

# Geluidsherkenning door de Aibo

Laura Helgering  
10198539

Fien Pluymaekers  
10458956

Loïs Mooiman  
10318364

28 Juni 2013

## Abstract

De Aibo is een robot hond die geprogrammeerd kan worden om verschillende opdrachten uit te voeren. Doordat in de Aibo verschillende sensoren zitten zoals een camera, een microfoon, een speaker en detectiesensors, kan de Aibo veel verschillende taken uitvoeren. Door een programma te schrijven, was het de bedoeling de Aibo onderscheid te laten maken tussen de frequenties van verschillende toonhoogtes en de Aibo bij elke toonhoogte een andere truc te laten doen. Door een aantal problemen die voorkwamen kan de Aibo alleen verschil maken tussen geluidsvolume.

## Inleiding

Een Aibo is een robot hond, die kan worden geprogrammeerd om verschillende taken te verrichten. Voor het eindproject van 'Zoeken Sturen en Bewegen' was het de bedoeling om een Aibo trucs te laten uitvoeren door middel van geluiden. Door de Aibo verschillende volumes te laten horen, reageert de Aibo anders. De Aibo kan voor verschillende doeleinden worden gebruikt doordat de Aibo zelf kan worden geprogrammeerd door de gebruiker. De Aibo kan als vervanging dienen voor huisdieren, bijvoorbeeld voor mensen die allergisch zijn. Ook kan deze worden ingezet als entertainment in bejaardenhuizen of ziekenhuizen. Doordat de Aibo uit de handel is gehaald, is er niet veel onderzoek meer naar gedaan en is er ook niet heel veel over bekend. Als er weer een herstart komt van de fabricatie van Aibo's, worden deze weer maatschappelijk relevant. Voor het onderzoek naar de Aibo is de volgende onderzoeksvraag opgesteld:

*Kan de Aibo een verschil maken tussen verschillende geluidsfrequenties en hier anders op reageren?*

De hypothese is als volgt: De Aibo kan onderscheid maken in verschillende frequenties van toonhoogte, hier anders op reageren en verschil maken in de richting van waar het geluid vandaan komt.

De verschillende frequenties kan de Aibo opnemen en onderscheiden. Bij deze onderscheiding kan aan elke frequentie een andere code gekoppeld worden en zo kan de Aibo op verschillende frequenties anders reageren.

Als proefopzet was het de bedoeling om een wifi verbinding te kunnen maken met de Aibo. Hierdoor zouden alle gegevens van de Aibo worden opgevraagd, uitgelezen en veranderd. Helaas kon er geen verbinding tot stand komen. Als oplossing moest de memorycard uit de Aibo worden gehaald en met behulp van een adapter op de computer worden aangesloten.

## Materiaal en methode

Het gebruikte materiaal om code mee te schrijven is een macbook pro en een ultrabook van Samsung en het daarbij gebruikte besturingssysteem is windows.

Het programma is gemaakt in de programmeertaal Urbi en geschreven door middel van Notepad++. Hierbij zijn sites gebruikt met algemene informatie over Urbi (Baillie, 2006). Als robot is er een Aibo gebruikt van versie ERS-MS7 Mind 2. Hierbij werden memorycards geleverd waar het programma voor de aibo op kon worden geladen en in de Aibo worden gestopt. Voor het inlezen van de memorycard in de computer was er een Sony adapter. Om de restricties en mogelijkheden die de Aibo heeft te weten, is er een online handleiding gebruikt (Gostai, 2007). Voor het maken van geluiden werd er een xylofoon gebruikt of er werd geklapt of gesprongen. De Aibo is getest in het robolab op het Science Park of de gang ernaast.

In de code kon al gebruik gemaakt worden van verschillende functies. Onder andere:

- swalk.u van Diego Pardo/gostai, (2006).
- motion.u van Laboratory of Robotics Versailles, (2006).

## Resultaten

Het primaire doel van het onderzoek, het onderscheiden van frequenties in toonhoogtes, is niet behaald. Wat wel bereikt is, is onderscheid maken tussen verschillende volumes. Als de Aibo opstart, gaat de Aibo eerst zitten. Daarnaast worden de ledlampen bij de oren blauw en krijgt de Aibo oogjes. Dan beweegt de Aibo zijn hoofd heen en weer en is er een scangeluid. Daarna zwaait de Aibo met zijn linkerpoot. Hierna kan de Aibo reageren. Bij een zacht geluid kijkt de Aibo naar de kant van waar de Aibo denkt dat het geluid vandaan komt. Hierbij blaft de Aibo en worden de ledlampen bij de oren groen. Bij een hard geluid staat de Aibo op en kleuren de lampen bij de oren rood. De Aibo loopt tien stappen waarnaar deze weer gaat zitten. Terwijl de Aibo loopt reageert het nog steeds op geluid. De Aibo kan ook over zijn hoofd geaaid worden. Dan blaft de Aibo en worden de lampen bij de oren groen. Tijdens het blaffen beweegt de mond van de Aibo.

## Discussie

Er is begonnen met het onderzoeken van de Aibo met het idee dat de Aibo geluidsopnames kan maken en dat er daardoor onderscheid kan worden gemaakt tussen frequenties door middel van geluidsfragmenten vergelijken. Dit bleek niet te werken omdat de Aibo geen goede microfoon heeft en hierdoor veel te veel ruis opving. Daarnaast kon het opgenomen geluid niet worden opgehaald omdat er geen wifi-verbinding kon worden opgesteld. Hierdoor is het doel bijgesteld. Het nieuwe doel is de Aibo verschillend te laten reageren op geluidsvolume. Het lopen van de Aibo gaat redelijk traag. Dit komt doordat de Aibo begint in een zitpositie maar vanuit een zitpositie niet direct kan gaan lopen. Uiteindelijk is dit opgelost om een andere vorm van lopen te gebruiken. Hierbij gaat de Aibo eerst liggen, waarnaar de Aibo gaat zitten en daarna staan en lopen.

Een ander probleem was het detecteren van een object voor de neus van de Aibo. De Aibo moest een poot geven als er een object dichtbij zijn neus zat. Dit heeft gewerkt. Alleen na een toevoeging in de code, werkte dit niet meer. Hierdoor is dit deel van de code eruit gehaald.

Daarnaast is er gewerkt aan detectie van object tijdens het lopen. Als er een object dichtbij kwam, moest de Aibo draaien en verder lopen. In combinatie met het probleem van het opstaan werkte dit niet. De Aibo ging willekeurig draaien en liep daarna niet door. Hierdoor is dit deel van de code eruit gehaald voor de demonstratie.

De conclusie die uit de resultaten kan worden getrokken, is dat de Aibo goed kan registreren waar het geluid vandaan komt en hierop kan reageren. Daarnaast kan de Aibo het geluidsvolume horen en hier onderscheid tussen maken.

Als vervolgonderzoek kan er worden gekeken naar het probleem met de microfoon en het ophalen van de data. Zodat de Aibo kan reageren op verschillende frequenties. Daarnaast kan er in het vervolgonderzoek de bovenstaande problemen worden opgelost.

## Literatuur

Baillie, J. (2006). URBI Tutorial. retrieved from <http://www.gostai.com/doc/en/urbi-tutorial/index.html>  
Gostai (2007). URBI DOG for Aibo ERS2xx ERS7 and URBI 1.0. retrieved from <http://gostai.com/doc/en/aibo/index.html>

## Appendix

Onze code:

```
load("std.u"); //StandardURBI file for Aibo
load("motion.u"); //Walk functions
load("swalk.u");
speaker -> blend = mix;
motorson;
robot.initial();
robot.sit();
#Eyes
ledF3.val = 1;
ledF4.val = 1;
ledF6.val = 1;
ledF7.val = 1;
ledF8.val = 1;

#Start up
headPan.val = -90 time:2s & modeB.val = 1 & speaker.play("client.wav");
headPan.val = 90 time:2s & modeB.val = 1 & speaker.play("client.wav");
```

```
headPan.val = 0 time:2s & modeB.val =1 & speaker.play("client.wav");
modeB.val = 0;
```

```
#wave
legLF1.val = 70 time:1000;
legLF3.val = -26 time:1000;
legLF2.val = 45 time:1000;
legLF2.val = -9 time:1000;
legLF1.val = -45 time:1000;
```

```
# Soft Sound Recognition + bark
at(micro.volume 6000){
  if(micro.volume10000){
    if(micro.volumeR > micro.volumeL){
      for(i = 0; i < 2; i ++){
        headPan.val = -90;
        earR.val = 1;
        modeG.val = 1;
        bark();
        resetBark();
      };
      headPan.val = 0time : 3s;
      earR.val = 0;
    };
    if(micro.volumeR < micro.volumeL){
      for(i = 0; i < 2; i ++){
        headPan.val = 90;
        earL.val = 1;
        modeG.val = 1;
        bark();
        resetBark();
      };
      headPan.val = 0time : 3s;
      earR.val = 0;
    };
  };
};
```

```
#HardSoundRecognition + runaway
at(micro.volume > 10000)modeR.val = 1; robot.ObstacleDetection(TRUE); robot.swalk(10); bark(); mode
```

```
functionbark()
for(i = 0; i < 2; i ++ )headTilt.val = 20; mouth.val = -58time : 1s; speaker.play("bark.wav"); ; mouth.v
```

```
functionresetBark()
headTilt.val = 0; modeG.val = 0;;
```

```
#Touchinputs  
#Barking  
at(headSensor.val > 0)bark();resetBark();;
```