An Interactive Autonomous Musical Robot

Creating Music with the Ugobe Pleo

Andria Poiarkoff
Sonic Arts Research Centre
Queen's University Belfast
Belfast, BT7 1NN
apoiarkoff01@qub.ac.uk

Abstract

As an exploration of autonomous musical robotics, the Ugobe Pleo has been used as a platform for experimentation. The robotic dinosaur has been researched, studied, dissected, augmented, rebuilt, and connected via serial interface to a digital signal processing application. The goal of the project is to utilize the Pleo's own actions to allow the autonomous creation of a soundscape, while allowing interactive participation and manipulation through the robot's external sensory system.

Keywords: autonomous robots, robot music, musical interfaces, interactive systems, human-robot interaction

1. Introduction

The field of musical robotics has existed for decades, from devices such as the player piano and metronome-based percussionists [5] all the way to the robotic instrumentalists of LEMUR. [8] While musical robots exist [1][2][9], even autonomous musical robots such as those of LEMUR, very few systems combine together autonomy and human interaction[12].

The use of robotic applications as tools for investigation of interaction or relevant behavior in general is not established within music research. There are, however, applications, which can be considered relevant for topics currently discussed with respect to music-related behavior, such as the use of gestures and more general bodily movements…[1]. However, when it comes to gesture and movement, most musical robots rely on standard mathematical approaches to music creation, and while they function autonomously, the sound is based entirely on strict algorithms. [4][7]

The Sony Aibo has been used for musical applications, such as a project creating sound in response to emotion [6], and another in which the Aibo recognizes tempo and percussive sounds [10]. The Aibo was full of technology that could be put to use in interactive applications, however its cost and later discontinuation left it an inefficient platform for development.

The goal of this project is to utilize the Ugobe PLEO as an accessible platform, developed to function as an interactive autonomous musical robot.

2. Default Configuration

All PLEO robots contain the same hardware and software configuration, though as the standard programming allows PLEO to develop an individual personality, after a period of time different PLEOs will show different actions and abilities.

2.1 Sensory System

Interaction with PLEO is based on a system of sensors:

- 8 touch sensors: head, chin, back, rear, and one on each leg
- Tilt/motion sensors which detect change in position
- IR proximity sensor located in the nose
- Color camera located in the nose
- IR interrupter located in the mouth
- Binaural microphones
- 4 force sensors, one on each foot

The sensors in the PLEO robot are programmed for specific interactions. Each touch sensor triggers a behavior or behavior set. The tilt and motion sensors allow PLEO to respond to being held in unnatural positions, such as upside-down, and to call for help when it has fallen over. The IR sensor in the nose is only functional within a distance of approximately 3 to 4 inches and is used to allow the robot to navigate around objects as well as to react to objects placed in front of it. The color camera, while holding great potential, is only programmed to react to particular shades of green and yellow. These colors are used in PLEO's leaf toy, and it will respond to other objects of the same color. The IR interrupter located in the mouth is used to detect objects in the mouth, which then triggers behaviors such as eating or tugging. PLEO's binaural microphones serve only the purpose of allowing PLEO to respond in the direction of a sound. For instance, if one were to talk to the left of PLEO's head, it may turn its head to the left. If a loud sound is made, such as a clap, it will turn its head away. PLEO also has force sensors in its feet, partly for the purpose of keeping it from walking off of surfaces, but also to allow for interactions such as "high five".

2.2 Mechanics

PLEO's movement relies on 14 servos and force feedback sensors. The motorized joints include the neck, tail, waist, mouth, eyes, and two in each leg. PLEO has three default poses: standing, sitting, and sleeping. From a physical perspective PLEO is able to walk, dance, talk, wag its tail, and respond in a manner not unlike a small dog.

Because of its internal motion sensors, PLEO is able to respond physically to force and acceleration. For instance, if one were to place PLEO on a skateboard and push it forward, PLEO would compensate for the movement and balance itself. In the same manner, if one places a finger in PLEO's mouth and tries to pull away, it will play 'tug of war'.

PLEO's mechanics do have some limitations. The servos controlling the legs and mouth are quite slow and can easily be overworked when programming new motions. In addition, some motions cannot be done at the same time because of shared servos. For instance, PLEO cannot open its mouth while its eyes are closed.

2.3 Controls

PLEO contains 6 separate processors. The main processor is an ARM7 32-bit processor, which handles most of PLEO's input as well as SD, USB, and Flash data. This can be considered the 'brain' of the PLEO robot. There is a separate ARM7 32-bit microprocessor in PLEO's head that handles input from the microphones, camera, and IR sensors. There are four additional 8-bit processors and two Dual H-Bridge chips that control the motors.

2.4 Operating System

PLEO runs an operating system called Life OS. Life OS is written in Pawn, a simplified version of the C programming language. The operating system is based on pseudo-AI, progressing through three stages to develop a personality based on experience. When a PLEO is first turned on it has the behaviors of a baby, sleeping regularly, cooing, and desiring of affection. The next stage involves PLEO beginning to clumsily explore its environment. Interaction during this stage will serve to impact the eventual personality of the PLEO. Once its personality has been developed, PLEO will explore on its own as well as learning new actions.

Currently Life OS is closed source. A PLEO Developer Kit was due to be released in early 2009, however with the collapse and following liquidation of Ugobe, it is unlikely the kit will ever be made public.

3. Tools and Resources

3.1 Third Party Software

Numerous programs have been developed to customize the PLEO robot. MySkit [20] is a program developed by DogsBody that allows one to build behaviors using a motion sequencer, programming each servo over a period of time, a concept similar to computer animation. Audio files can be used alongside the motions, and a 3D real-time representation of the movements allows one to preview the behavior without having to write to an SD card. YAPT (Yet Another Pleo Tool) [21] was released by the AiboHack group and is used to create personalities for PLEO. It is not a powerful tool, but can be combined with custom behaviors to develop different personalities. There is also a program called Dino-Mite (Dinosaur Monitor w/ Integrated Executables) [19] released by BAUER Independents Ltd. that allows one to receive and monitor data from PLEO's various inputs.

3.2 Resources

As the PLEO is relatively new, and now discontinued, resources for the robot are predominately composed by amateur robot enthusiasts and only found online. Most information is found on forums and hobby sites. *AiboHack* [17] contains a large amount of detailed software and hardware information. Sites such as *howstuffworks* [18], *iFixit* [16], and *Robostuff* [13] contain articles relating to the release and specifications of the PLEO as well as user-submitted projects and hacking tips. A forum from *Bob the Pleo* [15] is an excellent community containing numerous threads on technical problems, hardware and software, and projects in development.

GRIP, the Group for Interdisciplinary Psychology in Germany, is an academic group attempting facial recognition and emotion sensing with the PLEO. Their resources provide a step-by-step manual to upgrading the robot's camera and give insight into PLEO's visual system and what potential it holds.

4. Exploration and Development

4.1 Hardware Exploration

Exploring PLEO's hardware system involved careful dissection. The robot's 'skin' was removed by separating the glued seams, then cutting around areas that were attached to the plastic skeleton. Once this was accomplished, PLEO's outer shell (two large pieces creating the robot's back) was removed. Underneath the shell are additional shields preventing one from damaging the complex servo system. The next item removed was the skull. Underneath PLEO's skull lies a large amount of vital circuitry, including that controlling the eyes, mouth, camera, IR sensor, front speaker, and binaural microphone 'ears'. The next step involved carefully separating PLEO's torso joint. Once this was accomplished one could observe the main processors.

PLEO's speakers were replaced with larger, higher quality speakers to alleviate the roughness of the sounds produced. The robot's microphones were replaced with slightly larger, more sensitive microphones. While the speaker replacement did not result in much improvement, the effects of the new microphones wear instantly visible.

While PLEO previously only responded to loud sounds such as a clap or sharp command, it now focused curiously in the direction of my lab mates as they spoke at a normal conversational level.

4.2 Software Exploration

After experimenting with the three discovered third-party applications, software exploration with the PLEO robot was initially done through the use of MySkit via SD card. Movements and sound were combined and sequenced into complex actions. These actions were then programmed as responses to particular sensor stimuli. While vital in gaining an understanding of how PLEO's mechanical, audio, and personality systems work together, the programs created in MySkit were effectively replacing the original embedded program and as such removed PLEO's overall autonomy.

It was decided that access to PLEO's system would need to be through a means other than the autonomy-destroying SD card. PLEO has a built-in USB port, which is utilized by the Dino-MITE program, however the PC to which the robot was connected had difficulty recognizing the port. Accessing PLEO's additional serial port was attempted as detailed in in the article *How to Control Pleo Wirelessly with Wii Nunchuk.* [13] A cable was created as per the instructions, however while the PC recognized the serial port, there was no data stream. After numerous attempts at accessing both ports, it was realize that the issue was not with the robot itself, but with the computers' own serial drivers. PLEO was eventually successfully connected via USB port to a MacBook Pro using CoolTerm [22], a freeware serial port terminal application.

4.3 Development

Once an understanding of the hardware and software capabilities had been gained, it was time to move on to finding ways of utilizing the PLEO as a platform for development. PLEO's system is designed to accept text commands. Connecting the robot's serial port through CoolTerm allowed access to the internal system. Using a simple 'help' command generates a screen of the comprehensive workings of PLEO. Through this interface one is able to initiate behaviors, actions, and sounds, print details of the robot's current status, load programs, and most importantly access the data streams for every one of PLEO's sensory inputs. These data streams could observed for each sensor independently, or for all sensors, in real-time. At this point PLEO had developed from a partially hacked robot into a fully accessible hardware interface.

An unexpected problem occurred at this point. PLEO is made to accept software updates via USB, and because of this and the fact that removing the robot mid-update could destroy its operating system, it has a safety mode that causes it to cease physical activity while the USB is connected. In order to keep PLEO responding as its normal self while connected to the computer, the entire operating system was copied to an SD card. The SD card then serves as the primary boot drive, and PLEO is back to its active self.

5. Computer Interfacing

5.1 Interfacing with PLEO

The musical portion of this project utilizes Max/MSP (detailed in Appendix IV. Max/MSP Development). The first challenge in creating an interface to work with was creating a process by which Max/MSP would recognize the PLEO as a serial interface. The port itself was identified immediately by the program, as is expected for any standard USB device. However, the complexities emerged when communication with PLEO was attempted. PLEO responds only to its own set of text commands, and no data could be streamed from the robot until the robot was given a command to output the data. While this task was originally regarded as being very complex, after large amounts of unsuccessful work it was realized that the elusive solution was actually quite simple. One had only to create a message in Max/MSP, send that message character by character, and send a return command. Signal Routing

The PLEO robot has 40 sensor outputs plus a further 16 for joint feedback. As such, sending a single 'monitor enable all' command results in a real-time data stream of the status of 56 separate sensors. These streams needed to be converted into usable data. In addition, while the data from all 56 sensors came in a single block, the joint and sensor data were broken into two lines preceded by text.

The first step was convert the data from integers to ASCII. From there the ASCII symbols were separated into individual messages. At this point the data could be routed based on their text prefixes (joint and sensor). The joint and sensor data was then handled and unpacked separately, so that the data was now arriving in two groups, one of 16 and one of 40.

5.2 Creating the PleoPatch

The PleoPatch is a Max/MSP patch in the form of a simple interface that allows monitoring of the real-time status of all 56 sensors. The entire process detailed previously requires one to do nothing more than choose a port and click 'start'. This patch can be easily turned into an object (or two, to keep the sensor categories separate) to give instant access to PLEO's data streams in further patches.

6. Implementation

6.1 Utilizing Data Streams

PLEO's sensors output data in different manners based on what they are being used for. The data from the joints is sent in degrees, positive or negative, based on the possible range of motion of that joint. The neck can horizontally move from -90 to 90, the shoulders from -30 to 30. The external sensors only read charge, and therefore only output a 0 or 1.

6.2 Mapping

Due to the constant data stream being received (for example, a constant string of 1 when touching PLEO's head), it was essential to gate all of the data to prevent incessant multi-triggering. The mapping for this project was created in such a way that a signal meeting a positive condition results in the closure of that signal's gate, cutting off the data though-put until the signal itself changed in value. The joint signals are used to trigger a sound or effect when the output changes to a positive value, and the sensor signals trigger based on positive activity.

6.3 Creating the Soundscape

The patch can be used to map any sound to any data stream, but for the purposes of this project the sensors were used to create atmospheric sound, and the joints were programmed with either samples of a single note or a sample of a series of on a single instrument. The main sensors on the back and head were used for special effects, such as reverb and ptichshifting. The result was a complex, and at times surreal, 'prehistoric' stormy soundscape overlaid with single bells and wind chimes.

6.4 Interaction and Experience

Interacting with the musical PLEO is a unique experience. The PLEO itself, in its autonomous behavior, will create a soundscape whether it is interacting with itself, its environment, or external stimuli. The installation could run without the aid of human participants. However, actively interacting with PLEO adds another dimension to the piece. It can trigger sounds that PLEO normally wouldn't create on its own, and alter the sound through effects programmed into particular sensors. These interactions allow a participant to play with the robot while actively sculpting the soundscape.

7. Discussion

The developmental process of PLEO as a musical robot, while faced with numerous complexities, furthered the knowledge and possibilities of this particular dinosaur. The partial dissection of the robot allowed insight into how it physically functions, but also exposed the fragility of the mechanics inside PLEO. Even working with the few parts that were replaced in the experimental stages led to an incessant worry that something would be accidentally broken.

The only major setback was in gaining access to PLEO's serial port, and it would be advised that one allows time and patience for this aspect of PLEO hacking. If it simply will not work, it is probably an issue with the serial drivers within the computer.

Handling the data streams could be made more convenient by breaking the groups into further subsections, such as a group for the foot sensors, a group for the leg sensors, etc. If one were to record the possible degree movements of the joints, they could be mapped accordingly and used to manipulate data rather than trigger it.

The overall presentation of the project was very unique and the interaction extremely intuitive. It only takes a few seconds to realize how movement and touch within the robot are used to create the soundscape.

8. Conclusion

As an example of an interactive autonomous musical robot, PLEO combined with Max/MSP created a musical experience like none other. Robotic instruments, musicans, and composers are becoming commonplace, but most often function either entirely autonomously or entirely based on human input. The musical PLEO combines these two aspects to create an experience and sound that is both fun and sonically appealing. The PLEO is an excellent platform for autonomous, interactive, and in this case sonic experimentation.

Bibliography

[1] Burger, B. & Schmidt, L., 2009, Considerations Concerning a Methodology for Musical Robotics and Human-Robot Interaction, *7th Triennial Conference of European Society for the Cognitive Sciences of Music*.

[2] Jee, E.S., Kim, C.H., Park, S.Y. & Lee, K.W., 2007, Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on, *Composition of Musical Sound Expressing an Emotion of Robot Based on Musical Factors*. pp. 637-41.

[3] Jordà, S., 2001, Proceedings of the 2001 conference on New interfaces for musical expression, *New musical interfaces and new music-making paradigms*. pp. 1-5.

[4] Kapur, A. & Singer, E., 2006, Proceedings of the International Conference on Music Information Retrieval, *A retrieval approach for human/robotic musical performance*. pp. 363-4.

[5] Kapur, A., 2005, Proceedings of the International Computer Music Conference (ICMC), *A history of robotic musical instruments*..

[6] Miranda, E.R. & Tikhanoff, V., Musical Composition by Autonomous Robots: A Case Study with AIBO, *Proceedings of TAROS 2005 (Towards Autonomous Robotic Systems)*.

[7] Schmidhuber, J., 2006, Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts, *Connection Science*, 18(2), pp. 173-87.

[8] Singer, E., Feddersen, J., Redmon, C. & Bowen, B., 2004, Proceedings of the 2004 conference on New interfaces for musical expression, *LEMUR's musical robots*. pp. 181-4.

[9] Suzuki, K., Ohashi, T. & Hashimoto, S., 1999, Proc. of International Computer Music Conference, *Interactive Multimodal Mobile Robot for Musical Performance*..

[10] Tejada, S., AI, AIBO and ART: Inspiring Interaction with Computer Science, .

[11] Vallis, O., Hochenbaum, J. & Kapur, A., 2008, Tenth IEEE International Symposium on Multimedia, 2008. ISM 2008, *Extended Interface Solutions for Musical Robotics*. pp. 495-6.

[12] Weinberg, G., Raman, A. & Mallikarjuna, T., 2009, Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, *Interactive jamming with Shimon: a social robotic musician*. pp. 233-4.

Web Resources

[13] Andrey 2009, How to Control Pleo Wirelessly Using Wii Nunchuck, *Robostuff.* from http://robostuff.com/diy-projects/pleo-hacking/how-to-control-pleo-wirelessly-using-wii-nunchuck

[14] Group for Interdisciplinary Psychology, Hacking Pleo: Working with the Pleo Platform, from http://www.grip-online.com/en/pleo_hack

[15] Pleo Hardware and Software Forum, from http://bobthepleo.com/forums/index.php

[16] Pleo Teardown, *iFixit.* from http://www.ifixit.com/Guide/First-Look/Pleo/597/1

[17] Pleo Technical Info, *AiboHack.* from http://www.aibohack.com/pleo/index.html

[18] Wilson, T., How Pleo Works, *howstuffworks.* from http://science.howstuffworks.com/pleo.htm

Software

[19] BAUER Independents Ltd, Dino-MITE [computer file]. http://www.bauerindependents.com/SUBMAIN/dinomite.htm [distributor].

[20] DogsBody, MySkit Performance Editor for PLEO [computer file]. http://www.dogsbodynet.com/myskit/index.html [distributor].

[21] YAPT: Yet Another Pleo Tool [computer file]. http://www.aibohack.com/pleo/yapt.htm [distributor].

[22] CoolTerm [computer file]. http://freeware.the-meiers.org/ [distributor].