

**Dipl.-Ing.**

**Andreas Schulz**



## **Studium**

Technische Universität Berlin, Deutschland

- Diplom der Technische Informatik mit Schwerpunkt Rechnerorganisation und Architektur

## **Berufliche Karriere**

1998 - 2004 Studium der Technischen Informatik an der TU Berlin, Deutschland

2004 - 2008 Evologics GmbH, Berlin, Deutschland

- Soft- und Hardwareentwicklung für Mikrocontrollersteuerung bei den Projekten:
  - Humanoider Muskel-Roboter
  - Aqua Ray

2009 – z.Zt. Wissenschaftlicher Mitarbeiter, Technische Universität Berlin, Deutschland

- Soft- und Hardwareentwicklung für Mikrocontrollersteuerung im Projekt BROMMI

## **Derzeitiger Arbeitgeber / Institution**

Technische Universität Berlin, Fachgebiet Regelungssysteme

Link: [www.control.tu-berlin.de](http://www.control.tu-berlin.de)

E-Mail-Adresse: [schulz@control.tu-berlin.de](mailto:schulz@control.tu-berlin.de)

## Der Humanoide Muskelroboter Torso ZAR5 - Elektronik- und Softwareentwicklung -



Abbildung 1: Humanoider Muskelroboter<sup>1</sup> ZAR 5 (© FESTO AG)

*Andreas Schulz, Technische Universität Berlin, FG Regelungssysteme*

### Zusammenfassung

Derzeit werden dezentrale Steuerungssysteme hauptsächlich in stark modularisierten, homogenen Robotersystemen angewendet. Bei hierarchischen Aufbauten hingegen werden meist zentrale Steuerungen bevorzugt. Die Natur jedoch verfolgt bei höheren Lebewesen immer einen Mischweg. Diese Aufteilung in zentrale und dezentrale Steuerungszentren wurde beim Aufbau eines pneumatisch angetriebenen Roboters, angelehnt an einen menschlichen Torso, verfolgt. Spezieller Schwerpunkt war die Fehlertoleranz, Lastverteilung und hohe Reaktionsgeschwindigkeit ohne Interaktion der Zentraleinheit. Analog zur Begrenzung der biologischen Impulsweiterleitung bei kontinuierlicher Erregungsleitung in marklosen Nervenfasern begrenzt die Trägheit der Luft die Schaltgeschwindigkeit von pneumatischen Aktoren. Der einzige Weg dies zu umgehen ist die aktorennahere Positionierung der Schalteinheiten (Ventile) und damit auch die Einführung von dezentralen Steuerzentren.

Die neuartigen Aktoren und Ansätze werden in Abschnitt eins kurz vorgestellt und die entstehenden Anforderungen werden im Abschnitt zwei zu einem Modell zusammengefasst. Die Realisierung im darauf folgenden Abschnitt wird unter dem Gesichtspunkt der technischen Machbarkeit beschrieben. Ergebnisse und Aussichten sind im letzten Abschnitt zusammengefasst.

<sup>1</sup> Diese Projekt wurde unterstützt durch M. Fischer, TC-D, Corporate Design, FESTO AG & Co. KG

## 1 Einleitung

Heutige technische Problemstellungen fordern zunehmend neue Lösungen, die nur sehr kostenintensiv auf herkömmliche Weise zu bewerkstelligen sind. Aus diesem Grund wird derzeit verstärkt in der Natur nach Ideen gesucht. Neben der Biologie, fließen dabei Erkenntnisse aus der Werkstofftechnik, als auch aus neuen Bereichen wie der Bioinformatik und Neurologie in die Analyse biologischer Vorbilder mit ein. Dadurch werden neue Strukturen, Aktoren und informationsverarbeitende Verfahren entwickelt, die in der bionischen Robotik zusammenlaufen. Typische Aktoren sind zum Beispiel elektroaktive Polymere [1], Form-Gedächtnis-Metalle [2] und pneumatische Muskeln [3,7], die wiederum andere Ansprüche an Mechaniken und Steuerungssysteme stellen, als klassische Systeme.

Im Projekt Humanoider Muskelroboter ZAR (Zwei-Arm-Roboter) [3-6] sorgen fluidische Muskeln der Firma FESTO für die Krafterzeugung. Durch beaufschlagten mit Druckluft verkürzen sie sich und erzeugen so sehr hohe Zugkräfte pro Masse (gewichtbezogene Leistung) im Vergleich zu elektrischen Antrieben. Andererseits werden sie nicht durch elektrische Felder geschaltet sondern mit Hilfe von Druckluft. Die Lokalisierung des Schaltortes ist also deutlich wichtiger, als bei herkömmlichen Systemen. Hinzu kommt, dass die Steuerung von Robotern mit vielen Freiheitsgraden (>20) von einzelnen Rechnern auch mit Mehrkernprozessoren nur bedingt in harter Echtzeit bewerkstelligt werden können. Es muss damit, analog zu komplexen biologischen Systemen, eine Aufteilung erfolgen. Das kann dadurch erreicht werden, dass eine Aufgabe so weit nach unten delegiert wird, wie alle externen Informationen vom unteren Segment ohne hohen Aufwand eingelesen werden können. Dies führt zwangsläufig zu einer Verarbeitung der Daten an dem Ort wo sie entstehen, bzw. dort, wo sie zum ersten Mal in größerem Umfang zusammenfließen. Da Kommunikation immer mit Verzögerungen behaftet ist und diese bei Dezentralisierungen berücksichtigt werden, entstehen so nebenbei halbautonome Steuerungen, welche auch bei kurzzeitigem Ausfall des Kommunikationssystems noch funktionsfähig bleiben.

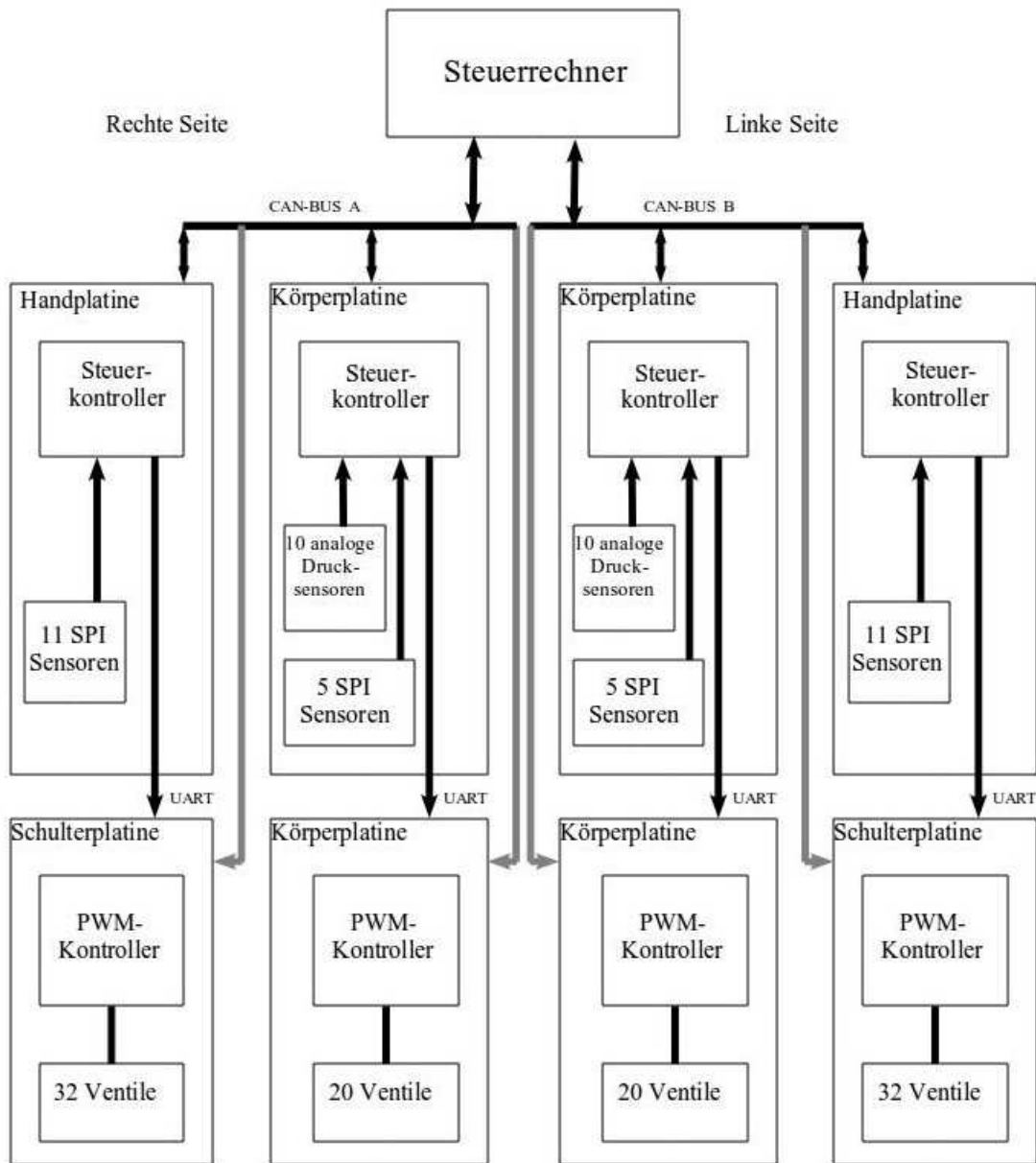


Abbildung 2: Hierarchisches Steuerungsmodell

## 2 Konzeptionierung

Beim ZAR kommt ein hierarchisches dezentrales Steuersystem zur Anwendung. Der zentrale Rechner als Hauptschaltstelle behandelt sämtliche Eingriffe des Benutzers und zeigt den Status des Roboters an. Der Status wird aus den Daten der dezentralen Steuereinheit beider Arme bestimmt. Durch Untersuchungen der Sensordatenflüsse, bot sich eine Unterteilung des Armes in die Hand und den restlichen Arm an. Somit kann auf eine Weiterleitung der Sensordaten über größere Strecken verzichtet werden und die Signalgüte der analogen Sensordaten verbessert werden. Diese Aufteilung muss auch unter dem Gesichtspunkt der zu erwartenden Rechenleistung von Mikrocontrollern betrachtet werden, um eine Überschreitung der Rechenleistung in jedem Fall zu vermeiden. Die Steuereinheiten haben direkten Zugriff auf alle Sensordaten ihres Segmentes. Sie müssen in der Lage sein, die Regelung aller ihnen untergeordneten Gelenke zu verarbeiten und die jeweiligen Ventile für die Muskeln anzusteuern. Um die bewegte Masse klein zu halten, wurden diese Ventile jedoch an der Schulter untergebracht. Somit wird eine zweite Kontrollereinheit (PWM-Kontroller) den jeweiligen Steuermikrokontroller untergeordnet und befindet sich direkt an den Ventilinseln am Torso (Abbildung 2). Da sie vom zentralen Steuerrechner selbst keine Daten benötigen, besteht keine Verbindung zum Zentralrechner.

Zusammenfassend sendet der Zentralrechner Parameter wie Geschwindigkeit, Krafteinsatz, Steifigkeit und Zielposition an die ihm untergeordneten Mikrocontroller. Danach wird die komplette Bestimmung des Ist-Zustandes und die Berechnung zum Erreichen der Zielposition auf der für das jeweilige Gelenk verantwortlichen Steuereinheit autonom durchgeführt. Die berechneten Öffnungszeiten für die Ventile werden dann weiter an die PWM-Kontroller geleitet, die die Ventile direkt ansteuern können. Die Echtzeitbereiche für die Regelalgorithmen und PWM-Generierung werden also nicht nur voneinander, sondern auch von der Benutzerschnittstelle getrennt, und für alle vier Segmente separat durchgeführt. Der Datenaustausch erfolgt je nach Echtzeitanpruch über Puffer oder Empfangs-Interrupts.

Ein weiterer wichtiger Punkt ist die schnelle Sensordatenerfassung. Da aus Platzgründen nicht für jedes Gelenk und jeden Sensor ein separater Mikrokontroller vorgesehen werden kann, müssen die Armkontroller zum Beispiel 15 Sensoren in möglichst kurzer Zeit einlesen, um ausreichend Rechenzeit für die Regelalgorithmen bereit stellen zu können. Hierbei werden digitale Sensoren bevorzugt, da einerseits die Anzahl der Analog/Digital-Wandler (ADC) bei Mikrocontrollern sehr beschränkt ist und andererseits so eine hohe Unabhängigkeit der Signalqualität von Schaltung und Verdrahtung gewahrt werden kann. Meistens ist jedoch auch die Zahl der hardwareseitig unterstützten digitalen Bussysteme begrenzt. Somit bleibt nur die Möglichkeit, in der Elektronik die Kanäle umzuschalten, oder, wie beim ZAR geschehen, ein sehr einfaches serielles Protokoll softwareseitig zu realisieren. Dies erlaubt es, sämtliche digitalen Sensoren parallel einzulesen, solange sie die gleiche Schnittstelle aufweisen. Die Analogsensoren werden jedoch weiterhin über verschiedenen Kanäle eines ADC nacheinander erfasst.

## 3 Realisierung

Als zentraler Rechner wird ein handelsüblicher PC verwendet. Er vereinfacht die Implementierung des Benutzerinterface und beinhaltet keine Einschränkung in den vorhandenen bzw. nachrüstbaren Schnittstellen. Die Software läuft auf einer Standard Windows XP Installation. Den weichen Echtzeitbedingungen genügt diese Basis, da für die Benutzerinteraktion die betriebssystemtypischen

Jiffies (minimale Zeiteinheit für Prozesse) von 10-20 ms ausreichen. Die gegebenen falls nötigen schnellen Reaktionen werden durch Echtzeiteigenschaften der Schnittstellenkarte realisiert. Dies trifft zum Beispiel für Not-Aus und Kollisionen zu. Um diese Echtzeitdaten, trotz normalen Datenverkehrs vom zentralen Steuerrechner, jederzeit senden zu können, muss der Bus echtzeit- und multimasterfähig sein. Dadurch kann auch der Ausfall von einzelnen Komponenten nicht das Gesamtsystem zum Erliegen bringen. Die hohe Informationsdichte und lange Gültigkeitsdauer der zu transportierenden Daten verlangen nach einer sehr guten Fehlererkennung. Hinzu kommt, dass das Kommunikationsprotokoll hardwareseitig von Mikrocontrollern unterstützt werden muss und nicht zuletzt eine ausreichend Bandbreite bietet. Die Wahl fiel auf CAN (controller area network), auch wenn die geringe verfügbare Nutzlast eine Aufteilung in je einen Bus pro jede Seite notwendig macht. Die Kommunikation der Mikrocontroller mit den Sensoren ist dagegen stark von den angebotenen Sensorbausteinen abhängig. Die zu messenden Größen im Roboter sind der Luftdruck in den Muskeln und der Winkel an den Gelenken. Für die letzter Größe wird der kontaktlose, magneto-resistive Sensor KMZ41 der Firma Philips verwendet, weil er mit Hilfe eines Zusatzbausteines UZZ9001 über SPI (serial peripheral interface) ausgelesen werden kann. Die Auswahl an Drucksensoren mit kleinen Abmessungen ist sehr eingeschränkt. Die Wahl fiel auf die analogen XFGM-3 Sensoren der Firma Fujikura. Das Herzstück der dezentralen Steuereinheit ist ein PIC18F458 von Microchip. Er bietet eine CAN Layer 2.0 Schnittstelle, genügend Speicher, 10 MIPS, sowie ausreichend schnelle I/O Pins zur softwareseitigen Realisierung von SPI. Um Synergien zu nutzen wurde der gleiche Mikrocontroller für die PWM-Einheit verwendet. Die Kommunikation zwischen diesen beiden erfolgt über UART, da die Gültigkeit der Daten 20ms nicht überschreitet und somit auf eine Fehlererkennung praktische verzichtet werden kann. An sich hätten die Daten auch mittels CAN übertragen werden können, jedoch ist der Overhead bei Paketen von 2 Bytes unverhältnismäßig groß.

Die Software des Steuerkontrollers besteht aus einem einfachen Echtzeit-Multitasking Betriebssystem. Dieses startet nach einer Initialisierungsphase die Hauptschleife, welche alle 20ms sämtliche laufenden Prozesse bearbeitet (Abbildung 3). Am Anfang werden alle Sensoren eingelesen. Dies spart Zeit, da die Winkelsensoren parallel, über ein softwareseitiges Clock-Signal getaktet, eingelesen werden können. Gleichzeitig werden die Werte der analogen Drucksensoren mit Hilfe des ADC bestimmt, wobei nur zum Schreiben des Ergebnisses und Setzen des neuen Kanals aktiv Rechenzeit verbraucht wird. Danach bearbeitet der Mikrocontroller alle aktiven Regelprozesse. Ankommende CAN Paket werden in dieser Zeit in vorhandene Puffer geschrieben. Nur wenn diese voll sind, werden sie via Interrupt-Programm in einen separaten Zwischenspeicher geschrieben. Einzige Ausnahme von dieser Regel bildet eine spezielle CAN Adresse, welche eine sofortige Bearbeitung erzwingt. Diese wird jedoch nur für die Not-Aus Funktion verwendet. Die berechneten Ventilschaltzeiten werden über nicht blockierendes Senden direkt von den Prozessen an die PWM-Kontroller weitergeleitet. Sind die Regelprozesse abgearbeitet werden zunächst ausstehende Terminierungen vollzogen. Dabei gibt es drei verschiedene Arten. Ersten kann ein Prozess sich selbst terminieren, zweitens kann bei zeitbeschränkten Prozessen die maximale Anzahl der 20 ms Zyklen erreicht sein, wodurch das Betriebssystem diese beendet. Um Prozesse zu beenden, welche eine Position unbegrenzt halten sollen, muss das Gelenk mit einer neuen Funktion überschrieben werden, um die vorherige

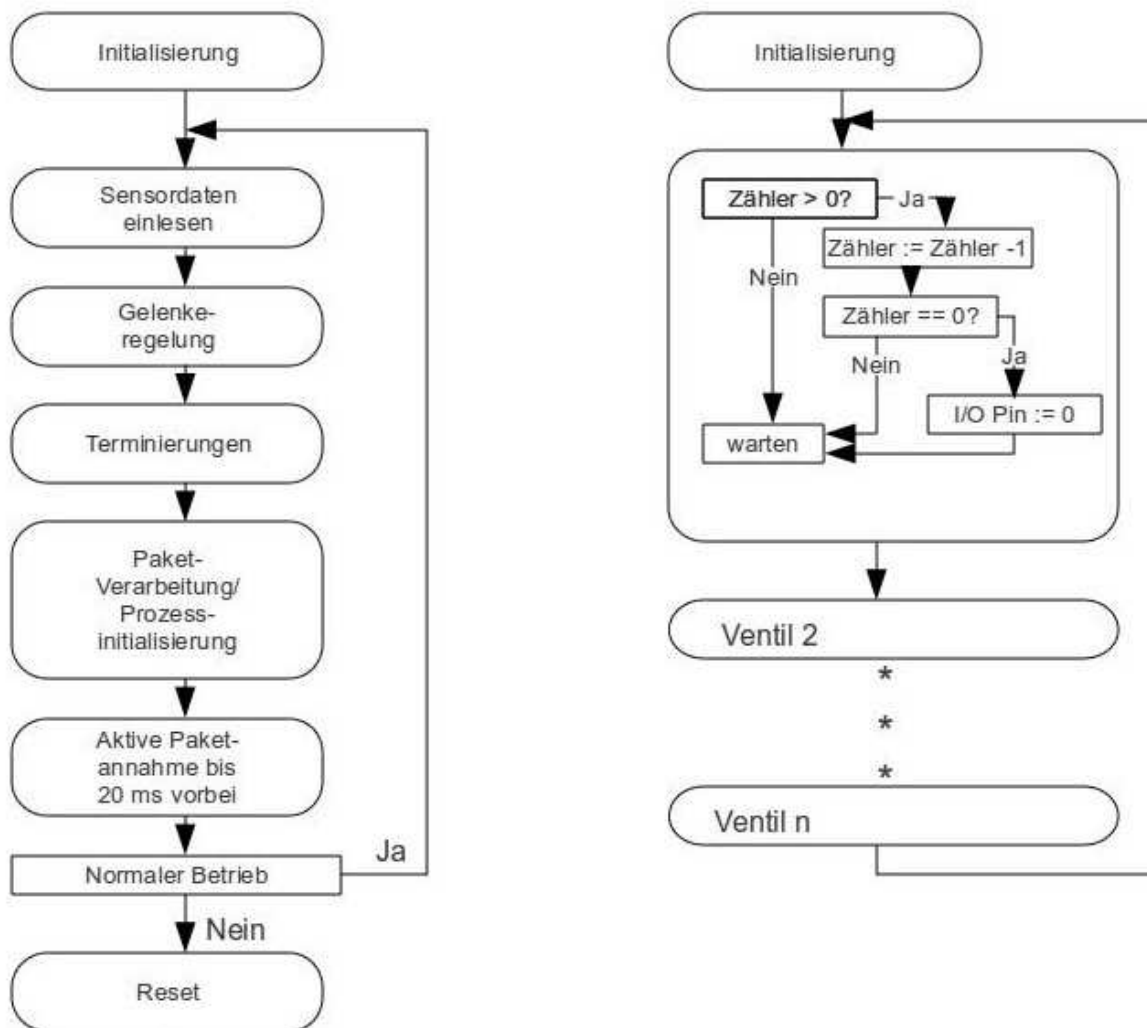


Abbildung 3: Flussdiagramm Steuerkontroller (links) und PWM-Kontroller (rechts)

Funktion zu beenden. Diese Vorgehensweise verhindert auch, dass zwei Regelfunktionen auf demselben Gelenk laufen. Bei einer Terminierung werden auch alle reservierten Speicherbereiche des Prozesses freigegeben. Nach dem Terminierungszeitbereich werden die empfangenen CAN-Pakete verarbeitet. Abfragen der Sensorwerte führen sofort zur Erzeugung eines neuen Paketes mit dem am Anfang ausgelesenen Werten. Pakete die neue Prozesse anlegen, führen zu einer Initialisierung des Prozesstyps und der Eingliederung in die Prozesstabelle für den nächsten Durchlauf. Im Verlauf dieses Prozesses, kann wie oben beschrieben, ein anderer Prozess für die Terminierung markiert werden, wodurch er nicht ein weiteres Mal bearbeitet wird. Die dritte Art von CAN-Paketen verändert die interne Konfiguration des Steuerkontrollers und hat somit Einfluss auf Statusmeldungen und Statistiken. Der verbleibende Rest der Zeitschleife wird durch aktives Pollen und Zwischenspeichern ankommender CAN Pakete verwendet. Sollte diese Zeit vor Erreichen des Terminierungsabschnittes ablaufen, wird ein Fehler generiert. Tritt dies erst bei der Verarbeitung der CAN Daten auf, wird eine Warnung verschickt. Beide Pakettypen haben dabei eine höhere Priorität als normale Datenpakete, um so nicht von einem vollen CAN-Bus behindert zu werden.

Die Software auf dem PWM-Kontroller (Abbildung 3) startet nach einer Initialisierungsphase eine 200 us Schleife, was einer PWM-Granularität von 1 % bei 50 Hz entspricht. Innerhalb eines Schleifendurchlaufes hat jeder PWM-Pin eine feste Position, unabhängig davon ob er aktiv (5 V) oder inaktiv (0 V) ist. Wenn er aktiv ist, wird in einem Zähler festgehalten, wie viele 200 us Perioden noch verbleiben. Ist dieser Zähler gleich Null, wird der Ausgang ohne weiteres Zutun des Steuerkontrollers deaktiviert. Der Steuerkontroller kann dabei einen Pin nur für die Dauer von maximal 50 ms aktivieren. Der Empfang von neuen UART Paketen führt dabei nicht zu einer Verschiebung der gesamten Schleife, sondern maximal um eine Verschiebung eines einzelnen aktiven PIN um weniger als 10 us.

Diese Vorgehensweise garantiert, dass bei Ausfall der Kommunikation zwischen Steuer- und PWM-Kontroller sämtliche Ventile nach spätestens 50 ms geschlossen sind und so die Position gehalten wird. Bei Problemen in der Kommunikation zwischen Zentralrechner und Steuermikrokontroller wird die letzte Zielposition noch angefahren und danach passive (ohne Nachregelung) oder aktive (bei nicht terminierenden Regelprozessen) gehalten. Der Reset eines Steuerkontrollers durch Kurzschluss, Watchdog oder Spannungsabfall wird wie ein Ausfall des UART behandelt. Der Reset eines PWM-Kontrollers schaltet hingegen alle Ventile aus, verfällt aber ab der nächsten 20 ms Periode wieder in den normalen Betriebsmodus

## 4 Technische Daten

Oberkörper mit zwei Armen und zwei Fünf-Finger Händen (45kg):

- 52 Muskeln für Hände und Arme
- 104 Schaltventile
- 32 Winkelsensoren
- 20 Drucksensoren
- 8 Mikrokontroller (je 10 MIPS)
- 32 Freiheitsgrade

Rollbarer Standfuß mit Display (40kg):

- Barebone PC (Windows XP)
- Stromversorgung für Elektronik

Abmessungen:

- gleiche Proportionen und Bewegungsradius wie ein Mensch von 190 cm
- Arbeitslast: 2 kg

Ansteuerung:

- festes Programm, Teach-In oder Datenanzug mit Handschuhen

## 5 Ergebnisse

Es konnte gezeigt werden, dass ein dezentraler Aufbau mit Verteilung der Aufgaben auf jeweils verschieden hierarchischen Ebenen bei komplex anzusteuernenden Aktoren nicht nur möglich ist, sondern sich auch als vorteilhaft erweist. Durch die Verlagerung der Regelung auf die dezentralen Einheiten, verringern sich die Echtzeitansprüche an den Zentralrechner extrem. Damit kann sich der



Zentralrechner um zeitlich schwer einzuschätzende Aufgaben wie, Kinematikmodell oder Benutzerinteraktion kümmern.

Die dezentralen Controllerbaugruppen übernehmen alle niederen Aufgaben. Sie führen die Ist-Positionserkennung, Regelung zur Soll-Position, und Sensorbearbeitung getrennt, also „unterbewusst“ ohne Zentralrechner aus. Sollten Fehler oder äußere Störungen einzelne Segmente oder die Kommunikationsverbindung unterbrechen, sind die Controller immer noch in der Lage, letzte Befehle oder einen Notfallplan durchzuführen. Selbst ein Neustart einzelner Segmente führt wieder direkt zu einem normalen Betrieb, da der Ausfall zu einem Verharren in der letzten Position führt, ausschließlich absolut messende Sensoren verwendet werden und eine Rückmeldung über empfangene und verarbeitete Befehle erfolgen kann.

Eine schnelle Aktivierung der Muskeln durch naheliegende Ventile konnte gewährleistet werden, die Verdrahtung und Verschlauchung wurde stark reduziert und nicht zuletzt die Fehlersuche deutlich vereinfacht. Letzteres ist speziell auf die streng räumlich als auch dantenflusstechnisch getrennten Schnittstellen zurückzuführen. Als Nachteil sollte man den erhöhten Verwaltungsaufwand der verschiedenen Programme der Mikrocontroller berücksichtigen, der bei fehlenden Synergien entsteht.

## 6 Literatur

- [1] Bar-Cohen Y: „Electroactive Polymer (EAP) Actuators as Artificial Muscles – Reality, Potential and Challenges“, 2nd Edition, SPIE Press , ISBN 0-8194-5297-1 March 2004, pp. 1-765
  
- [2] Kazuhiro Otsuka, Xiaobing Ren: „Recent developments in the research of shape memory alloys“, Intermetallics, Volume 7 Issue 5, , May 1999, Pages 511-528
  
- [3] Boblan I., Bannasch R., Schulz A. and Schwenk H., „A human-like robot torso ZAR5 with fluidic muscles: Toward to a common platform for embodied AI“, 50th Years of Artificial Intelligence, Essays Dedicated to the 50th Anniversary of Artificial Intelligence, Lungarella M., Iida F., Bongard J., Pfeifer R. (Eds.), in Lecture Notes in Artificial Intelligence 4850, Springer, 2007, p. 347-357, ISSN 0302-9743, ISBN-10 3-540-77295-2
  
- [4] Boblan I., Bannasch R., Maschuw J., Schulz A. and Schwenk H., „A Human-like Robot Hand and Arm with Fluidic Muscles: Modelling of a Muscle Driven Joint with an Antagonistic Setup“, 3rd Int. Symposium on Adaptive Motion in Animals and Machines, Technische Universität Ilmenau, Germany, September 25th-30th, 2005, Pages 7, ISBN 3-938843-03-9
  
- [5] Boblan I., Bannasch R., Schwenk H., Miersch L., Schulz A., „A Human like Robot Hand and Arm with Fluidic Muscles: Biologically Inspired Construction and Functionality“, Embodied Artificial Intelligence, Dagstuhl Event 03281, Springer, 2003, p.160-179
  
- [6] FESTO AG & Co. KG, Projekte 2006, [http://www.festo.com/cms/de\\_de/5001.htm](http://www.festo.com/cms/de_de/5001.htm)
  
- [7] FESTO AG & Co. KG, Fluidic Muscle, [www.festo.com/net/de\\_de/downloads/Download.ashx?lnk=25738/info\\_501\\_de.pdf](http://www.festo.com/net/de_de/downloads/Download.ashx?lnk=25738/info_501_de.pdf)