Using Accelerometers to Command a Cleaning Service Robot

Armando Sousa[1,2], Luís Paulo Reis[1,3]

[1] FEUP – Faculty of Engineering, University of Porto, Portugal
[2] INESC-P – Instituto de Engenharia de Sistemas e Computadores do Porto, Portugal
[3] LIACC – Artificial Intelligence and Computer Science laboratory, Porto, Portugal
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
{asousa, lpreis}@fe.up.pt

Abstract. This work studies the effectiveness of using accelerometers to control a service robot. Two modes are proposed, a steering wheel and a movement identification mode. The validation platform is an autonomous cleaning service robot that still needs a local Human Robot Interface. For convenience, accelerometer readings are obtained from the remote command of the Nintendo Wii console. The implementation of the Steering Wheel is based on a Fuzzy Logic controller and the implementation of the movement identifier uses Case Based Reasoning approach upon identified characteristics. Results from experimental tests and user surveys evaluate the effectiveness and usefulness of the presented approach on low skilled users.

Keywords: Human-Robot-Interaction; Service Robot; Robotics; Accelerometers

1 Introduction

With the price of modern day robotics dropping and its usefulness rising, the use of service robotics as human-robot working partnerships will become more common. The search for robotic autonomy is one of the current trends in order to make the robots intelligent and useful to the human society. The notion of usefulness of a service robot includes robustness, mission efficiency and the need for efficient interfaces to command the robot and treat it as a partner in a working team of robots and humans.

The interfacing device to the service robot should be robust, easy to use and preferably inexpensive. As service robotics become more sophisticated, the users of the apparatus are likely to become less skilled thus imposing additional importance on the easiness of usage of the interfacing device.

1.1 Goal of present study

This paper explores accelerometer readings to command a robot, specifically a partially autonomous cleaning service robot. For easiness of data collection, these accelerometer readings were obtained by the Nintendo Wii's [9] remote controller, short named "Wiimote" but many other consumer "gadgets" include accelerometers such as cellular phones among others (a somewhat long list of consumer gadgets with accelerometers is available at [25])

1.2 Structure

After the current introduction, Section 2 deals with related context and research. After that, chapter 3 shows details for the actual implementation. Results are presented in Chapter 4 that led to the conclusions and future work presented in the last chapter.

2. Context and related work

The current chapter will lead to the need to develop the presented work while presenting other interesting research. The work presented here was initially published in [26] and further information can be retrieved at [2].

2.1 Classical Robotic User Interfaces

Classical solutions for Human Machine Interface use a Graphical User Interface (GUI) where the user views commands and often introduces commands through touch screen, joystick, mouse and/or keyboards of several kinds [5]. Frequently these devices were local and wired to the robot.

2.2 Modern interfaces and local command and supervision

Present day autonomous service robots frequently feature some type of wireless communication strategy for development, configuration and eventually remote operation. These operations are typically done in a desktop, laptop or tablet PC frequently using standard Wi-Fi networks that connect to the robot. This type of interface is interesting and is typically done at a distance but has its limitations as some pragmatic commands benefit from locality with the robot, for instance, in a service cleaning robot, a frequent command would be "start cleaning here".

As the service robot is put into present day (or near future) exploration, it is likely that the human worker that teams-up with the robot (or team of robots) tends to be a low skilled person and wants a pragmatic interface to the robot where a few repetitive commands can be issued. The robot or team of robots will probably be cost limited and have a reasonably high level of autonomy but there is still the need for a human supervisor that, besides other tasks, validates the quality of the work done and solves unforeseeable situations like failures, inaccessibilities, etc. Such human worker probably does not want to carry a laptop nor other complex, sensible, expensive and perhaps heavy device.

As mentioned before, it is interesting to have a Local Human Robot Interface (LHRI) to issue a small number of commands that benefit from locality. It thus appears interesting to study the effectiveness of using inexpensive COTS hardware and innovative types of interfaces, even if some limitations are recognized.

2.3 Robotic autonomy and service robots

As autonomy rises, some service robots may pass on extensive Graphical User Interface as Human-Robot communication is less frequent. For an autonomous robot, communication needs include changing mission or operation parameters: redefining mission goals, operation modes or other internal parameters (and such operations are not done frequently).

Changing a small number of parameters may not require a full featured GUI if other interesting interface methods are found. An autonomous Service Robot should be aware of the presence of a human and should accept commands coming from that person if it is a valid supervisor. Present day service robots make up a team of human and robot (or more likely, in future, a team of robots and humans).

2.4 Target Platform – Service Cleaning Robot

The presented work envisages a real world application target that is the Service Cleaning Robot [1]. This robot is built with great flexibility at interesting costs as permanent concerns. It also uses mostly commercial off the shelf (COTS) hardware that is intended to produce a suitable performance at a very interesting costs. At present time, only one prototype is built but the idea of the project includes studying team work of several robots. The prototype is functional and uses wheel chair drive. It includes a fully passive cleaning apparatus purchased at a discount store and adapted to become an integral part of the robot. The processing platform is a common PC and only a limited number of pure distance sensors are available for operation. The project is connected to the structured Wi-Fi network and the robot's state can be viewed and commands issued remotely.

2.5 Needs for the LHRI

Aside from remote operation, the need was identified to have a LHRI for a limited number of commands. Such commands would be given by a human supervisor near the robot. It appears it would be would be interesting to “wave” commands to the robot. The registered needs include easily relocating the robot, starting and stopping missions (for example special cleaning of current area), error recovery, etc [1][26][14]). These needs benefit from a supervisor close to the robot and this means that implementing an effective LHRI is important [21][22].

2.6 Accelerometer background

An adequate solution is "waving" commands with a "magic wand" that leads to registering accelerometer readings over time. With the development of micro electro-mechanical systems (MEMS), accelerometers are becoming more frequent in user grade apparatus (cellular phones, recreational devices, etc). The performance of such devices is increasing to deliver devices of higher and higher sensitivity at very interesting costs [25][11].

Such user "gadgets" offer interesting hardware with very reasonable features and some such devices are very easy to work with because little or no hardware development is necessary. Of course it is necessary to be able to access to the raw accelerometer data easily.

Returning to the notion of "waving" commands at the robot, it would be interesting to have a small, rugged, inexpensive "command stick" with 3D accelerometers, preferably with wireless communications and some means of feedback.

2.7 Wiimote

The Nintendo Wii console has a revolutionary remote control device (abbreviated as "Wiimote") that is a very interesting solution because it includes a 3D accelerometer device and BlueTooth (BT) communications. It can be considered Commercial off-the-shelf (COTS) as it is widely available at European consumer stores at approximately 40€ (2009 pricing). The Wiimote and some of its interesting features are shown in Figure 1.

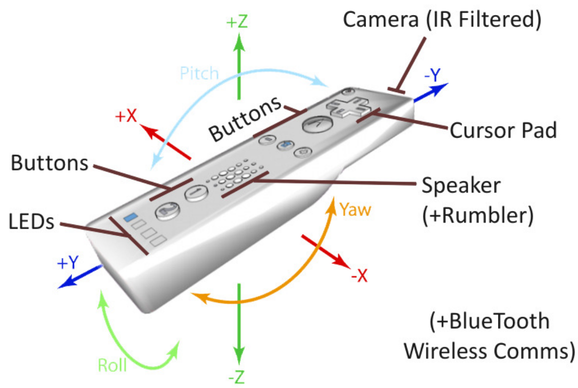

Fig. 1. Wiimote internals and defined axes

Another reason for the ease of use of the Wiimote is that although Nintendo does not release communications protocols, the user community has been successful in reverse engineering portions of the protocols and free and open source interface libraries exist. The used library for the presented work is the libwiimote (see references section).

2.8 The ADXL330 inside the Wiimote

The particular accelerometer chip used inside the Wiimote is the Analog Devices ADXL330 3D accelerometer [3] that has a full range measuring of ±3g, approximately equivalent to $\pm 30 m.s^{-2}$. This device is sampled at 8 bits and samples sent over BT link at approximately 100 reports per second.

The ADXL330 inside the wiimote is able to measure motion. The wimote and associated accelerometer ADXL330 axes are shown in figure 1.

The internal workings of the ADXL330 chip may be very briefly introduced as a set of micro machined springs that suspend a mass of the same kind; accelerations deflect the moving mass and unbalances a differential capacitor that produces measurements resulting in an output whose amplitude is proportional to acceleration. Negligible aging problems and superior temperature immunity are claimed by the maker (see the references section for a link to the datasheet).

The mentioned workings explain that accelerations applied will be measured but if no force is applied, then gravitational force may be measured. Gravitational forces will be measurable as about one sixth of the full range of the sensor.

2.9 State Of The Art

Service robots are becoming common, even in commercial grades (example: "Roomba") but the configuration and navigation options are very limited and many times fully custom built interfaces [8].

The goal of this paper is to study the usefulness of hand and arm movement detection with the Wiimote's accelerometer; latter on, the accelerometer is also use as a wireless steering wheel.

The presented ideas may, of course, be adapted for any other types of interfaces such as the ones mentioned in [23]. Adapting movement analysis to a given application or a given service robotic system is generally a matter of imagination to explore additional, not commonly used sensory data that is easily available from the Wiimote as general users are not used to using wireless "command sticks". Some authors have previously researched gesture recognition for example using tracks on sensitive pads [6], vision systems [20][24] and time of flight cameras [15]. These authors explore creation of interfaces on sensible and expensive systems, less adequate for a working human to use in the command of a robot.

[5] and [16] present interesting work where wireless sensors are read and some conclusions drawn but the current work extends the mentioned works and presents ideas for the platform at hand and additional unskilled user tests. The current work also uses different implementation based on Case Based Reasoning and Fuzzy Logic for the steering wheel application.

3. Implementation

This work explores the potential of accelerometer readings (taken from the Wiimote) as a wireless control structure, exploring a single button and the accelerometers as a consumer grade, widely available device featuring robust mechanics, interesting hardware, inexpensive, lightweight and easy to interface.

3.1 Limitations of the Wiimote

The reader should take into consideration that admittedly using the Wiimote has some limitations. The BlueTooth (BT) standard starts by "pairing" two devices together that must be done at the start of the communication. Only paired devices can communicate. This must be done once whenever a given Wiimote is to control a robot (and this must be done with each robot and each Wiimote). The BT standard plays well alongside with Wi-Fi networks but the implemented protocol does not ensure delivery of all packets, that is, no guarantees about delivering all the reports from the wiimote. The wiimote interfacing libraries are having some trouble using the loudspeaker included in the wiimote when it concerns to playing more than simple beeps. Playing audio samples for user feedback would probably be interesting but is presently not possible. The current implementation takes advantage of the features of the wiimote thus uses only beeps and LEDs for user feedback.

Even in the presence of these limitations, the Wiimote is still interesting to use as Human Robot interface device, specifically in the part of the accelerometer's readings.

3.2 Mode Selection

Current implementation has two modes: wireless steering wheel and movement identifier.

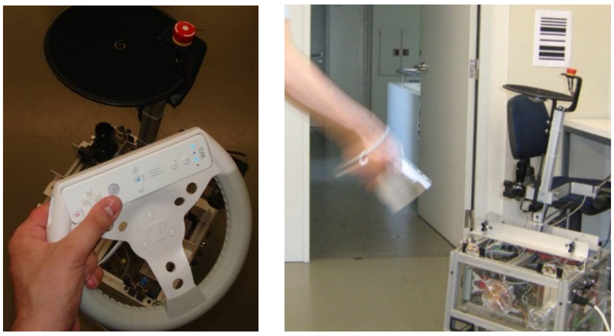

Fig. 2: Left: Wiimote as a Steering Wheel;
Right: User controlling the robot with movements of the Wiimote

In the first place, the controller of the robot was changed in order to integrate communications from the Bluetooth communications stack via the libwiimote library.

If a Wiimote is detected and the “home” key is detected, then the robot suspends previous mission and halts. The user should press buttons “1” and ”2” to awaken the Wiimote and then press the “1” button on Wiimote for steering wheel mode and the “2” button for movement identifier mode.

To inform the user of the selections and actions, each selection is followed by a single medium pitched short sound. Final mode selection that will lead to action is confirmed via a long beep. Erroneous selections activate the rumbler and sound a triple short beep.

3.3 Steering Wheel Mode

This operating mode allows the user to use the Wiimote like if it was an automobile steering wheel and its set of brake and acceleration pedals, in order to control the robot’s linear and angular velocities. The control is done by tilting the Wiimote forward or backwards and/or leftward or rightwards to increase or decrease the linear velocity (υ) and/or increase or decrease the angular velocity (ω), respectively.

As explained before, if the Wiimote is held still, the accelerometer readings determine the direction of the gravitational vector. Using these readings, a wireless steering wheel is produced. This usage of the Wiimote is natural and several games within the console world use this idea; this idea is so common that the market has produced a plastic steering wheel shaped adapter, shown in figure 2 (left).

To start the steering wheel command mode, a centre position is determined to detect each user’s normal way of handling the steering wheel. In this centre position, the robot is stopped. This calibration is done once only, at the start of the usage of this mode.

The steering wheel routines implement a MIMO non linear controller that converts the readings of the 3D gravitational vector into reference values for the robot’s linear velocity υ and angular velocity ω. The conversion process is based on a non-linear Fuzzy Logic inference system [7] based on the Mamdani controller architecture [19]. This approach is interesting due to the non-linearity associated with the analysis of the accelerations on each axis and the difficulty on establishing limits for intuitive operation.

Table 1 presents the rule base for the controller: inputs are the projected components of the vector difference from the gravitational vector to the centre position and the outputs are the robot’s velocities (υ, ω). NHT represents No Horizontal Tilt (horizontal centre) and NVT represents No Vertical Tilt (vertical centre). The implemented rules are very simple and easy to tune and create the notion that horizontal tilt makes the robot turn (controls ω) and vertical yaw alters the velocity of robot (controls υ). All movements are accounted for and it is even possible to make the robot turn over himself.

Table 1. Rules for the steering wheel’s Fuzzy Logic controller;

IF Forward THEN υ_Positive;	IF Leftward THEN ω_Positive;
IF NVT THEN υ_Zero;	IF NHT THEN ω_Zero;
IF Backward THEN υ_Negative;	IF Rightward THEN ω_Negative;

3.4 Movement Identification mode

The other working mode implemented in the LHMI was a movement identifier. In this mode the tridimensional accelerometer is used to read and save the accelerations on each axis in vectors, in order to detect and calculate the characteristics of those accelerations that would make it possible to identify which movement has been done, from a set of thirteen possible movements – an illustration of this is shown in Figure 2 (right).

The movements to be detected are:

- Linear, single stroke, over a single axis (left, right, up, down, front and back)
- Repetitive over a single axis (left and right, up and down and front and back)
- Circular on the xz plane, clock and counter clockwise.
- Square on the xz plane, clock and counter clockwise.

The acceleration's graphic of a linear movement over an axis will be similar to a one-period sine wave, due to the inversion of the acceleration at deceleration instant (braking instant). All movements have recognizable characteristics. Some movements will have more than one instants of acceleration and/or braking, thus the linear movements can be differentiated from the others through the number of maximums and minimums of their acceleration. The direction of a linear movement over an axis can be found analyzing the amplitude of the accelerations maximums and minimums. The orientation of that movement can be found through the analysis of the order in which the maximum and minimum value appear.

Samples readings for the registered accelerometers measurements for the 3 axis are shown in figure 3 for a clockwise square along the XZ plane.

Other movements are identifiable by analyzing the mentioned characteristics of the signals.

The Implementation used Case Based Reasoning (CBR) as presented by [1]. This method is interesting because it is easy to implement, light weight to execute and easy to configure. The motivation behind using this particular method is its simplicity and the expectation that with simple one time training it is possible to achieve good classification results.

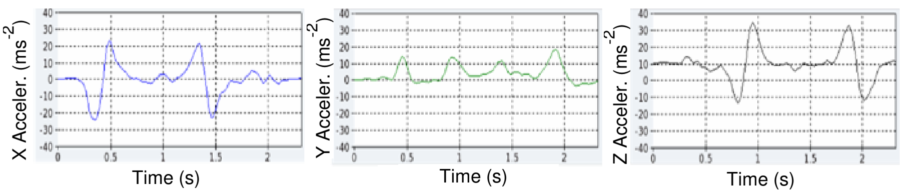

Fig. 3. Sample of accelerometer readings for a clockwise square over the XZ plane.

To distinguish all the movements, a number of very simple and interesting attributes were considered to describe all movements and that are likely to allow distinction of all movements: (i) The number of maximums and minimums of the acceleration in each axis (components); (ii) The global amplitude of maximums and minimums of the amplitude of acceleration 3D vector; (iii) The number of maximums and minimums of the acceleration components compared to the absolute value of the 3D acceleration vector; (iv) The sequence of axis where the components of the acceleration are maximum (over time); (v) The time difference between the samples tagged as maximums and the samples tagged as minimums;

These angles work as pointers to the tridimensional point in which the Wiimote's acceleration is located, and by analyzing the order in which they appear it is possible to know what the Wiimote's trajectory was.

This mode is an improvement of the steering wheel since the robot will be prepared to do a set of complex movements and/or actions that are associated with the movement the user does, thus making the mid/high level actions control simpler. A detected movement would be able to start high level task such as "follow wall to next corner" but implementation is not yet completed.

4. Results

The current chapter presents user feedback and numeric analysis of the previously presented work, improved and further tested than the work presented in [26]. Please note that the users are not the same for all modes. The selected users for the tests are all persons without special computer knowledge and all have never handled the Wiimote before. They would be comparable to the low skilled workers of a service robot.

4.1 User feedback for the Mode Selection operation

The mode selection routine is very simple. Initially it was briefly explained to 14 users. Then the users were asked to rate how hard it was to use: hard, medium or simple. 8 users rated it "Easy to use" and the other 6 "medium easy to use". User opinion's found the rumbler "interesting" and some users wanted more positive feedback for the final mode selection. Understandably, common users that have never used the Wiimote are not used to hearing feedback as beeps and LEDs from a "command stick".

4.2 User feedback for the Steering Wheel mode

The Steering Wheel allowed the user an easy way to control the robot's movement, although it was a little bit hard to stop it due to constant movement, voluntary or involuntary, of the user's hands. To make stopping the robot an easier task it was created a threshold around the υ and ω zero values. To ensure that the control is not triggered involuntarily, its start and stop is done by pressing one of the Wiimote's buttons: the "A" button.

Users were asked to rate this control method depending on the difficulty of it use: hard to use, medium or easy to use. The results were very interesting since all 14 users answered "Easy to Use".

4.3 Results for movement identification

To test the movement identification algorithm it was necessary to compile the case base with examples of all the thirteen movements that we want to identify. In order to be able to perform a complete test of this algorithm, the movements sent into the case base were from only one person.

The commands were thoroughly explained but no rehearsing of the testers was done. As stated before, all users never used a Wiimote before. The results shown in the figure 4 prove that the CBR algorithm and the attributes mentioned before can be used to correctly identify movements made by the testers. Proof of this is the algorithm's capacity to distinguish the simple movement's orientation, the capacity to distinguish squares from circles and in which way they were done.

One of the limitations of working with accelerometer readings is that concerned users use low velocities that generate low readings that, in turn, may result in bad attributes that will make the system answer incorrectly.

The conclusion from this test is that only up and down and left and right is similar for almost all untrained users (over 90% recognition success). This means that this input method for untrained user is of limited success except for the one axis, repetitive stroke movements.

Fig. 4. Correct identification percentage of the CBR algorithm.

5 Conclusions and Future Work

This paper studied accelerometer readings as a candidate device for implementing a Local Human Robot Interface intended for use with a service cleaning robot. The used data was retrieved from the "Wiimote", a promising COTS device. The Wiimote is a feature rich device, containing, among others accelerometers and buttons. Other devices could have been used.

The Steering Wheel uses accelerometers to measure the gravitational vector. This mode allows the user to control the robot like if it was an automobile with separate speed and turning controls in the wireless controller. The steering wheel has a very short learning curve to drive the robot in a demanding environment, like between people working or passing by. Even in this mode, the robot's sensors are still used to prevent bumping into things. This mode was considered easy to use by the users. Fuzzy Logic was used in the implementation of the controller.

The Movement Identifier mode in which the user's movements are identified would allow for a large variety of orders. To identify the movements, a Case Based Reasoning (CBR) algorithm was used. Except for very simple up and down and left and right repetitive movements, this method has limited success ratings in recognizing the movements of untrained users that have never used a Wiimote. Globally, the movement identifier has a 67.7% average success rate on user movements. This success rate is for a case base composed of movements from a single person only.

The global results for using the Wiimote are very satisfactory but additional work is needed to conclude some open issues like providing feedback. To improve the results, the case base should be expanded and it should be added movements made by other people, in order to add greater variety and increase classification rations.

Possible causes for misclassifications is that some users move the elbow a lot more than others and that produces very different measures (example: yawing the command left and right is different from translating the command left and right). Also, some users are afraid and use low speeds that are difficult to read and classify.

The overall conclusion is that accelerometers are interesting devices because they can easily become part of inexpensive, widely available user gadgets but interpreting commands from unskilled, untrained users is a complex task and only the simplest commands are useful.

Future work includes testing new types of movements and adding more both valid and invalid training movements to the case base.

The usage of the wiimote is interesting despite mentioned limitations. Interface improvement is probably possible by using multiple Wiimotes or controlling multiple robots using one Wiimote. The security of the system can be increased by creating a PIN code, composed by a sequence of buttons or a sequence of movements, which unlocks the interface. It would also be very interesting to allow the possibility of using the Wiimote's speaker to give feedback to the user about what the robot is doing with recorded audio.

Acknowledgements

This work has been partially funded by FCT under grant ACORD - Adaptative Coordination of Robotic Teams (PTDC/EIA/70695/2006). The authors wish to thank the Portuguese FCT for the financial support.

References

1. "CleanRob" Autonomous Cleaning Service Robot – http://www.fe.up.pt/cleanrob
2. Aamodt A. and Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches – In Artificial Intelligence Communications 7, no. 1, pp. 39-52, 1994.

3. ADXL330 http://www.analog.com/static/imported-files/data_sheets/ADXL330.pdf (visited Feb 2009)
5. Cheng Guo and Ehud Sharlin. Exploring the Use of Tangible User Interfaces for Human-Robot Interaction: A Comparative Study. Conference on Human Factors in Computing Systems - Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, Florence, Italy, Pages 121-130 2008
6. Dean Rubine. Specifying gestures by example. Computer Graphics, 25(4):329-337, July 1991.
7. Driankov, D.: An Introduction to Fuzzy Control – Second Edition, Berlin, Springer, 1996.
8. Forlizzi, J and DiSalvo, C. 2006: Interface Design and analysis Service Robots in the Domestic Environment: A Study of theRoomba Vacuum in the Home – In HRI '06, Utah, USA, March 2-4, 2006, ACM Press, New York, USA, 2006, 258-265.
9. Generic Information about the Nintendo Wii http://en.wikipedia.org/wiki/Wii [visited Feb 2009]
11. H. Luo, G. Fedder, and R. Carley 2000, A 1 mG Lateral CMOS-MEMS Accelerometer, In Proceedings of The 13th IEEE International Conference on Micro Electro Mechanical Systems (MEMS '00), January, 2000, pp. 502 - 507.
12. Helge Huttenrauch 2007. From HCI to HRI: Designing Interaction for a Service Robot. Doctoral Thesis, University of Stockholm, Sweeden, 2007
13. J. Schlenzig, E. Hunter, and R. Jain, 1994. Recursive identification of gesture inputs using Hidden Markov Models. Proceedings of the Second Annual Conference on Applications of Computer Vision, December 1994.
14. Kawanura, K., Pack, R.T., Iskarous, M. 1995: Design philosophy for service robots. Proceedings of the 1995 IEEE Conference on Systems, Man, Cybernetics. IEEE Press, 3736-3741
15. Kollorz, E., Penne, J., Hornegger, J., and Barke, A. 2008. Gesture recognition with a Time-Of-Flight camera. Int. Journal on Intelligent. System Technology Appl. 5, 3/4 (Nov. 2008), 334-343.
16. Lapides, P., Sharlin, E., and Costa Sousa, M. 2008. Three dimensional tangible user interface for controlling a robotic team. In Proceedings of the 3rd ACM/IEEE international Conference on Human Robot interaction, Amsterdam, The Netherlands, March 12 - 15, 2008. HRI '08. ACM, New York, NY, pp. 343-350.
17. LibWiimote - Simple Wiimote Library for Linux - http://libwiimote.sourceforge.net/ [visited on January 2009]
19. Mamdani, E.H. and Assilian, S. 1975: An experiment in linguistic synthesis with a fuzzy logic controller - International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.
20. Masaki Shimizu, Takeharu Yoshizuka and Hiroyuki Miyamoto 2007. A gesture recognition system using stereo vision and arm model fitting, International Congress Series, Volume 1301, July 2007, pp. 89-92
21. O. Mayora-Ibarra 2003. From HCI to HRI - Usability Inspection in Multimodal Human-Robotics Interactions, Proceedings of the 2003 IEEE International Workshop on Robot and Human Interactive Communication, 2003.
22. Perzanowski, D., Schultz, A.C., Adams, W., Marsh, E.; Bugajska, M. 2001: Building a Multimodal Human-Robot Interface, Intelligent Systems, IEEE, vol. 16, Issue 1, Jan- pp. 16 – 21, February 2001.
23. Schraft, R. D., Schmierer, G. 2000: Service Robots – A. K. Peters, 2000.
24. Timo Pylvänäinen,. Accelerometer Based Gesture Recognition Using Continuous HMMs. Lecture Notes in Computer Science, Volume 3522/2005 Pattern Recognition and Image Analysis, pp. 639-646
25. Wikipedia Accelerometer - http://en.wikipedia.org/w/index.php?title=Accelerometer&oldid=272179926 (visited Fev 2009, permanent link given)
26. Daniel Coutinho, Armando Sousa, Luís Paulo Reis, Creating a Local Human-Machine Interface for a Cleaning Robot Using the Nintendo Wii's Remote Controller, IROBOT 2008 – 3rd International Workshop on Intelligent Robotics, October 14th, 2008, Lisbon University Institute – ISCTE, Lisbon, Portugal pp. 61-70