

Christoph Veigl:

Entwicklung eines Geräuscheortungs- und -klassifikationssystems für mobile autonome Roboter (Sem. körperbasierte AI von Erich Prem) behandelt: Motor, Mikrofon, Schnittstelle, Fourier

I. Motivation und Zielsetzung

Das Soundortungssystem soll einem mobilen Roboter die Möglichkeit geben, Geräuschquellen im Raum zu lokalisieren. Das heißt, ausgehend von der Fahrtrichtung des Roboters ermittelt ein Softwaremodul die Richtung einer Geräuschquelle, die bestimmte Charakteristika erfüllt.

Portabilität bzw. leichte Modifizier- und Erweiterbarkeit werden angestrebt. Hauptsächlich wird aber das Augenmerk auf die konkrete Implementierung des Systems auf dem mobilen Robotersystem NUCK2000 gelegt. NUCK2000 ist ein raupenartiger Roboter, der sich auf der Stelle drehen kann. Die Soundortung soll so funktionieren, daß ein mechanischer Aufbau vergleichbar mit einem Radar ein Richtmikrofon dreht, die Daten über die Lautstärke durch eine Soundkarte digitalisiert und auf einem Mainboard, das sich auf dem Roboter befindet, weiterverarbeitet werden.

Der Roboter erhält so die Möglichkeit, die Informationen über Geräuschlokalitäten in sein Verhalten einzubauen, Soundquellen gezielt anzusteuern, die Geräusche zu analysieren und aufzunehmen.

2. Hardware und Mechanik

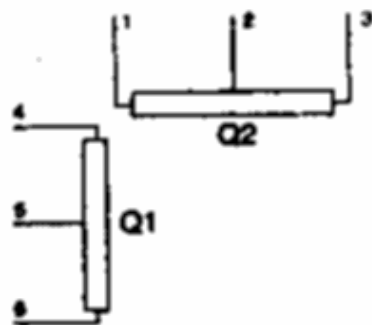
2.1 Motoren

	Gleichstrommotor	Schrittmotor
Vorteile	einfach anzusteuern billig	präzise keine Positionsrückgabe erforderlich
Nachteile	Positionsrückgabe über Potentiometer erforderlich	

Verwendung eines Schrittmotors

Voraussetzung für eine gewissenhafte Ortung ist, daß die Steuerungslogik Informationen über die aktuelle Position der Drehvorrichtung für das Mikrofon besitzt. Um dies zu gewährleisten, braucht man bei der Verwendung eines Gleichstrommotors eine Potentiometer-Positionserfassung. Dies hat den Nachteil, daß ständig die Potentiometerposition digitalisiert werden muß. Besser geeignet ist daher die Verwendung eines Schrittmotors.

Ein Schrittmotor ist impulsgesteuert, d.h. der Motor bewegt sich pro Spannungsimpuls um eine bekannte Winkelgröße. Der Motor besitzt nicht wie ein Gleichstrommotor nur 2 sondern 6 Anschlüsse. In eine kontinuierliche Drehbewegung versetzt man den Motor dadurch, daß man ein Potential von 12V von Pin 1 nach Pin 4 wandern läßt, und dann wieder bei Pin 1 beginnt. (siehe Graphik). Wenn man rückläufig vorgeht (von Pin 4 nach Pin 1 und wieder bei Pin 4 beginnend) dreht sich der Motor in die entgegengesetzte Richtung. (Pin 5 und 6 sind mit GND belegt) Man kann den Schrittmotor also mit verschiedenen Geschwindigkeiten und in beide Richtungen drehen, wobei man immer rechnerischen Aufschluß über die aktuelle Position des Motors erhält, ohne ein externes Feedback zu benötigen.



Schaltreihenfolge der Spulen:

- 2-1
- 2-3
- 5-6
- 5-4

Abb 1: Ansteuerung des Schrittmotors

2.2 Mikrofone

	Funkmikrofon	dyn. Richtmikrofon	Parablmikrofon
Vorteile	kabellose Übertragung mehrere Einheiten können senden und empfangen	gute Richtqualität	gute Richtqualität preisgünstig vorverstärkt
Nachteile	kaum richtungsspezifische Geräuschinformation	teuer groß	

Verwendung eines vorverstärkten Kondensatormikrofons mit Parabolspiegel

Die ersten Experimente bezüglich des Mikrofons wurden mit einer Funkwanze durchgeführt. Das Funkmikro ist ca. 2 cm lang und überträgt die Lautstärkeschwankungen in den UKW-Bereich, sodaß mit einem Radioempfänger im Umkreis von ca. 100 m das Geräusch abgehört werden kann. Durch die kabellose Übertragung besteht 360 Grad Drehfreiheit für das Sound-Radar, mehrere autonome Einheiten können gleichzeitig senden und empfangen.

Die besten Ergebnisse wurden mit einem Bausatz für ein verstärktes Parabolmikrofon erreicht. Ein Brennspiegel mit 28 cm Durchmesser konzentriert die gerichteten Schallwellen auf einen Punkt, wo ein Miniatur-Knopfmikrofon (Kondensator-Elektret) befestigt ist. Das Mikrofonsignal wird mit einer Transistorschaltung vorverstärkt und dient als Line-In-Signal für die Soundkarte.

2.2.1 Konstruktion eines Mikrofonverstärkers

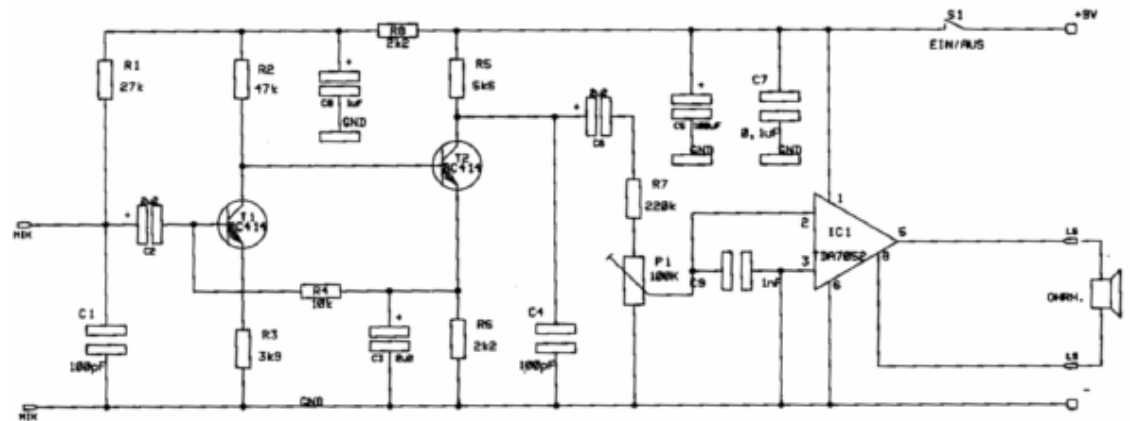


Abb2: Schaltplan des Mikrofonverstärkers

3. Hardware-Interface

3.1 Klangdigitalisierung

	Prozessorboard	Soundkarte auf Mainboard
Vorteile	klein preisgünstig	Speicher und Rechenzeit ausreichend
Nachteile	zu geringer Speicherplatz	groß und schwer Spannungsversorgung über Akkus

Verwendung eines Mainboards mit Soundkarte

Bei diesem Punkt bestehen leider nicht viele Wahlmöglichkeiten, da das Soundsampling, die Fourier-Analyse und die Speicherung der Klangergebnisse nur von einem Mainboard erledigt werden können (80486, 8MB RAM, 500 MB Harddisk)

3.2 Motorsteuerung

	Prozessorboard	I/OCard	Parallellport
Vorteile	klein	analoge Spannungsausgänge, sicher	preisgünstig einfach zu programmieren
Nachteile	teuer (Parallport vorhanden)	teuer	

Verwendung des Parallellports

Es wäre möglich, den Motorsteuerung über ein Prozessorboard zu erledigen. Da die Entscheidung für die Verwendung eines Mainboards aber bereits beim Sound-Sampling notwendig wurde, ist es einfacher, die Schrittmotoren über den Parallellport anzusteuern.

Die Schrittmotoren benötigen nur einen Spannungsimpuls zum Drehen. Diese Möglichkeit ist einfacher und kostengünstiger als die Verwendung einer D/A-Wandlerkarte.

Der Parallellport kann zur Ausgabe von 8-Bit Information auf 5V verwendet werden. Durch den Aufbau einer kleinen Relais- oder Transistorschaltung können die Schrittmotoren mit den nötigen 12V Spannung versorgt werden.

3.2.1 Konstruktion einer Schaltung für den Parallellport

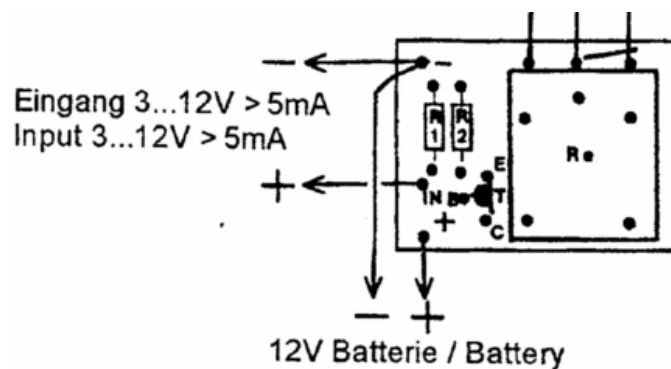


Abb 3: Schaltplan für die Relais-Steuerung

4. Software

4.1 Fourier-Analyse der digitalisierten Daten

Um Aufschluß über die Eigenschaften von Geräuschen zu erhalten, eignet sich eine Konvertierung vom Amplitudenspektrum ins Frequenzspektrum, das durch eine Fourier.-Analyse erreicht wird. Dadurch wird es möglich, spezielle Charakteristika der Känge wie Frequenzanteile, Hüllkurve über das Frequenzspektrum, etc. zu berechnen. Man kann somit zB. mit einer großen Wahrscheinlichkeit Stimmen von anderen Geräuschen unterscheiden und gezielt Geräuschquellen mit bestimmten Eigenschaften herausfiltern. Im konkreten Fall wurde ein FFT-Algorithmus verwendet, der in C++ erstellt und in eine DLL exportiert wurde.

SPECTR-DLL:

```
function SpecInit(dev:longint;smpfreq:longint;
    chns:longint;n:longint;ov:longint;cbtp:longint;
    cb:longint): longInt;
    stdcall; external 'SPECTR';
    ---> Frequenz, Kanäle, Bänder
function SpecExit: longInt;
    stdcall; external 'SPECTR';
    ---> Speicher freigeben
function SpecGet(dest:intptr): longInt;
    stdcall; external 'SPECTR';
    ---> Poll Frequenzdaten
function SpecWindow(wnd:longint): longInt;
    stdcall; external 'SPECTR';
    ---> Hanning, Gauss etc.
function SpecDamping(dmpfct:double): longInt;
    stdcall; external 'SPECTR';
    ---> Dämpfungsfaktor
```

4.1.1 Initialisierung

Die Initialisierung der Fourier-Transformation erfolgt mit 22050 Hz Sampling-Frequenz, Mono (1-Kanal) auf 512 Frequenzbandern:

```
i:=specinit(-1,22050,1,512,1,0,0);
i:=specwindow(0);
```

4.1.2 Aufruf

Bei jedem Aufruf (polling) wird ein 512 longints großer Puffer mit den Daten über die Frequenzbandaktivität gefüllt:

```
i:=specget(@buf);
```

4.2 Klassifikation der Fourier Daten.

In der jetzigen Ausbaustufe werden die Frequenzbänder nach dem größten Pegel durchsucht. D.h., das Modul sucht den lautesten Ton aus der Fourier-

Analyse. Diese Tonhöhe und der Winkel der Soundquelle zur aktuellen Fahrtrichtung des Roboters können vom Modul zurückgeliefert werden. (Weiterentwicklung der Klassifikation siehe 6.1)

4.3 Motorsteuerung

Die Motoransteuerung funktioniert denkbar einfach: 4 Bits der parallelen Schnittstelle werden vor- bzw. rückläufig auf 1 gesetzt (je nach Drehrichtung). Das Relais, das die 5V Eingangsspannung vom Paralellport erhält, schaltet durch, die zugehörige Motorspule wird unter 12 V Spannung gesetzt, der Motormagnet richtet sich an die entsprechende Stelle aus. Die Parallele Schnittstelle liegt auf Port 378h:

```
asm
    mov al,actpos
    xor al,0ffh
    mov dx,378h
    out dx,al
end;
```

Die Prozedur motorstep dreht den Motor um 1 Schrittweite (1,6 Grad) nach rechts, wenn direction 1 ist, im anderen Fall um 1 Schrittweite nach links:

```
procedure motorstep(direction:boolean);
begin
    if direction=1 then
    begin
        if (actpos>128) or (actpos<16) then
            actpos:=128
        else actpos:=trunc(actpos/2);
        end
    else
    begin
        if (actpos>128) or (actpos<16) then
            actpos:=16
        else actpos:=actpos*2;
        end;
    put;
end;
```

4.4 DLL, Interface zum aufrufenden Programm:

Folgende Funktionen werden als DLL-Routinen zur Verfügung gestellt:

```
procedure initsoundsearch;
    ---> Start: Fourier-Settings, Initialisierung
procedure exitsoundsearch;
    ---> Speicherfreigabe
procedure resetsearchpos;
    ---> Dreht den Spiegel nach 0 Grad
procedure motorstep(direction:boolean);
    ---> Dreht den Spiegel 1,6 Grad
        links bzw. rechts
function getvolume : longint
    ---> liefert Frequenz des lautesten Tones
```

```
function getangle: integer  
---> liefert aktuelle Spiegelposition
```

5. Test und Laufzeitanalysen

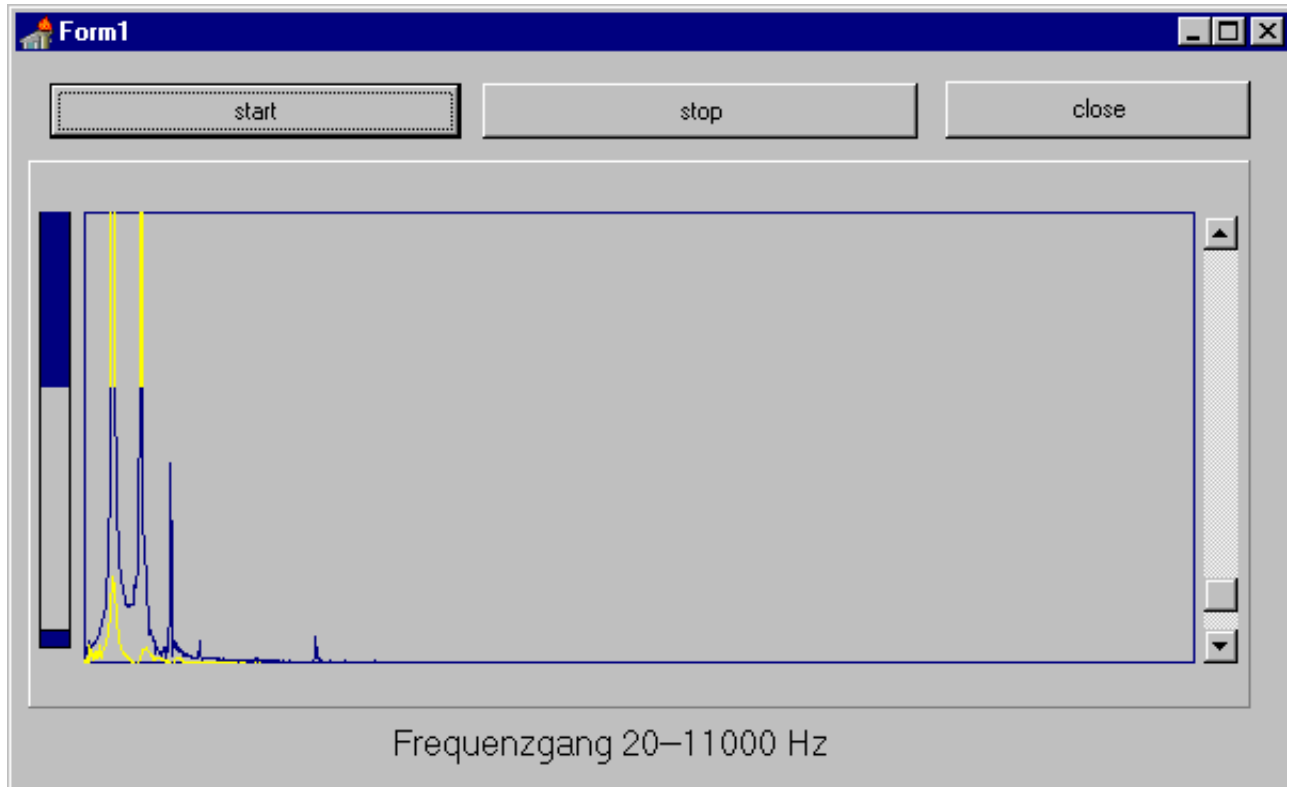


Abb 4: Screenshot Fourier-Analyse einer sprechenden Person aus ca 3 m Entfernung

Erste Tests ergaben eine gute Funktionalität des Moduls in großen Räumen. In kleinen Zimmern mit vielen Gegenständen und vielen Reflektionsflächen für den Schall war nicht immer eine eindeutige Auffindung der Schallquelle möglich. Hier könnten wahrscheinlich durch die Verwendung eines auf diese Umweltbedingungen speziell angefertigten Parabolspiegels noch Verbesserungen erzielt werden. Da NUCK2000 in großen Räumen aber sowieso besser navigieren kann, wurde von dieser Weiterentwicklung abgesehen.

6. Ausbau und geplante Weiterentwicklungen

6.1 Weiterentwicklung der Soundanalyse

In der jetzigen Ausbaustufe liefert das Modul lediglich den lautesten Ton in der Umgebung. Allein für diese Funktion wäre die Fourier-Analyse gar nicht notwendig gewesen. Diese Funktionalität stellt allerdings nur einen erweiterbaren Grundstock an Auswertungskriterien des Soundsignals dar. Ziel der Weiterentwicklungen wäre es, eine „Situating Activity im Klangraum“ zu ermöglichen: Die Klänge können nach Frequenzhöhen, -zusammensetzung, Klangbild und auch im Rhythmus analysiert werden. Diese Analysen von Geräuschen können vom Roboter im Zusammenhang mit den aktuellen Sensorwerten betrachtet werden.

6.2 Recording und Weiterverarbeitung der Samples

Der Roboter soll die Möglichkeit bekommen, Klänge aufzunehmen und zu speichern. Diese Klangdatenbank dient als Ausgangsbasis für die Berechnung von eigenen Geräuschen, mit denen der Roboter auf Umgebungsbedingungen reagieren kann.