Improving AIBO with Artificial Intelligence Technique

Gun A. Lee
endovert@postech.ac.kr

Virtual Reality and Interactive Media Laboratory,
Department of Computer Science & Engineering,
POSTECH, Pohang, 790-784, Republic of Korea.
+82-54-279-5661

Abstract

The robot industry is rapidly growing recently, and now the application fields of robots are not restricted to industrial purpose, but are expanded to entertainments, since the advent of 'AIBO,' the first robot pet of Sony corporation. In this paper, we describe the current state-of-the-art of the robot pet AIBO. In addition, we suggest some new functions for AIBO and show those can be achieved by using modern artificial intelligence techniques.

1. Introduction

A robot with an artificial intelligence is one of the famous subject matter of scientific fictions, and films. To make a robot that can talk with humans, think like humans and behave like humans is a dream of all from long time ago. Nowadays, the rapid growth of science and engineering field, especially artificial intelligence in computer science and robotics in mechanical engineering, is realizing this dream, and the entertainment market is the one that adopted these technologies rapidly.

Sony corporation[1] has announced their first entertainment robot AIBO[2] in 1999, and it was the first step of robot industry entering the public entertainment market. The appearance of AIBO was really sensational among the people. Although there are several other robot pets, such as Furby[3], Teckno[4], i-Cybie[5] and NeCoRo[6], AIBO has much far different and advanced features than the others.

In this paper, some newly suggested functions for AIBO using artificial intelligence techniques are illustrated. For this, we first review about robot pets and current state-of-the-art of AIBO. And then suggest some new expected functions and illustrate how to achieve this functions using artificial intelligence techniques.

2. Robot Pets

To say about the history of robot pets, we may trace back to when the first doll was made. A doll is very popular toy in the human history, and it is one of the natural desires of human, willing to have something resembling him/herself. Though the human race had a long history of making dolls, it was not easy to make dolls act or behave like humans.

In recent days, some of the people tried to achieve this will inside the virtual space using computer simulation techniques. And several computer raising simulation games, such as 'Princess maker[7]' and 'Creatures[8],' were developed and introduced to the PC game market. Today, lots of raising simulation games are developed and sold in the game market and it is one of the popular genre in the game industry.

Actually, the first game for raising a pet was built on a portable mini video game device, and it was called 'Tamagochi' which was developed by Bandai[9], a toy company in Japan. With 'Tamagochi' children could raise a virtual pet with their mini video game device. It asks the user to feed when it feels hungry, needs remedy when it is sick, grows as the time flows, and can play games with users.

Though the users were able to raise and take care of virtual pets with these kind of video games, users were not allowed to touch and fully interact with their pets, users were only allowed to watch them through a display device. Inspired by the desire of touch, a lot of interactive dolls were designed, some of them are shown in figure 1. Today, we have so many toys moving around using electronic components, and they vary from one that only repeats a simple movement to complicated ones that show complex responses to the user. Among these, AIBO is the most advanced one, and rather than treating it as a toy, it is almost a full computer system embedded autonomous robot.

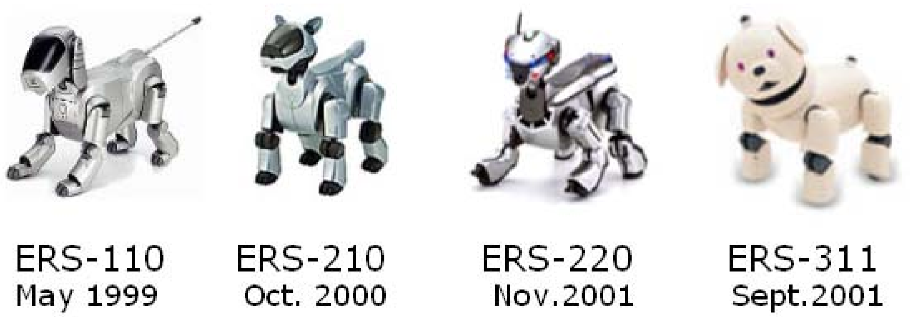
[Figure 1] Interactive toy pets: Furby(top left), i-Cybie(top right), Tekno and its variations(bottom left) and NeCoRo(bottom right).

3. AIBO

AIBO is an entertainment robot developed by Sony corporation[1]. The word 'AIBO' is an acronym of 'Artificial Intelligence roBOt,' and it has a same pronunciation with a Japanese word '相棒(あい-ぼう),' meaning 'close friends.' As its name suggests, AIBO is developed as an robot with artificial intelligence, and its purpose if to be a good friend of human beings, just like real pets. Figure 2 shows the appearance of AIBO.

[Figure 2] AIBO

From 1999, when the first AIBO was born, AIBO has been refined and improved and now we have several models. Figure 3 shows the product line up of the AIBO and now only the three latest models are available to the customers at the market. ERS-100 series were sold out about 50,000 units in Japan, the U.S., and Europe around December of year 2000 [10].



[Figure 3] The product line up of AIBO

3.1 Specifications of AIBO

Here we describe the detail hardware and software specifications of AIBO.

The hardware of AIBO can be separated into several units: core, head, tail and 4 leg units. The core unit is a box shaped unit, which is placed inside the body of AIBO, and all the other units

are attached to this. It contains computing equipments such as CPU and memories. It uses 64-bit MIPS RISC CPU chip and has a 32MB SDRAM and a built in real time clock. Sony's 'Memory Stick[1]' media is used as a secondary memory device and a PC card slot(type-II) is supported for connecting additional peripheral devices. Usually the PC card slot is used for wireless LAN.
While the core unit contains computing equipments, other units contain mechanical parts and sensors for gathering inputs from the real world. AIBO has many motors to move its body parts and it supports up to 20 degree of freedoms on legs, tail, head, mouth and ears. AIBO also has a lot of sensors: a CMOS camera on its head for sensing visual information, a pair of stereo microphone on its ears, IR sensors for sensing distances from the obstacles, acceleration and vibration sensors for detecting the movement of the body, and finally temperature and pressure sensors for sensing the touch of the users. Mechanical devices and sensors are connected to the core unit through a bus[11]. Table 1 contains the detail hardware specifications of AIBO ERS-210 model. For detail illustration of AIBO's features, refer to the appendix A.
AIBO is based on the OPEN-R architecture, Sony's standard open architecture for entertainment robots[12], and it runs on Aperios[13], an real-time operating system kernel of Sony corporation. There are two types of application software for AIBO: an AIBO-ware and PC applications for AIBO. AIBO ware is loaded in a Memory Stick media and runs on the AIBO itself. On the contrary, PC applications for AIBO run on a conventional personal computer. There are several AIBO-wares and PC applications available from Sony and it will be described in detail later.
AIBO also has software development tools: the R-CODE interpreter and the OPEN-R SDK. The former is easy to learn and use and it is for general users. The latter is for professionals and it supports full features for controlling and accessing the AIBO hardware. We will go into details of both development environments later.

3.2 Functions of AIBO

The basic function of AIBO is similar to that of computers. AIBO gathers information from all the sensors on it, and processes it, and then gives output using its output devices, like controlling motors, etc.
The designers of AIBO have defined basic instincts of AIBO: love, curiosity, movement, hunger & sleep, and AIBO behaves under these instincts. It tries to fulfill these five desires and feels good if those are achieved, or bad if not. AIBO can also express its feelings through body motions, tonal language and LED display, which is placed on its head.
As computers behave according to the software it is running, AIBO's behavior is also determined by what software(AIBO-ware) is loaded on it. According to the software it runs, AIBO could be able to learn how the behave, recognize voice or tonal language commands, sing a song, track specific objects, take pictures, recognize the face of its owner, or keep houses from

intruders. AIBO even can play a robot soccer, there is also an international robot soccer game league for AIBO[14][15].

Detail functions of each AIBO-ware and PC applications are described in appendix B and C. Also voice commands that AIBO can recognize are listed in appendix D.

[Table 1] Detailed Specification of AIBO ERS-210 hardware.

"AIBO" ERS-210:

Components	Body, Head, Tail, Leg x 4, "Removable"
Program Storage Medium	Memory Stick for AIBO
Movable Parts	Mouth: 1 degree of freedom Head: 3 degrees of freedom Legs: 3 degrees of freedom x 4 legs Ears: 1 degree of freedom x 2 ears Tail: 2 degrees of freedom Total: 20 degrees of freedom
Input/Output	PC Card slot Type2 In/Out Memory Stick slot In/Out AC IN Power Supply connector Input
Image Input	CMOS Image sensor
Audio Input	Miniature Microphones
Audio Output	Miniature Speaker
LCD Display	Time, Volume, Battery condition
Built-in Sensors	Temperature Sensor Infrared Distance Sensor Acceleration Sensor Pressure Sensors (Head, the Back, Chin & Legs) Vibration Sensor
Built-in Clock	Date & Time
Power Consumption	Approx. 9W (tentative) Standard operation in autonomous mode
Operating Time	Approx. 1.5Hours (tentative) Standard operation in autonomous mode
Charging Time	Approx. 2 hours (with a supplied AC Power Adaptor and the "Lithium Ion Battery pack" ERA-201B1)
Dimensions(W/H/D) (not including ears and tail)	Approx. 152 x 281 x 250 mm (Approx. 5.98 x 11.06 x 9.84 inches)
Weights(including a battery and a memory stick)	Approx. 1.5kg (Approx.3.3lb)
Color	Gold/Silver/Black
Supplied Accessories	AC Adapter, Lithium Ion Battery Pack ERA-201B1(1), Ball, Documentation, etc
Operating Temperature	41 F to 95 F (5C to 35C)
Operating Humidity	10% to 80%

3.3 Programming AIBO

As previously mentioned, there are two ways to write a program for AIBO, one is using R-CODE and the other is using OPEN-R SDK.

R-CODE[16] is a simple programming language, easy to learn and use. It looks similar to BASIC and runs on R-CODE interpreter. Users can write new simple behaviors of AIBO using R-CODE. In R-CODE only 16-bit integer variables are allowed and users can obtain sensor data from system predefined variables. R-CODE supports basic arithmetic operations, control statements, stacks and subroutines. Users can write R-CODE programs using conventional text editors. To execute the program users must copy the program into a Memory Stick that has R-CODE interpreter on it.

Figure 4 shows an example program written in R-CODE. The program tells AIBO to play a motion named 'Akubi_sit' when a tonal command ID number 1 is recognized.

Full list of R-CODE commands are given in appendix E. R-CODE interpreter can be freely downloaded from the website[16].

```
:100
IF:=:Melody_id:1:200
GO:100

:200
PLAY:AIBO:Akubi_sit
SET:Melody_id:0
GO:100
```

[Figure 4] An example R-CODE program

OPEN-R SDK gives the user full access and control of AIBO. It is an object oriented software library written in C++ language, and consists of OPEN-R APIs. It is also free for downloading from the website[17]. It allows the user to control each joints and other display devices, such as LED displays and speakers. Users can get raw data from all sensors, e.g. image data from camera, wave formatted sound data from microphone, etc. It also has APIs to support networking(IPv4), so the users can easily utilize networking devices, such as wireless LANs. Unfortunately, high level functions, such as gaits, voice and object recognition functions are excluded.

Each program modules written for AIBO run concurrently and they are able to communicate with each other. For example, motion control module may ask the voice recognition module, whether a specific command was recognized, before playing a specific motion.

Each program modules are complied separately and the files containing each complied modules are loaded into a 'Programmable AIBO Memory Stick' in order to be executed on AIBO. The 'Programmable AIBO Memory Stick' contains Aperios(the OS kernel of AIBO) and when the AIBO is turned on it boots up and loads all the program modules on the Memory Stick into the memory and executes them.

For further information about programming with OPEN-R SDK, please refer to 'OPEN-R SDK Programmer's Guide[12],' which is available at the website[17].

4. Improving AIBO with Artificial Intelligence Technique

Although AIBO already has a lot of features that real pets have, more sophisticated behaviors requiring intelligence is expected in order to help users in their daily lives. Real pets do not only response to lovely touch of its master, but they can also help them, e.g. carry small objects, find something or somebody the master is looking for, etc.

Here we suggest two new behaviors for AIBO, those are seemed to need artificial intelligence techniques to carry out. These behaviors could be implemented as a program module using OPEN-R SDK, and are feasible enough to cope with modern artificial intelligence techniques.

4.1 'AIBO, come here!'

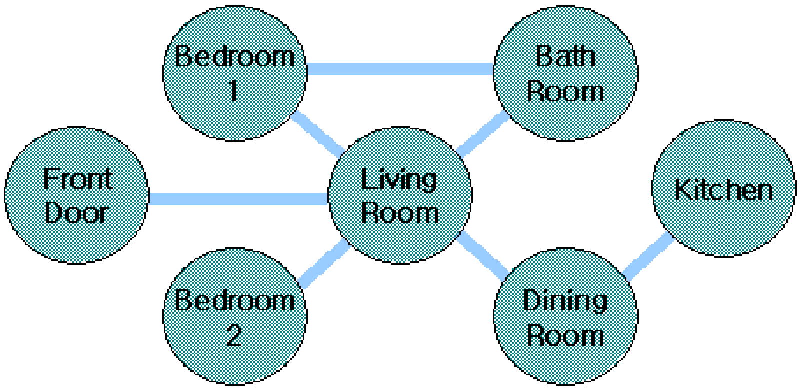
The first new behavior we suggest is named 'AIBO, come here!' As its name shows, AIBO may come to its owner wherever he(or she) is. The current implementation of AIBO also has a similar function that lets AIBO to head to the owner whenever he(or she) calls it. But, since the behavior is currently implemented to simply turn around to where the calling sound came and find the owner there, AIBO will fail to find its owner when the owner is out of sight, e.g. in the other room. To over come this defect, AIBO needs to know the map of the whole house. Then if AIBO fails to find its owner, it may ask the owner where he(or she) is and may find the way and come to the room where he(she) is in.

To achieve this, some new basic functions are needed: map construction, room recognition and way finding.

There are three ways to construct a map. First, the owner can manually construct the map using some software tools, and upload it into AIBO's memory. This is the most naïve approach to map construction, but it is not a natural way at all.

Another approach is to make the process fully automated. AIBO may construct the full map of the house while it wanders around the house. In order to make this work out, AIBO must have full object recognition function rather than the current implementation, which is recognizing simple shapes and colors. In addition, it may also require common sense to tell where the room is, e.g. whether it is a bathroom or a bedroom. This seems quite difficult with current AIBO's computing power and modern artificial intelligence techniques.

Finally we suggest a semi-automatic approach. This is a similar way how people show their houses to guests. The owner may take AIBO to each place of the house and tell where it is. A map can be represented with a graph, each nodes representing a room, and each link representing a door. Figure 5 shows an example map represented with a graph.



[Figure 5] An example map represented with a graph.

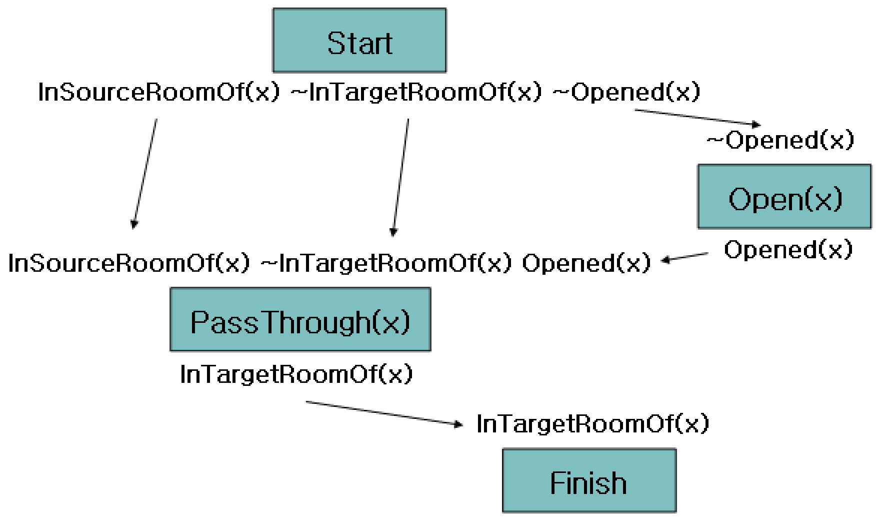
There is one thing missing to realize the semi-automatic approach. AIBO needs to recognize each room. Though AIBO has an object recognition function, the current computer vision technique seems to be insufficient to recognize each room, since there are lots of object in it and their configuration changes continuously. As an alternative to using full object recognitions, we suggest to use visual tags as an aid for object recognition. An appropriate solution for visual tagging technique can be found from one of the Rekimoto's work: the CyberCode[18]. Since CyberCodes give information about position and orientation of the tag in addition to IDs, AIBO may utilize this information, too.

Tagging doors rather than rooms have lots of advantages. Although AIBO can recognize each room, and it is able say where it is, AIBO have to find doors to move to another room. In addition, since a room is a large space for AIBO, it may fail to find tags within the room, and where to place a tag in the room has no clear answer. If doors are tagged instead of rooms, it will be much more efficient and clear for recognition. Additionally it has another advantage that users may show and tell AIBO only the doors(ways) to each rooms, and carrying AIBO from one room to another will be not necessary.

Current voice recognition function of AIBO seems to be sufficient for recognizing the command 'AIBO, come here!' and other words needed to indicate a place, such as 'bedroom' or 'kitchen.'

Finally, AIBO has to find the way from current position to the target position using the map. This behavior can be separated into two steps. First, AIBO must find the best path, and then it must move from one room to another, recognizing the doors along the path previously found. Finding a best(shortest) path in a graph from one node to another is a basic search problem that has been treated in artificial intelligence field for a long time. And since most of the house has not much complicated structure, the map graph would be simple enough to treat it with simple searching algorithms, such as depth-first-search. As a result, a sequence of rooms(or doors) will be given and now AIBO has only to follow the resulted path.

To follow through the path, AIBO first has to find the tag on the door and recognize it. After recognition, it has to pass trough the door and then find the next door(actually the tag). During this some unintended situations may happen, e.g. the door could be closed. To resolve these situations continuous planning must be carried out. The planning could be done using STRIPS operators and POP algorithm. Figure 6 shows an example expected result of planning.



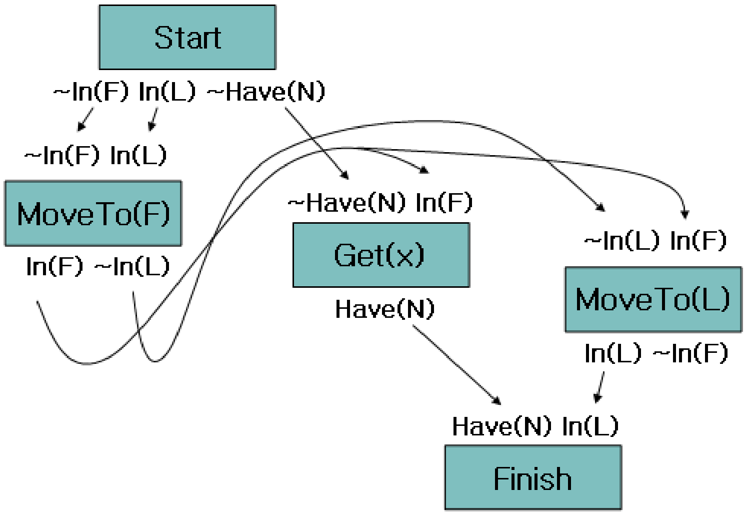
[Figure 6] An example expected result of planning for passing through a door

4.2 'Go, get the newspaper.'

This new behavior also can be implemented using artificial intelligence techniques that we previously described. Basically, it is same to way finding behavior except carrying an object. So we only need an additional function to recognize and carry an object. Current AIBO implementation has a physical problem for carrying objects. It has neither a manipulator to pick up an object nor a space to put it on. To solve this problem, maybe the mouth joint could be redesigned to make it appropriate to carry things.

Since object recognition of current implementation is much limited, we suggest to design an easy recognized basket for AIBO and carry things within it, instead of carrying things directly.

The same way finding and planning techniques used before could be applied here again. Figure 7 shows an example expected result of planning for this behavior.



[Figure 7] An example expected result of planning for carrying a newspaper from the front door to the living room.

We can think of more behaviors similar to these two, e.g. 'Take this to the living room', 'Take the newspaper to daddy', etc. And these behaviors also could be easily implemented, utilizing the previous way finding and planning functions we've implemented.

5. Conclusion

We've reviewed the current state-of-the-art of AIBO, a robot pet developed by Sony corporation. AIBO was sufficiently equipped to mimic real pets, it had lots of sensory devices, fully motor-controlled body parts and enough computing power. There were lots of softwares enabling AIBO to mimic behaviors of the real pets. Software development environments were also provided for advanced users and users were allowed to program new behaviors for AIBO with it. According to this, we suggested to add some new behaviors those are a little bit sophisticated, requiring artificial intelligence techniques to achieve them. Outlines were given for applying artificial intelligence to implement those behaviors.

References

[1] http://www.sony.net

[2] http://www.aibo.com

[3] http://www.furby.com

[4] http://www.tekno-robot.com

[5] http://www.tigertoys.com

[6] http://www.necoro.com

[7] http://www.gainax.co.jp

[8] http://www.creaturelabs.com

[9] http://www.bandai.co.jp

[10] *AIBO ERS 210 Entertainment Robot 2nd Generation "Basics" Reference Guide: Frequently Asked Questions*, Sony Corporation, Dec. 19, 2000.

[11] *OPEN-R SDK: Model Information for ERS-210*, Sony Corporation, 2002.

[12] *OPEN-R SDK: Programmer's Guide*, Sony Corporation, 2002.

[13] http://www.jp.aibo.com/openr/e_noregi/faq.html

[14] http://www.robocup.org

[15] Wako Tojima, '로봇의 시대(Robot Kyoshitu),' Science Books Co. Ltd., Seoul, Korea, 2002.

[16] http://www.us.aibo.com/clubaibo/rcode.php

[17] http://www.jp.aibo.com/openr/e_regi/index.jsp

[18] Jun Rekimoto and Yuji Ayatsuka, 'CyberCode: Designing Augmented Reality Environments with Visual Tags,' in *Proceedings of Designing Augmented Reality Environments*, 2000.

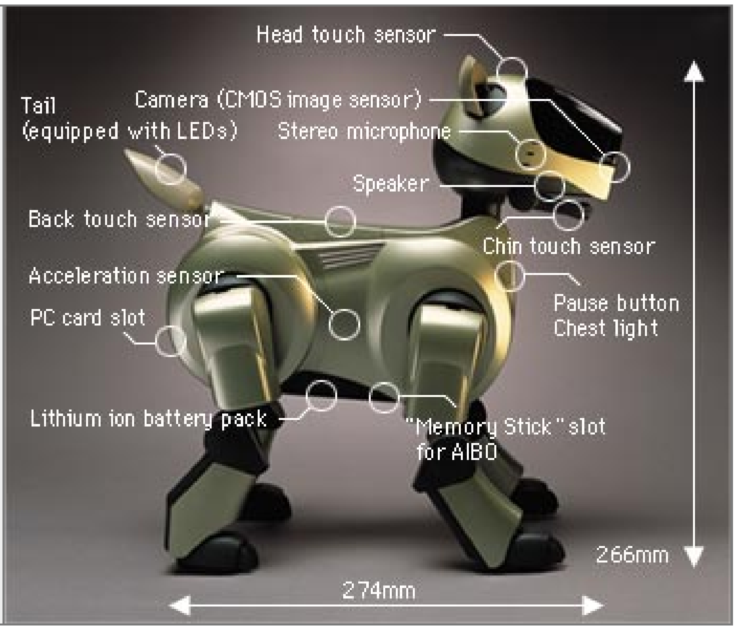
Appendix A Feature illustrations of AIBO

The main features of the "AIBO" ERS-210:

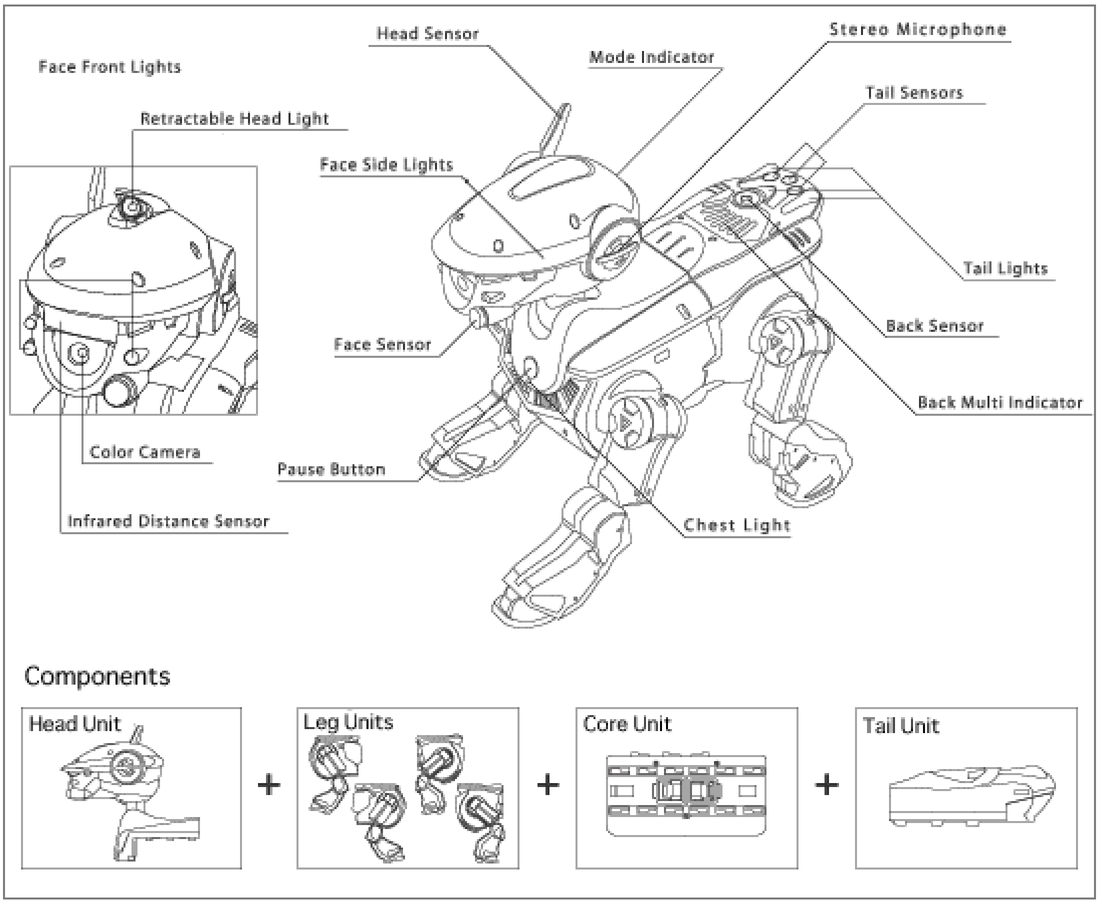
- Head touch sensor
- Camera (CMOS image sensor)
- Stereo microphone
- Speaker
- Chin touch sensor
- Pause button Chest light
- Back touch sensor
- Acceleration sensor
- "Memory Stick" slot for AIBO
- Lithium ion battery pack
- PC card slot
- Tail (equipped with LEDs)
- Joints (20 "degrees of freedom")

Weights (inclusing a battery and a memory stick) : Approx. 1.5kg (Approx.3.3lb)

Dimensions (W/H/D) (not including ears and tail) : Approx. 152 x 281 x 250 mm (Approx. 5.98 x 11.06 x 9.84 inches)



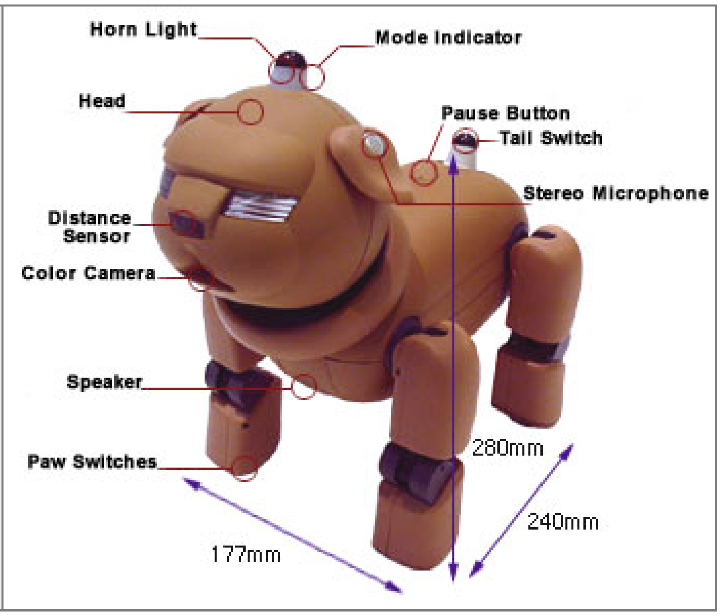
AIBO "ERS-220" Exterior



AIBO "ERS-311 / ERS-312 / ERS-31L" Exterior * Photo is "ERS-31L"

- Mode indicator
- Horn light
- Head
- Stereo microphone
- Distance sensor
- Color camera
- Speaker
- Paw switches
- Back light
- Pause button
- Tail switch

Mass Approx. 1.5 kg (3 lb 3 oz) (including the battery pack and the "Memory Stick")
Dimensions Approx. 177 ' 280 ' 240 mm (6 ' 11 1 / 8 ' 9 7 / 8 inches) (w/h/d)



Appendix B List of AIBO-wares

AIBO Life

- Raise from toddler to child, teen & adult
- Interacting with AIBO accelerates the maturing.
- Teach Actions
- Give a name to AIBO
- Tell your name
-

Hello AIBO

- Fully matured adult AIBO
- 75 Voice Command Recognition
- Mimicking
- Social behaviors

Party Mascot

- Games
- Sing a song and dances

AIBO Speed Board

- Skate with voice commands
- Record and play routines

AIBO Explorer

- Fully matured AIBO
- Surveillance mode
- Hyperactive Boost mode

AIBO Recognition

- Fully matured AIBO
- Owner Recognition
- Self Recharging on Energy Station

Appendix C List of PC application softwares for AIBO

AIBO Fun Pack

- Check AIBO status
- Download photos

AIBO Messenger

- Notice e-mail received
- Read an e-mail
- Read favorite websites

AIBO Navigator

- Control and Navigate
- Wireless LAN
- Video and sound monitoring
- Take Picture
- Send commands through wireless LAN
- Force feedback Joystick support

AIBO Master Studio

- Create and edit original sound, motion, LED data and Beha
- Voice recognition
- Wireless LAN debugging
- Import/Export Actions for other AIBO wares

Appendix D Voice Commands

Calling out

- AIBO (or Registered name)
- What's your Name?

Greetings

- Good morning?
- Hello
- Good bye/Bye bye
- I'm here
- Say hello
- Shake
- Other hand / Other paw
- Shake hands

Admonishing, praising and encouraging

- Good boy! / Good girl!
- Don't do it / Don't do that
- Go for it

Questions

- How old are you?
- Are you alright?
- Are you tired?
- Are you Hungry?
- I love you / I like you

Commands

- Take a picture
- Sit down
- Stand up
- Go forward
- Go Back
- Stop
- Lay down
- Go right / Turn right
- Go left / Turn left
- Walk around
- Get up
- Find the ball / Where's the ball
- Kick the ball
- Let's play / Let's talk
- Be quiet
- Let's dance
- Take a pose / Pose for me / pose
- Karate chop

Appendix E List of R-CODE commands

Control Structures

:	defines a line to be a label.
GO	jumps the process to the specified label.
IF	conditional test statement.
SWITCH	branches the process.
CSET	sets a context value.
CASE	branches the process.
FOR	generates a FOR loop.
NEXT	terminates a FOR Loop.
WHILE	generates a WHILE loop.
WEND	terminates a WHILE Loop.
REPEAT	generates a REPEAT loop.
UNTIL	terminates a REPEAT Loop.
DO	generates a DO loop.
LOOP	terminates a DO loop.
BREAK	breaks a loop.
CALL	calls a subroutine.
ARG	fetches an argument of a subroutine.
RETURN	returns from a subroutine.
RET	returns from a subroutine (for context version).
ONCALL	registers an interrupt routine.
RESUME	returns from an interrupt routine.

Variables

GLOBAL	declares a variable as a global variable.
LOCAL	declares a variable as a local variable.
LET	assigns a value (simple assignment).
SET	assigns a value (assignment with a special function)

Appendix E List of R-CODE commands (continued)

Operations

ADD	Addition
SUB	Subtraction
MUL	Multiplication
DIV	Division
MOD	Remainder
AND	Logical AND
IOR	Logical OR
XOR	Exclusive OR
NOT	Negation
LAND	Logical AND (Boolean operation)
LIOR	Logical OR (Boolean operation)
LNOT	Negation (Boolean operation)
EQ	Equal to
NE	Not equal to
LT	Less than
LE	Less than or equal to
GT	Greater than
GE	Greater than or equal to
RND	Random number

Stack Operations

PUSH	pushes data onto a stack.
POP	pops data from a stack.
DUP	copies the top entry on a stack.
JT	jumps if the top entry on a stack is true.
JF	jumps if the top entry on a stack is false.