und nicht auf verschiedene Versionen angewiesen ist. Dies führt zu einer Verkürzung der Einarbeitungszeit. Die implementierten Software-Mechanismen für eine Fernhantierung über das Netzwerk arbeiten in akzeptablen Bereichen.

Eine Internetfernhandlung über ein Modem ist generell möglich. Die Schwachstelle liegt dabei eindeutig in der geringen Übertragungsrate von max. 56 kBaud über ein normales Modem per Telefonleitung. Dabei ist die Rate für die Übertragung der Steuerbefehle, der Aktualisierung der Bedienoberfläche und der 3D-Visualisierung in der Regel durchaus ausreichend. Allerdings ist die Geschwindigkeit für die Übertragung der Videodaten nur unzureichend, da die Bildrate mit ca. 0,2 Frame/s deutlich zu gering ist, um das Fahrzeug manuell führen zu können. Abhilfe kann hier eine Verkleinerung des Bildes oder der Wechsel von der Farbübertragung auf Schwarz/Weiß-Übertragung brin-

gen, was aber beides die Orientierung im Arbeitsraum deutlich erschwert. Ein reaktionsschnelles Handhaben kann damit bei rein manueller Führung im Modem-Betrieb nicht sichergestellt werden.

#### 4.5.2 Testszzenarien für Fernhantierung und Fernüberwachung

Für den Netzwerkbetrieb wurden verschiedene Funktionen implementiert und getestet. Es handelt sich dabei sowohl um eine reine Fernüberwachung als auch um die Durchführung von Fernhantierungen eines oder auch mehrerer Bediener im Verbund.

Für eine reine Fernüberwachung können die 3D-Visualisierung und die Videobilder gleichzeitig innerhalb einer Webseite dargestellt werden (Abb. 54). Die VRML-Visualisierung und das zur Kommunikation erforderliche Applet befinden sich auf der linken, die Live-Videobilder auf dem rechten Teil der Seite. Der Bediener erhält damit sowohl eine Unterstützung durch die virtuellen Modelle als auch durch die realen Bilder der externen Kamera, die den Arbeitsraum beobachtet, und einer internen Kamera des Stereokamerasystems.

Über solche oder ähnlich gestaltete Webseiten kann sowohl eine Überwachung über das LAN als auch über das Internet (Modem) erfolgen. Oft ist es hilfreich, wenn überhaupt ein Videobild des Fahrzeugs bzw. der Arbeitsumgebung vorliegt, wobei auf hohe Bildübertragungsraten verzichtet werden kann.

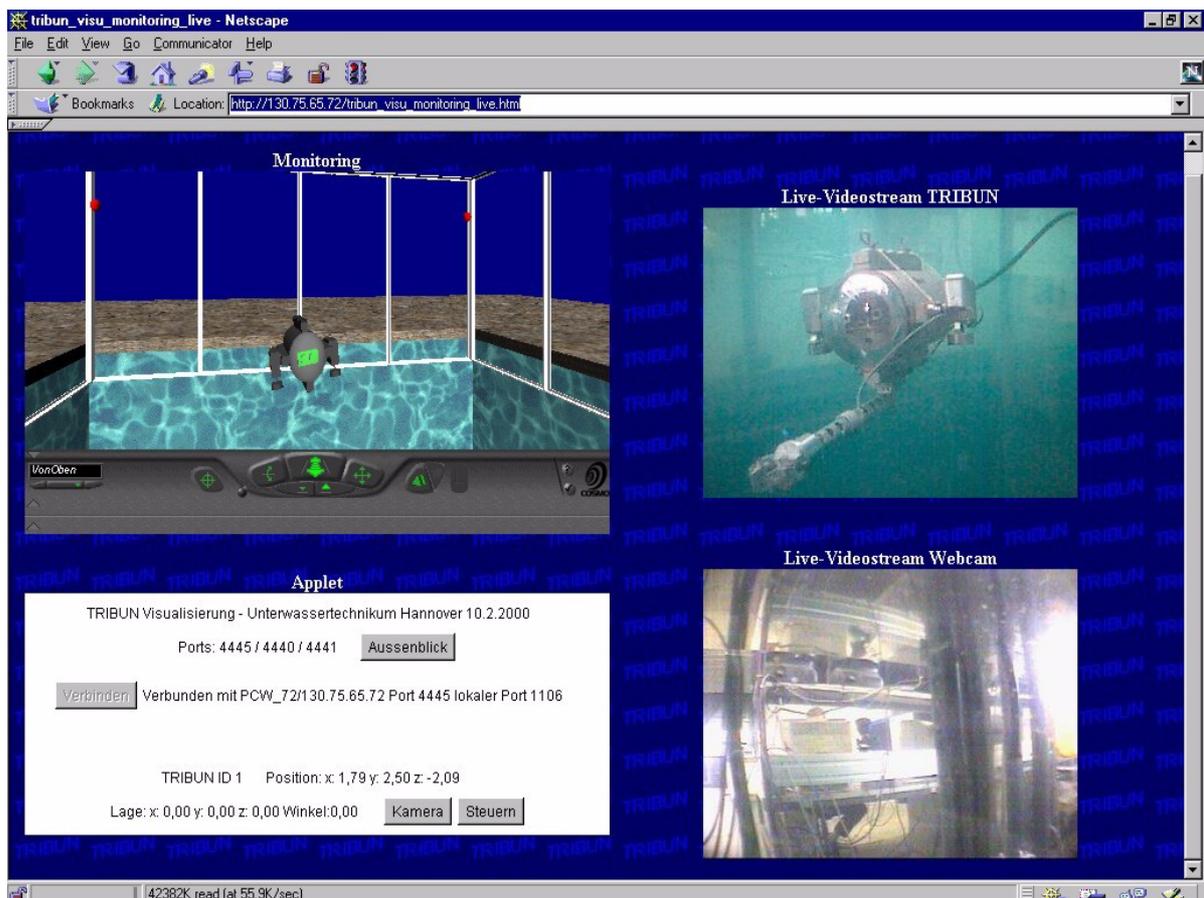


Abb. 54: Darstellung von 3D-Visualisierung, Live-Bilder und Applet im Webbrowser

Für die Durchführung von komplexen Arbeitsaufgaben kann eine Aufteilung der Arbeitsaufgaben auf mehrere Bediener sinnvoll sein. So ist beispielsweise eine Trennung der Steuerung der Fahrzeugbewegung und der eines Werkzeuges denkbar. Während ein Bediener das Fahrzeug im Arbeitsraum positioniert, führt der zweite Bediener die eigentliche Aufgabe aus. Dies kann beispielsweise das Greifen eines Objektes sein,

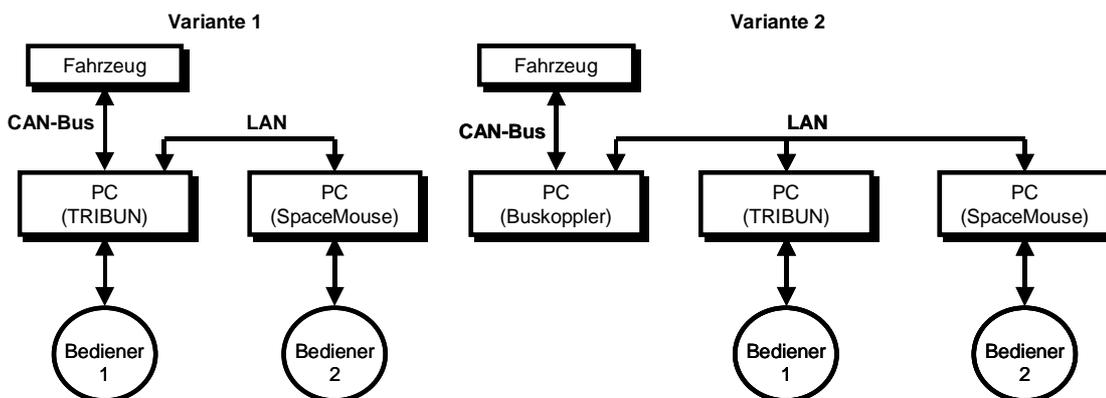


Abb. 55: Varianten für einen Verbundbetrieb mit zwei Bedienern

wobei er den Greifer und Manipulator steuert sowie den Kamerablickwinkel ändert. Für einen solchen Verbundbetrieb wird die SpaceMouse über eine Netzwerkverbindung mittels der Applikation *COMSock* angebunden (Abb. 55). In beiden Fällen steuert Bediener 2 das Fahrzeug, während Bediener 1 über die normale TRIBUN-Bedienoberfläche verfügt und die eigentliche Aufgabe ausführt. In Variante 1 kann Bediener 1 dabei direkt die Videobilder mittels der Monitore bzw. des HMD ansehen, während der Bediener 2 auf eine der oben beschriebenen Videonetzwerklösungen zurückgreifen muss. Bei Variante zwei ist dies für beide Bediener erforderlich. Verstärkte Bedeutung gewinnt hierbei die 3D-Visualisierung, da die Qualität der übertragenen Videobilder schlechter ist als wenn diese direkt auf einem Monitor dargestellt werden.

Beide beschriebenen Varianten wurden erfolgreich getestet und arbeiten fehlerfrei. Wenn beide Bediener zusätzlich räumlich getrennt sind, ist in der Regel eine Audioverbindung notwendig, damit sie ihre Aktionen absprechen können. Für einen solchen Fall kann natürlich ein übliche Telefonverbindung aber auch die oben angesprochene Internet-Telefoniesoftware zum Einsatz kommen.

Wie dieses einfache Beispiel zeigt, kann mit relativ einfach zu integrierenden, netzwerk-basierten Lösungen ein Arbeiten im Verbund von mehreren Bedienern realisiert werden. Sollten weitere Aufteilungen sinnvoll oder notwendig sein, lässt sich das TRIBUN-Steuerprogramm problemlos mit weiteren Socket-Schnittstellen ausrüsten.

Die erstellte Schnittstelle *COMSock* erlaubt auch eine direkte Ansteuerung mittels eines Webbrowsers. Es kann auf Basis der Formular- und Javascript-Funktionalität schnell ein Bedienprogramm im Webbrowser erstellt werden, das z. B. für Testzwecke genutzt werden kann. Eine Kompilierung ist dabei nicht notwendig, da die HTML-Seite direkt mit einem einfachen Texteditor erstellt und mittels des Javascript-Interpreters des Browsers ausgeführt werden kann. Damit ist z. B. auch eine einfache Parametrisierung über einen Webbrowser realisierbar.

## 4.6 Regelungstechnische Bedienerunterstützung

Obwohl entsprechende Echtzeitaufsätze fehlen, lassen sich über die implementierten Windows-Timerfunktionen auch regelungstechnische Vorgänge realisieren.

Zur Erhöhung des Autonomiegrades und damit zur Entlastung des Bedieners wurde eine Tiefenregelung implementiert. Dabei übernimmt die Tiefenregelung die Ansteuerung der Propeller für die Tauchtiefe. Alle anderen Freiheitsgrade sind von der Regelung nicht betroffen, so dass diese manuell gesteuert werden können. In umgekehrter Weise wird der Zugriff der SpaceMouse entsprechend deaktiviert, die normalerweise zur Steuerung der Tauchtiefe dient. Der Bediener ist damit in der Lage, sich innerhalb einer frei wählbaren Arbeitstiefe zu bewegen. Die Tiefenregelung arbeitet bei einer maximalen Tauchtiefe von 50 m auf 10 Zentimeter genau. Ein mögliches Einsatzszenario ist dabei z. B. die optische Inspektion einer waagrecht verlaufenden Schweißnaht. Die Entlastung ist gerade bei diesem Freiheitsgrad spürbar, da durch den Restauftrieb permanent störende Kräfte auf das Fahrzeug eingebracht werden, die eine Nachregelung notwendig machen.

Beim Greifen und Bergen von Kleinteilen kommt es zwangsläufig zu einer Verlagerung des Schwerpunktes, was zum Nicken des Fahrzeugs führt. Orientiert sich der Bediener an seiner Umgebung, ist die veränderte Lage störend, da der Blickwinkel ebenfalls verändert wird. Es besteht daher die Möglichkeit, den Vertikal-Winkel der Kamera mit dem Lagesensor zu koppeln, wobei die voreingestellte Lage immer gemäß der Schräglage korrigiert wird. Die Regelung arbeitet natürlich nur im Bewegungsbereich der Kamera, so dass nicht beliebige Schrägstellungen ausgeglichen werden können.

Die Führungsmaschine kann wahlweise manuell oder automatisch betrieben werden. Für den automatischen Betrieb verfügt sie über die Möglichkeit, Ablaufprogramme abzuarbeiten, die auch Offline erstellt werden können.

## 4.7 Systemüberblick

Nachfolgend werden noch einmal die Komponenten des Gesamtsystems sowie deren Verbindung untereinander in einer Übersicht dargestellt, die innerhalb dieser Arbeit entwickelt und realisiert wurden (Abb. 56).

Die entwickelten Geräte umfassen den TRIBUN-Grundkörper mit Multifunktionsleisten, die Energieversorgung mit einer Kabelverbindung, ein Stereokamerasystem mit Steuereinheit (Head-Tracker), einen Lagesensor, vier Propellerantriebe, zwei Scheinwerfermodule, einen Greifer, eine Unterwasserführungsmaschine mit Sauggreifern, einen 3-Achsenmanipulator sowie ein Handbediengerät.

Die erstellte Software umfasst das TRIBUN-Steuerprogramm, die beiden Buskoppler *CANSock* und *COMSock* für die Netzwerkfernhandlung sowie die 3D-Visualisierung *TRIBUNVisu*.

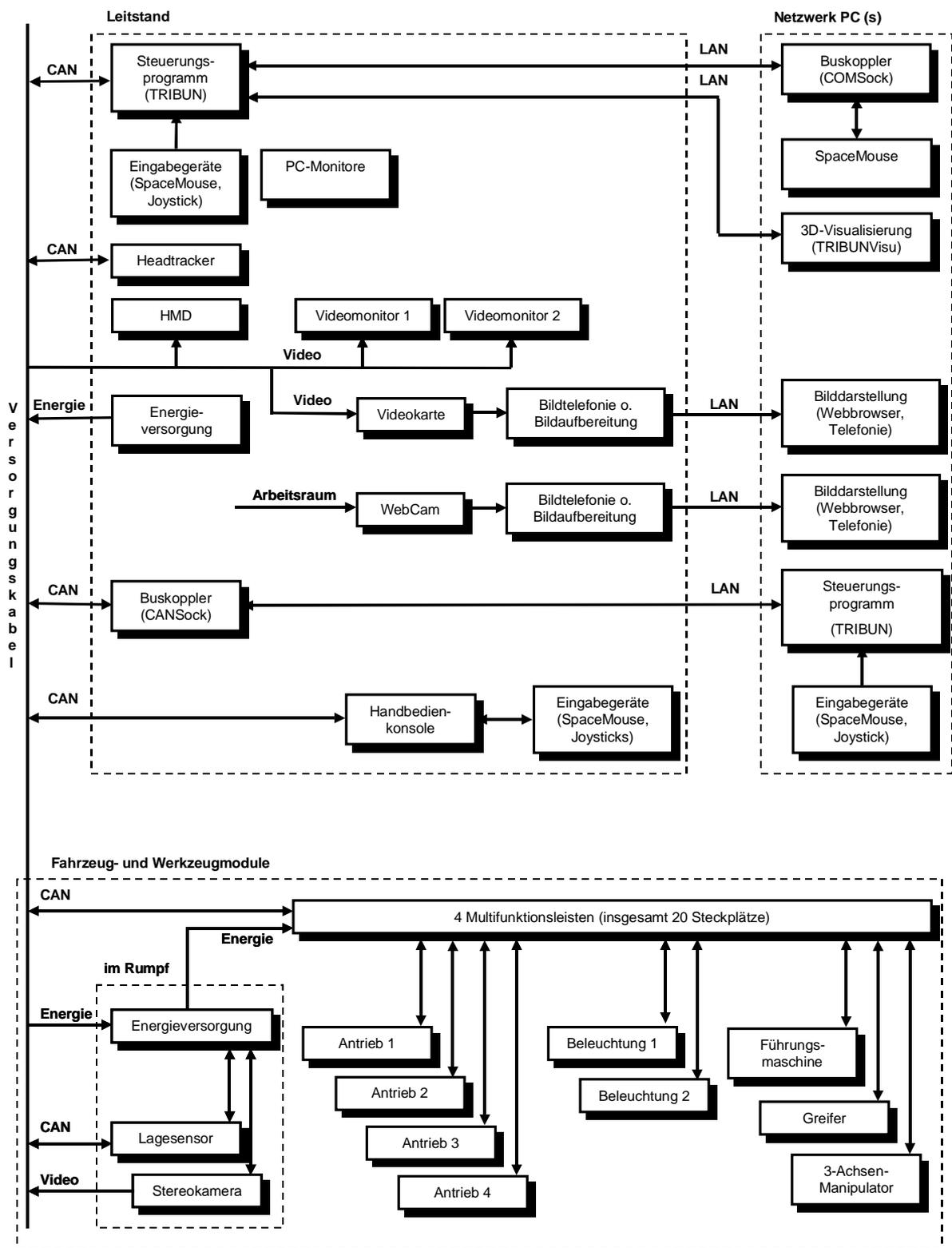


Abb. 56: Gesamtsystem mit Systemverbindungen

## 5 Zusammenfassung

Die Durchführung von Wartungs-, Inspektions-, Reparatur- und Stilllegungsarbeiten in kerntechnischen Anlagen setzt in weiten Bereichen den Einsatz von fernhantierbarer Technik voraus, da das Personal dieser Anlagen nur minimal der radioaktiven Strahlung ausgesetzt werden darf. Für viele Arbeiten werden daher Führungsmaschinen, Industrieroboter oder auch fernhantierbare Werkzeuge eingesetzt, die von den Bedienern vom Leitstand aus gesteuert und kontrolliert werden. Auf Grund des eingeschränkten Aktionsbereichs, der Abmessungen oder der technischen Auslegung dieser Geräte können sie nicht in allen Bereichen zum Einsatz kommen. In diesem Fall finden dann häufig mobile Systemträger Verwendung. Für die Durchführung von Arbeiten im Unterwasserbereich, z. B. innerhalb des Brennelemente-Lagerbeckens, können Unterwasserfahrzeuge eingesetzt werden.

Innerhalb dieser Arbeit wird ein Unterwasserfahrzeug mit der Bezeichnung TRIBUN (Teilautonomes Robotersystem für Inspektion und Bearbeitung im Unterwasserbereich) konzipiert und realisiert. TRIBUN stellt eine Kombination aus universellem Trägerfahrzeug, der als Werkzeugträger fungiert, und einer Vielzahl von adaptierbaren Funktionsmodulen dar. Die entscheidende Neuerung dieses Ansatzes ist es, dem Endanwender eine einfache und hochflexible Handhabungstechnik zur Verfügung zu stellen, die von ihm auf Basis der aktuellen Gegebenheiten am Einsatzort für die unterschiedlichsten Einsatzfälle konfiguriert werden kann. Gegenüber am Markt verfügbaren Geräten stellt dieses Fahrzeug mit seinen Funktionsmodulen damit eine Art Baukastensystem für die Unterwasser-Handhabungstechnik dar.

Eine Verlagerung der Konfiguration vom Hersteller zum Endanwender setzt spezielle Maßnahmen für eine bedienerfreundliche Inbetriebnahme voraus. Dies erfolgt durch die Verwendung von Mechanismen, wie sie aus der PC-Plug&Play-Technik bekannt sind, für die Handhabungstechnik. Damit werden kurze Rüstzeiten und geringe Wartungszeiten gewährleistet und der Austausch defekter Module erleichtert. Die vier am Trägerfahrzeug angebrachten Steckerleisten erlauben Aktoren und Sensoren auch während des Betriebes zu integrieren. Dies wird zum einen durch den Einsatz eines industriellen Feldbussystems, dem echtzeit- und multimasterfähigen, verdrahtungsunaufwendigen CAN-Bus, erreicht. Ein vollständiges Modul besteht dabei aus dem Aktor bzw. Sensor, der Leistungselektronik, der mikrocontroller-basierten Steuerung sowie der Busanbindung. Zur Adaption von Funktionsmodulen an den Steckerleisten stehen 20 einheitliche elektromechanische Steckplätze zur Verfügung.

Nach der Konfiguration des Systems durch den Bediener erkennt dieses selbständig die angeschlossenen Module und stellt die verfügbare Funktionalität bereit. Auf Grund der Verwendung bedienerfreundlicher und ergonomischer Mensch-Maschine-Schnittstellen, wie sie im Bereich der virtuellen Realität zum Einsatz kommen (Head-Mounted-Display, SpaceMouse, Headtracker), lässt sich das Fahrzeug intuitiv steuern, was wiederum die Zeit für ein Bedienertraining reduziert. Neben einem rein manuellen Betrieb wurde das System mit teilautonomen Funktionen ausgerüstet, die den Bediener bei seinem Einsatz entlasten. Zu diesen gehören z. B. die automatische Steuerung des Stereokamerasystems, der Tauchtiefe oder auch der Führungsmaschine.

Die fernhantierte Durchführung von komplexen Handhabungsaufgaben stellt hohe Anforderungen an den Bediener. Da Videobilder, die das Fahrzeug im Arbeitsraum zeigen, im Allgemeinen nicht oder nur mit Einschränkungen vorliegen, kann sich der Bedie-

ner oft nur mit Hilfe der Bordkamera orientieren. Daher wurde zur Unterstützung des Bedienpersonals eine 3D-Visualisierungs-Software entwickelt. Sie basiert auf der Internet 3D-Standard VRML97 und gestattet virtuelle Welten aus Teilwelten zu modellieren und zu animieren. Die interaktive Software kann damit zur Darstellung des Unterwasserfahrzeugs in der aktuellen Konfiguration im Arbeitsraum sowie für die Visualisierung von Roboterkinematiken genutzt werden. Im Zusammenspiel mit einer Software zur Bewegungs- und Handhabungssimulation kann sie auch für ein Bedienertraining verwendet werden. Der Einsatz der Software bietet sich überall dort an, wo Profisysteme zu kostenintensiv und deren meist hohe Funktionalität nicht erforderlich ist.

Über die netzwerkfähige Steuerungssoftware kann das Fahrzeug über ein lokales Rechnernetz oder auch über das Internet überwacht und auch vollständig fernbedient werden. Dabei gestattet es die Software die Bedienfunktionen bei Bedarf auch auf mehrere Bediener aufzuteilen.

Auf Grund der besonderen Einsatzkriterien müssen für viele Handhabungsaufgaben spezielle Systeme - oft in Stückzahl eins - entwickelt und aufgebaut werden. Baukastenlösungen für die Handhabungstechnik erlauben dabei die Test- und Entwicklungszeit deutlich zu reduzieren. Die hier vorgestellte Lösung zeigt, dass sich die zusätzlichen Kosten für den Einsatz einer Feldbustechnik auch bei kleinbauenden Systeme rechnen, wenn man die dadurch gestiegene Funktionalität insbesondere bei Inbetriebnahme und Wartung oder die leichte Austauschbarkeit von einzelnen Komponenten zwischen verschiedenen Handhabungssystemen berücksichtigt. Die Plug&Play-Funktionalität wurde konsequent auf alle Bereiche des Handhabungssystems wie die konstruktive Auslegung, die Steuerungstechnik und Architektur, die Vernetzbarkeit und die 3D-Visualisierung angewandt.

Durch den Einsatz der hier vorgestellten Technik für die Steuerung und die 3D-Visualisierung können zukünftig die Entwicklungskosten und -zeiten bei Entwurf und Erprobung von Prototypen deutlich reduziert und die Funktionalität gleichzeitig gesteigert werden.

Die Arbeit zeigt dabei auch wie verfügbare kostengünstige Technik direkt genutzt bzw. für den gewünschten Einsatzfall modifiziert und erweitert werden kann. Viele Lösungen basieren dabei auf Netzwerk- und Internettechnologien, die sich in den letzten Jahren extrem schnell und vielseitig entwickelt haben.

## 6 Literatur

- /1/ Schraft, R. D.; Schmierer, G.  
Serviceroboter  
Springer-Verlag, Berlin u. a., 1998
- /2/ Haferkamp, H.; Bach, Fr.-W. ; Hahn, M.; Rose, N.  
Unterwasserhandhabungssystem für Meß- und Schneidarbeiten bei der  
Demontage stillgelegter kerntechnischer Anlagen  
BMFT 02 S 7329 5, Abschlussbericht, Institut für Werkstoffkunde Hannover,  
1992
- /3/ Haferkamp, H.; Bach, Fr.-W. ; Seevers, J.  
Mobile Roboter für kerntechnische Anlagen  
ATW Atomwirtschaft - Atomtechnik, Bonn, Bd. 40, Heft 10, S. 638-639, 1995
- /4/ Haferkamp, H.; Bach, Fr.-W.; Reinhardt, U.; Bachmann, M.  
Ferngesteuerte Unterwasserfahrzeuge für Inspektionen  
ATW Atomwirtschaft - Atomtechnik, Bonn, Bd. 42, Heft 10, S. 633-635, 1997
- /5/ Schreck, Goswin  
Aspekte zum Rückbau kerntechnischer Anlagen unter Einsatz fernbedienter  
Unterwasser-Demontagetekniken  
Dissertation, Institut für Werkstoffkunde, Universität Hannover  
Fortschrittberichte VDI, Reihe 6, Nr. 406, VDI-Verlag, Düsseldorf, 1998
- /6/ Leushacke, D.-F.  
Zerlegetechniken für metallische und nichtmetallische Strukturen, Stilllegung  
und Rückbau kerntechnischer Anlagen  
Vorlesungsskript, Universität Hannover, Teil I (Bd. 1-7) und Teil II (Bd. 1-5),  
Energiewerke Nord, Greifswald 1997/98
- /7/ Herbst, Christian  
Roboter in der Sicherheits- und Kerntechnik  
Spektrum der Wissenschaft, Spektrum Akademischer Verlag, Heidelberg,  
Heft 4/1998, S. 36-38
- /8/ Tillery, Brian  
EC's Telemat programme gathers momentum  
Nuclear Engineering International, Bd. 37, Heft 452, S. 35-36, März 1992
- /9/ Diverse Autoren  
Remotely Technology: Products and Services  
Nuclear Engineering International, Bd. 37, Heft 452, S. 39-43, März 1992
- /10/ Gerriets, Wesley  
TROD cleans up at Nine Mile Point  
Nuclear Engineering International, Bd. 37, Heft 452, S. 32-34, März 1992
- /11/ Weber, Patricia J.; Vaneck, Carl W.  
SIMON combines radiation hardness with computer power  
Nuclear Engineering International, Bd. 35, Heft 430, S. 34-38, Mai 1990

- /12/ Banner, Joachim; Till, Walter  
Sensory-guided CATROB speeds up remote maintance  
Nuclear Engineering International, Bd. 35, Heft 430, S. 38-40, Mai 1990
- /13/ Brillowski, K.  
Struktur und Koppelmechanismen von Handhabungsgeräten für extreme Umgebungsbedingungen  
Dissertation, Institut für Fertigungstechnik und spanende Werkzeugmaschinen, Universität Hannover  
Fortschrittberichte VDI, Reihe 1, Nr. 305, VDI-Verlag, Düsseldorf, 1998
- /14/ Padadimitriou, J.  
Steuerungskonzepte für autonome mobile Unterwasser-Roboter  
Dissertation, Institut für Fertigungstechnik und spanende Werkzeugmaschinen, Universität Hannover  
Fortschrittberichte VDI, Reihe 8, Nr. 258, VDI-Verlag, Düsseldorf, 1991
- /15/ Reinhardt, Ulf  
Beitrag zur Simulation modular aufgebauter Unterwasser-Handhabungssysteme  
Dissertation, Institut für Werkstoffkunde, Universität Hannover  
Fortschrittberichte VDI, Reihe 2, Nr. 560, VDI-Verlag, Düsseldorf, 2000
- /16/ Pedersen, S.  
Cyberbase: The man machine computer  
Interact 95, Human-Computer Interaction, Oslo (N), S. 368-371, März 1995
- /17/ National Oilwell Hitec  
Produktinformation ROV, Leistand: HiRov3000, Cyberbase  
<http://www.hitec.no/>, 2001
- /18/ Given, D.  
ROV Review, 1993-1994  
Waves, Windate Enterprises Inc., Spring Valley (CA), 1994
- /19/ Mariscope Meerestechnik  
Produktinformation ROVs: SPY, STAR, FO  
<http://www.mariscope.de/>, 2001
- /20/ HYDROVISION LIMITED  
Produktinformation ROVs: Hyball, Offshore Hyball, Seaeye  
<http://www.hydrovision.co.uk/>, 2001
- /21/ Deep Ocean Engineering  
Produktinformation ROVs: FIREFLY, PHANTOM150, DRAGONFLY  
<http://www.deepocean.com/>, 2001
- /22/ Inuktun Services Ltd.  
Produktinformation ROV: Scallop  
<http://www.inuktun.com/>, 2001
- /23/ Q.I. Incorporated  
Produktinformation ROV: FM-3100  
<http://qi-inc.com/index.htm>, 2001

- /24/ Benthos, Inc.  
Produktinformation ROV: E-ROV, MiniROVER MK II  
<http://www.benthos.com>, 2001
- /25/ J. W. Fishers Mfg. Inc.  
Produktinformation ROV: SEAOTTER  
<http://jwfishers.com/>, 2001
- /26/ Ronam, H. T.  
Robots in PSE&G Nuclear power plants - Experience an future projections  
40th Conference on remote system technologies, Volume 2, 1992
- /27/ Rizzo, Joel; (Benthos, Inc.)  
Small underwater/land robotic vehicles and related technology  
39th Conference on remote system technologies, S. 10-15, November 1991
- /28/ Bauman's Moscow State Technical University  
Research Institute of Special Machinery  
Department of Special Machinery and ROVs  
Übersicht: Unterwasserfahrzeuge  
[http://aqua.sm.bmstu.ru/fun/wordgalery/wordgalery\\_e.htm](http://aqua.sm.bmstu.ru/fun/wordgalery/wordgalery_e.htm), 2001
- /29/ Rose, Nils  
Entwicklung einer Handhabungstechnologie zur Durchführung von Unterwasser-Inspektions- und Zerlegearbeiten  
Dissertation, Institut für Werkstoffkunde, Universität Hannover  
Fortschrittberichte VDI, Reihe 2, Nr. 319, VDI-Verlag, Düsseldorf, 1994
- /30/ Hahn, Michael  
Steuerungstechnologie für modulare, kabelgeführte Unterwasser-Handhabungssysteme  
Dissertation, Institut für Werkstoffkunde, Universität Hannover  
Fortschrittberichte VDI, Reihe Nr.2, VDI-Verlag, Düsseldorf, 1997
- /31/ Universität Hannover  
Sonderforschungsbereich 264, Automatisierte Fertigung unter Wasser  
Arbeits- und Ergebnisbericht 1995-1996-1997, S. 13-16, April 1997
- /32/ Haferkamp, H.; Schreck, G.; Reinhardt, U.  
Teilautomomes Robotersystem für Inspektion und Bearbeitung im Unterwasserbereich  
Die Profitaucher, Heft 5/1998, S.54-55
- /33/ Bachmann, Marc  
Beitrag zur Verbesserung der Genauigkeitskenngrößen von Großmanipulatoren im Mehrmedieneinsatz mit Hilfe der digitalen Photogrammetrie  
Dissertation, Institut für Werkstoffkunde, Universität Hannover, Januar 2000
- /34/ Dubbel; Beitz, W.; Grote, K.-H.  
Taschenbuch für den Maschinenbau  
Springer-Verlag, Berlin u. a., 19. Aufl. 1997

- /35/ Decker, Karl-Heinz  
Maschinenelemente - Gestaltung und Berechnung  
Carl Hanser-Verlag, München u. a., 12. Aufl. 1997
- /36/ Tietze U., Schenck Ch.  
Halbleiterschaltungstechnik  
Springer-Verlag, Berlin u. a., 10. Aufl. 1993
- /37/ Kühn, Eberhard  
Handbuch TTL- und CMOS-Schaltungen  
Hüthig-Buch-Verlag, Heidelberg, 4. Aufl. 1993
- /38/ Seinsch, Hans Otto  
Grundlagen elektrischer Maschinen und Antriebe  
B.G. Teubner-Verlag, Stuttgart, 1988
- /39/ Schön, Felix  
Mit Schrittmotoren steuern, regeln und antreiben  
Franzis-Verlag, Poing, 2. Aufl. 1996
- /40/ Isermann, Rolf  
Digitale Regelsysteme, Band I  
Springer-Verlag, Berlin u. a., 1988
- /41/ Isermann, Rolf  
Digitale Regelsysteme, Band II  
Springer-Verlag, Berlin u. a., 2. Aufl. 1988
- /42/ Föllinger, Otto  
Regelungstechnik  
Hüthig-Buch-Verlag, Heidelberg, 8. Aufl. 1994
- /43/ Alte, R.; Matthiesen, H.  
Schiffbau kurzgefaßt  
Schiffahrts-Verlag „Hansa“, Hamburg, 2. Aufl. 1980
- /44/ Bronstein, I. N.; Semendjajew, K. A.  
Teubner-Taschenbuch der Mathematik  
Verlag Harri Deutsch, Thun u. Frankfurt/Main, 23. Aufl. 1987
- /45/ Weck, Manfred  
Werkzeugmaschine, Fertigungssysteme Band 3.1, Automatisierung und  
Steuerungstechnik  
VDI-Verlag, Berlin u. a., 1995
- /46/ Schreff, Birgit; Bartels, Jan  
Automatisierungskonzepte im Vergleich  
Elektronik, WEKA Fachzeitschriften-Verlag, Poing, Heft 11/1999, S. 50-60
- /47/ Lawrenz, Wolfhard  
CAN Controller Area Network, Grundlagen und Praxis  
Hüthig-Verlag, Heidelberg, 1997

- /48/ Bonfig, Karl Walter  
Feldbus-Systeme  
Reihe Kontakt & Studium; Band 374  
expert-Verlag, Heidelberg, 1992
- /49/ Kernigham; Ritchie  
Programmieren in C  
Carl Hanser-Verlag, München u. a., 2. Aufl. 1990
- /50/ Sontheim Industrie Elektronik  
Manual für den Gebrauch des IBS-Scheckkartenrechners mc68 mit CAN, SIE  
GmbH 1996  
<http://www.s-i-e.de>, 2001
- /51/ CAN in Automation (CiA)  
CANopen Device Profile for I/O Modules, Version 1.04, Dezember 1996
- /52/ CAN in Automation (CiA)  
CANopen Communication Profile based on CAL, Version 3.0, Dezember 1996
- /53/ CAN in Automation (CiA)  
CAL: CAN Application Layer for Industrial Applications, Version 1.1, Februar  
1996
- /54/ Sontheim Industrie Elektronik  
CAL-kompatibles Kommunikationsprotokoll für mc68, Revision B, 1996
- /55/ Schober, H.  
Das Sehen, Band II  
Fachbuchverlag Leipzig, Leipzig, 1964
- /56/ Mahner, H.-J.  
Komponenten automatisierter Handhabungssysteme für den Einsatz unter  
Wasser  
Dissertation, Institut für Fertigungstechnik und spanende Werkzeugmaschi-  
nen, Universität Hannover  
Fortschrittberichte VDI, Reihe 2, Nr. 545, VDI-Verlag, Düsseldorf, 2000
- /57/ Blaschke, M.; Haferkamp, H.; Tönshoff, H.K.; Bedief, E.; Mahner, H.-J.  
Träger- und Handhabungssysteme  
Tagungsband VI. Stilllegungskolloquium Hannover und 5. Statusbericht Stillle-  
gung und Rückbau kerntechnischer Anlagen, S. 7-21  
PtWT+E, April 2000
- /58/ Bartenschläger, J.; Hebel, H.; Schmidt, G.  
Handhabungstechnik mit Robotertechnik  
Vieweg-Verlag, Braunschweig/Wiesbaden, 1998
- /59/ Gesellschaft für Anwender und Systemsoftware  
Produktinformation 6-Achsen-Joystick: Spacemaster  
<http://www.basys-gmbh.de>, 2001

- /60/ Labtec, Inc. Corporate Headquarter  
<http://www.labtec.com/>, 2001  
Produktinformation 6-Achsen-Joystick: Spaceball  
<http://www.labtec.com/product/family.cfm?CategoryID=4>, 2001
- /61/ LogiCad3D GmbH  
Produktionformation 6-Achsen-Joystick: SpaceMouse  
<http://www.spacemouse.com>, 2001
- /62/ Okoshi, T.  
Three dimensional imaging techniques  
Academic Press, New York, 1976
- /63/ Teittinen, Marko  
The Encyclopedia of Virtual Environments: Depth Cues in the Human Visual System  
<http://www.hitl.washington.edu/scivw/eve/index.html>, 1993
- /64/ Arthur Zwern  
How To Select The Right Head-Mounted-Display  
<http://kohinoor.csa.iisc.ernet.in/~pritha/howtochoose.html>, 2001
- /65/ Lanc, Corde  
The Encyclopedia of Virtual Enviroments: Displays  
<http://www.hitl.washington.edu/scivw/eve/index.html>, 1993
- /66/ Sperlich, Tom  
Welt im Helm  
c't, Heinz Heise-Verlag, Hannover, Heft 11/1995, S. 200-202
- /67/ Post, Hilde-Josephine  
Cybersick im Cyberspace  
c't, Heinz Heise-Verlag, Hannover, Heft 11/1995, S. 203-205
- /68/ Bauer, Christian  
Ein Head-Mounted-Display der Superlative und Übersicht: Head-Mounted-Displays  
c't, Heinz Heise-Verlag, Hannover, Heft 11/1995, S. 206-208
- /69/ Olympus Optical Co. (Europa) GmbH  
Produktinformation Videobrille: Eyetrek  
<http://www.eye-trek.de/>, 2001
- /70/ Booch, Grady  
Objektorientierte Analyse und Design  
Addison-Wesley, Bonn u. a., 2. Aufl. 1995
- /71/ Burkhardt, Rainer  
UML, Unified modeling language, Objektorientierte Modellierung für die Praxis  
Addison-Wesley, Bonn, 1. Aufl. 1997
- /72/ Peak  
Produktinformation PC-CAN-Schnittstelle: CAN-Dongle  
<http://www.peak-system.com/>

- /73/ Borland C++, Version 5  
tewi-Verlag, München, 1996
- /74/ Wlotka, Dieter W.  
Robotersimulation  
Springer-Verlag, Berlin u. a., 1991
- /75/ Tecnomatix Technologies Ltd.  
Produktinformation 3D-Robotersimulation: ROBCAD  
<http://www.Tecnomatix.com/>, 2001
- /76/ Fa. CAMELOT  
Produktinformation 3D-Robotersimulation: ROBSIM  
<http://www.camelot.dk/>, 2001
- /77/ Schmid, Dietmar  
Virtuelle und telematische Roboter  
Spektrum der Wissenschaft, Spektrum der Wissenschaft Verlagsgesellschaft mbH, Heidelberg, Heft 4/1998, S. 32-35
- /78/ Miss, R.-W.; Böke, M.; Deeg, H.-J.  
PUSSY - Ein Programm für die Offline-Programmierung und grafische Simulation von Robotereinsätzen  
Forschungsberichte GKSS (Gesellschaft für Kernenergieverwertung in Schiffbau und Schifffahrt), Report-Nr. GKSS 94/E/72, S. 1-97, 1994
- /79/ Forschungszentrum Karlsruhe  
Produktinformation 3D-Simulations-Software: KISMET  
<http://iregt1.iai.fzk.de/>, 2001
- /80/ Silicon Graphics Inc.  
Produktinformation Grafikstandard: OpenGL  
<http://www.SGI.com/software/opengl/>, 2001
- /81/ Kloss, J.; Rockwell, R.; Szabo, L.; Duchrow, M.  
VRML97, Der neue Standard für interaktive 3D-Welten im World Wide Web  
Addison Wesley, Bonn u. a., 1. Aufl. 1998
- /82/ Web 3D Consortium  
The Virtual Reality Modeling Language, Version 1.0, November 1995  
<http://www.web3d.org/technicalinfo/specifications/vrml1.0.htm>, 2001
- /83/ Silicon Graphics Inc.  
Produktinformation Grafikstandard: Open Inventor  
<http://www.europe.sgi.com/Technology/Inventor/>, 2001
- /84/ Web 3D Consortium  
The Virtual Reality Modeling Language, VRML97  
International Standard ISO/IEC 14772-1:1997  
<http://www.web3d.org/Specifications/VRML97>, 2001
- /85/ Web 3D Consortium  
The External Authoring Interface  
<http://www.web3d.org/WorkingGroups/vrml-eai/>, 2001

- /86/ Campione, M.; Walrath, K.  
Das Java Tutorial  
Addison-Wesley, Bonn u. a., 1. Aufl. 1997
- /87/ Beier, K.-P.  
Web-based virtual reality in design and manufacturing applications  
HANSA. International Maritime Journal, Bd. 137, Heft 5/2000, S. 42-47
- /88/ Tam, E.-K; Badra F.; Marceau, R.-J.; Marin, M.-A.; Malowany, A.-S.  
A Web-based virtual environment for operator training (for power stations)  
IEEE Transactions on Power Systems, Bd. 14/1999, Heft 3, S. 802-808
- /89/ Xiaohong, Qiu; Fang, Liao  
Visual flight simulation of UAVs in real time programmed in Java language  
Modeling, Simulation and Visualization for Real and Virtual Environments, 7-8  
April 1999, Orlando, FL, USA, Proceedings of the SPIE- The International  
Society for Optical Engineering, Band 3694/1999, S. 194-201
- /90/ Gracanin, D.; Matijasevic, K.; Tsourveloudis, N.-C.; Valavanis, K.-P.  
Virtual reality testbed for mobile robots  
ISIE '99, Proceedings of the IEEE International Symposium on Industrial Elec-  
tronics, 12-16 Juli 1999, Bled, Slovenia, Band 1/1999, S. 293-297
- /91/ Colon, E.; Baudoin, Y.  
Viewing and controlling a mobile robot with common Web technologies  
Enabling Technology for Simulation Science II, 14-16 April 1998, Orlando, FL,  
USA, Proceedings of the SPIE - The International Society for Optical Engi-  
neering, Bd. 3369/1998, S. 278-286
- /92/ Lehmann, P.; Richter K.  
Der Ingenieur im Internet, neue Chancen durch Engineering-Netze  
VDI-Ges. Entwicklung, Konstruktion, Vertrieb, Tagung, Karlsruhe (D), 6.-7.  
Okt, 1997, VDI-Verlag, Düsseldorf, Bd. 1362 (1997), S. 61-79
- /93/ Howe, A.-S.  
A network-based kit-of-parts virtual building system  
CAAD futures 1997, 7th Internat. Conf. on Computer Aided Architectural  
Design Futures, München, 4-6 August 1997, S. 691-706
- /94/ Beier, Ekkehard  
Lücken im Cyberspace  
iX, Heinz Heise-Verlag, Hannover, Heft 3/2000, S.108-114
- /95/ Ammann, Eckhard  
Programmierung animierter Welten, Java, JavaScript und VRML  
International Thomson Publishing, Bonn u. a., 1. Aufl. 1997
- /96/ van Hoff, A.; Shaio, S.; Orca, O.  
Java-Applets erstellen und nutzen  
Addison Wesley, Bonn u. a., 1996
- /97/ Seeboerger-Weichselbaum, Michael  
Javascript  
hbv-Verlag, Kaarst, 1. Aufl. 1999

- /98/ Ramm, Frederik  
Recherchieren und Publizieren im World Wide Web  
Vieweg-Verlag, Braunschweig u. a., 1995
- /99/ Request for Comments (RFC) Editor; Information Sciences Institute, University of Southern California  
RFC 793, Transmission Control Protocol, September 1981  
<ftp://ftp.isi.edu/in-notes/rfc793.txt>, 2001  
<http://www.rfc-editor.org>, 2001
- /100/ Merziger, G.; Wirth, T.  
Repetitorium der höheren Mathematik  
Feldmann-Verlag, Springe, 1991
- /101/ Crispen, Bob  
Mathematische Beschreibung: Rotation von VRML-Objekten  
[http://hiwaay.net/~crispen/vrmlworks/eai\\_faq/rotations.txt](http://hiwaay.net/~crispen/vrmlworks/eai_faq/rotations.txt), 2001
- /102/ Gosling, James; Joy, Bill; Steele, Guy  
The Java Language Specification  
<http://java.sun.com/docs/books/jls/html/index.html>, 2001
- /103/ Sun Microsystems  
Produktinformation Programmiersprache: Java  
<http://java.sun.com>, 2001
- /104/ Request for Comments (RFC) Editor;  
Berners-Lee, T.; Masinter, L.; McCahill, M.  
RFC 1738, Uniform Resource Locators (URL), Dezember 1994  
<ftp://ftp.isi.edu/in-notes/rfc1738.txt>, 2001  
<http://www.rfc-editor.org>, 2001
- /105/ Netscape Communications Corporation  
Produktinformation Internetbrowser: Netscape Communicator  
<http://www.netscape.com>, 2001
- /106/ Microsoft Corporation  
Produktinformation Internetbrowser: Internet Explorer  
<http://www.microsoft.com/>, 2001
- /107/ Cosmo Software, Computer Associates International, Inc.  
Produktinformation VRML-Browser: Cosmoplayer  
<http://www.cai.com/cosmo/>, 2001
- /108/ Blaxxun Interactive  
Produktinformation VRML-Browser: Blaxxun Contact  
<http://www.blaxxun.com>, 2001
- /109/ ParallelGraphics  
Produktinformation VRML-Browser: Cortona VRML client  
<http://www.parallelgraphics.com/products/cortona/>, 2001
- /110/ RIT Research Labs  
Produktinformation Webserver: Tinyweb  
<http://www.ritlabs.com/tinyweb/>, 2001

- /111/ Eyematic Interfaces  
<http://www.eyematic.com/>  
Produktinformation Internettelefonie-Software: iVisit  
<http://www.ivisit.com>, 2001
- /112/ NetSpeak Corporation  
<http://www.netspeak.com/>  
Produktinformation Internettelefonie-Software: Webphone  
<http://www.webphone.com>, 2001
- /113/ Vocaltec  
Produktinformation Internettelefonie-Software: Internet Phone  
<http://www.vocaltec.com>, 2001
- /114/ Universität Ulm, Fakultät Informatik  
Produktinformation: WebVideo  
<http://www-vs.informatik.uni-ulm.de/soft/wv/>, 2001

**Lebenslauf:*****Persönliche Daten***

Name		Matthias Blaschke
geboren am	24.05.1967	in Minden/ Westfalen
Eltern		Gerhard Blaschke Christa Blaschke, geb. Graf
Familienstand		ledig, keine Kinder
Staatsangehörigkeit		deutsch

***Ausbildungsdaten***

Schulausbildung	1973-1977	Grundschule, Minden
	1977-1986	Bessel-Gymnasium, Minden Abschluss: Hochschulreife
Hochschulausbildung	1987-1995	Studium der Elektrotechnik, Universität Hannover, Fachrichtung: Nachrichtenver- arbeitung, Nachrichtentechnik Abschluss: Diplom-Ingenieur

<b><i>Wehrdienst</i></b>	1986-1987	Grundwehrdienst, Hamburg/ Flensburg
--------------------------	-----------	--

<b><i>Berufspraxis</i></b>	1990-1995	Wissenschaftliche Hilfskraft am Institut für Werkstoffkunde, Universität Hannover
	1995-2000	Wissenschaftlicher Mitarbeiter am Institut für Werkstoffkunde, Universität Hannover Fachgruppe: Stilllegungstechnik, Steuerungstechnik und Simulation