Gaming controllers for research robots: controlling a humanoid robot using a WIIMOTE

Andrej Gams[1] and Pierre-André Mudry[2]
[1] "Jožef Stefan" Institute
Department of Automation, Biocybernetics and Robotics
Jamova cesta 39, 1000 Ljubljana, Slovenia

[2] *École Polytechnique Fédérale de Lausanne (EPFL)*
Biologically Inspired Robotics Group (BIRG)
Station 14, CH - 1015 Lausanne, Switzerland
E–mail: andrej.gams@ijs.si, pierre-andre.mudry@epfl.ch

Abstract

In this article we describe the use of a standard game console joystick, namely the Nintendo WIIMOTE, for controlling a HOAP-2 humanoid robot. We give a short overview on the use of tangible user interfaces, followed by the description of the used game controller and the measurements of some of its characteristics. We show the ease of applicability of inexpensive and robust standard game controllers for direct translation between the user's intent and the robot actions on a two-handed robot drumming task.

1 Introduction and motivations

Recent achievement in the *Micro Electronic Mechanical Systems* (MEMS) field enabled several developments in today's consumer electronic devices. For example, force plates, devices that were costing several thousands of euros only a few years ago, can now be replaced by inexpensive micro-engineered pressure sensors and be used in game consoles to exercise yoga or train aerobics.

In addition to this, recent advances in MEMS technology also permitted the development of precise, low-cost accelerometers that were rapidly adopted in a wide range of devices ranging from mobile phone to computers. Of particular interest is the appearance of such devices in mass-market products such as games consoles where accelerometers are used to extract users' hand gestures or arm movements. Thanks to their low price, wide availability and robustness, these haptic interfaces possess several key elements that make them interesting for controlling robots.

In this article, we will demonstrate the ease of applying standard game controllers for the task of controlling anthropomorphic robots. We proceed as follows: in the next section, a brief overview of existing human-robot interaction methods is presented. After that, focus is put on the WIIMOTE controller itself, shown in Fig. 1, presenting its specifications and characteristics, and the means of interfacing with a standard PC in section 3.2. Finally, applying the WIIMOTE to a drumming task in the real world is examined before concluding.

2 Related work on tangible user interfaces

The need to capture arm and hand motion is of great interest when interacting with anthropomorphic robots because this method can provide a direct translation between the user intent and the robot action. Thus, the usage of gestures for HRI is not new: several systems that use gloves [7], vision [4] or motion-capture techniques [6] have been successfully used to control robots in the past. To

Figure 1: The WIIMOTE controller

recognize motions and be able to control robots, accelerometers-based systems have also been built and discussed in several publications such as [5].

When considering game controllers in particular, several research projects have been recently using them in various scenarios such as interface for digital painting [2] or even HRI, [3] being a very interesting study of this kind of interface.

3 System design and implementation

3.1 Modern game controllers

The two main providers for accelerometer-based game controllers are Nintendo with its WIIMOTE controller and Sony with the SIXAXIS controller. When used as haptic interfaces, both possess several qualities that make them interesting to be used. Notably, due to their mass product origin, they are *inexpensive*, both controllers examined here costing around 50 € . Moreover, they are widely available and, because they are meant to be used by children, relatively *solid*.

However, if they show many similarities, the WIIMOTE only possess the capability of precise movement tracking. Thanks to an embedded one million pixels infrared (IR) *camera* placed in front the device, the controller can track several IR LEDs placed in a small plastic casing (the sensor bar) that should normally be placed under or above the TV. The presence of this capability allows precise detection of where the user points and how far away from the LEDs the device is placed, the distance between the LEDs being known. Thus, it becomes possible to get pitch, yaw and roll information for the device in addition to the three-axis accelerometers information. In addition to that, the controller also possess several buttons, a small speaker and a motor that can make it rumble.

Interfacing the WIIMOTE to a computer is relatively straightforward in the sense that its Bluetooth implementation (a BCM2042 chip from Broadcom) complies with the HID standard. As such, a Bluetooth interface and some widely-available open-source programs (that exist for different operating systems) suffice for the controller to be recognized on a standard PC. After the connection has been established, the WIIMOTE can be used as a mouse emulation or interfaced directly to get the various sensor values.

The accelerometers themselves are comprised in a standard ADXL330 chip from Analog Devices. The chip provides 8 bits of resolution on a $\pm 3g$ m s^{-2} range on each axis for a sample rate of about 100 Hz. Using these figures, it is possible to estimate the resolution limit of the device to be around 0.23 m s^{-2} per bit of information, with a 10% sensitivity. Because accelerations are measured by the forces exerted on a small moving mass suspended by polysilicon springs, it is not possible to statically measure the tilt of the WIIMOTE. However, it is still possible to get this information using the IR camera by measuring the deviation from horizontal angle between the LEDs.

3.2 Interfacing the Wiimote and the HOAP-2 robot

The HOAP-2 is a 25-DOF humanoid robot developed by Fujitsu and can be controlled using a standard PC running the *Real-time Linux* environment. In our work, we separate the tasks of robot control and data acquisition/processing, devoting each task to a dedicated computer as depicted in Fig. 2. This way, we can effectively control the robot using the operating system/environment of our choosing.

As is presented in Fig. 2, the computer running Matlab/Simulink environment connects to the WIIMOTE using Bluetooth. The GNU *Wiiuse* library provides built-in functions for connecting to the WIIMOTE and was interfaced in Simulink S-functions for the task. Thus, the data received from

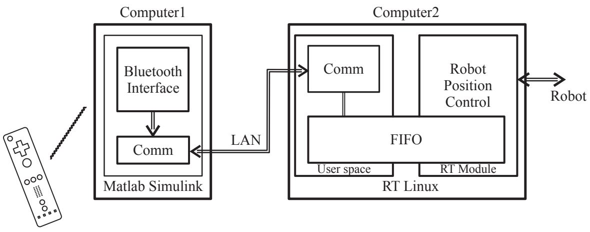

Figure 2: Control scheme for controlling the HOAP-2 robot with a Wiimote

the Wiimote are first processed locally and then sent via the network to the computer controlling the HOAP-2 robot using the UDP protocol.

When the second computer receives the data, they are copied to a FIFO structure, an operation that then triggers the transfer of data to the *Real-time module*, a component that controls the servos of the robot. The data acquisition and communication between the computers run at 100 Hz, while the real-time position control of the robot runs at 1000 Hz. Referential values for position between two consecutive packets from the control computer are generated by interpolation.

3.3 Data from the Wiimote

In this subsection we show the correlation of the data of some of the DOFs of the Wiimote. Fig. 3 shows the results of changing the orientation of the device around the axis along its length, namely the roll.

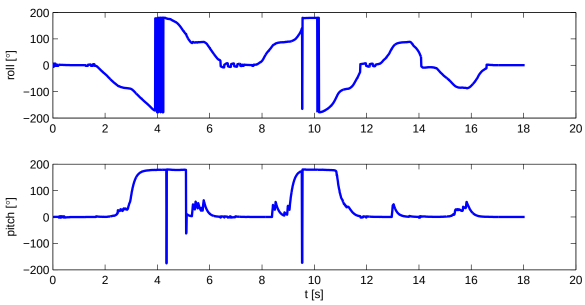

Figure 3: Correlation of roll and pitch when changing roll of the device. The Wiimote itself can output both filtered and raw data. Only the filtered data is presented.

To demonstrate the correlation of the recorded orientations we changed several times the roll of the device by 90°, while trying to maintain all other orientations unchanged. As it can be seen in the figure, when the device is rotated by 90°, the output of the pitch values changes as well. The reason for such a behavior is the well-known singularity of the Roll-Pitch-Yaw orientation. The same happens when we change only the pitch as shown in Fig. 4.

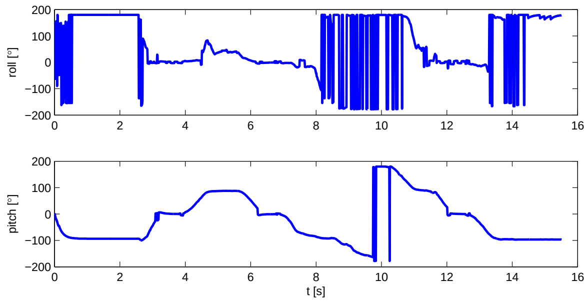

Figure 4: Correlation of roll and pitch when changing pitch of the device. Only the filtered data is presented.

To measure yaw, the Wiimote has to be pointed in the direction of the IR LEDs, as it utilizes the change of the LEDs' location to calculate its yaw. This can be clearly seen in Fig. 5, where we show the coupling of the position and the orientation.

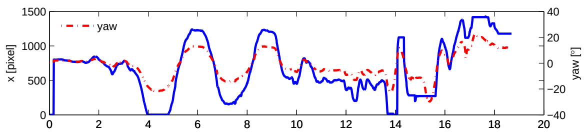

Figure 5: Correlation of yaw and x DOF of the device.

4 Drumming task

Different ways of controlling robots using the Wiimote can be implemented. One way is to use pre-recorded moves and execute them when recognized, i.e. when performing a certain move, the measured data are compared with a set of recorded moves that form a trajectory and, when a match in the database is found, the corresponding pre-recorded move is executed. Another possibility is to couple the motion of the Wiimote and the robot directly.

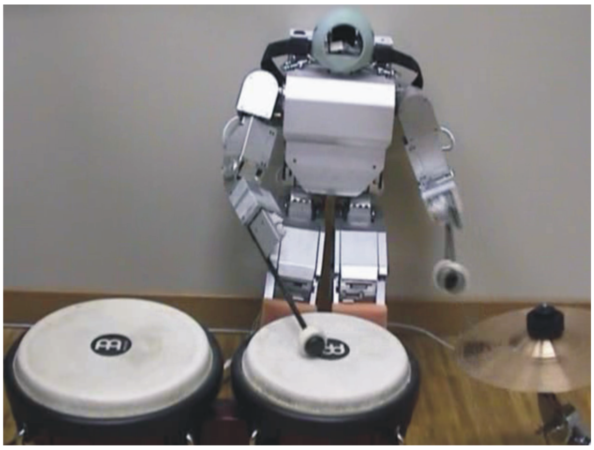

Figure 6: Two handed drumming of the HOAP-2 robot.

This approach requires some filtering of the data that would be otherwise too noisy.

We used the latter approach but we extended it to teaching the robot the moves. This was done with a system for learning and imitating trajectories of unknown frequency and waveform. The system, described in [1], also acts as a sort of filter and with it we could successfully teach the robot the desired drumming trajectories. The experimental setup was based on two WIIMOTEs, one to control each arm of the HOAP-2, managing 4 DOF per arm. The control itself was done my mapping the motion of the two WIIMOTEs to the tips of the drumsticks and, thanks to an inverse kinematics algorithm, to achieve motion in task space. Fig. 6 shows the HOAP-2 robot drumming.

5 Conclusions and future work

We showed in this article how standard consumer electronics components can be used to provide a robust and widely-accessible way to explore innovative HRI methods. Notably, we showed that tight spatial mapping between user and robot translates to a very effective control method. Using the Nintendo WIIMOTE, even though it has some drawbacks, is an effective way of intuitive control of robots. We expect that off-the-shelf electronic devices will prove even more useful in future HRI applications, one such step is also the announced add-on for the WIIMOTE, codenamed *MotionPlus*, that will provide a 1:1 tracking of arm movements thanks to an integrated micro-gyroscope.

References

[1] A. Gams, S. Degallier, A. J. Ijspeert, and L. J. Dynamical system for learning the waveform and frequency of periodic signals - application to drumming. In *Proc. of the Int. Conf. RAAD 2008*, to appear.

[2] J. Grubert, M. Hanckock, S. Carpendale, E. Tse, and T. Isenberg. Interacting with Stroke-Based Rendering on a Wall Display. Technical Report 2007-882-34, Department of Computer Science, University of Calgary, Canada, October 2007.

[3] C. Guo and E. Sharlin. Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. In *CHI '08: Proc. of the 26th SIGCHI Conf. on Human factors in computing systems*, pages 121–130, New York, NY, USA, 2008. ACM.

[4] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, P. Kiatisevi, Y. Shirai, and H. Ueno. Gesture based human-robot interaction using a frame based software platform. *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 3:2883–2888 vol.3, Oct. 2004.

[5] F. G. Hofmann, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden markov models. In *Proc. of the Int. Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 81–95, London, UK, 1998. Springer-Verlag.

[6] S. Iba, J. Weghe, C. Paredis, and P. Khosla. An architecture for gesture-based control of mobile robots. *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS'99)*, 2:851–857 vol.2, 1999.

[7] D. Sturman and D. Zeltzer. A survey of glove-based input. *Computer Graphics and Applications, IEEE*, 14(1):30–39, Jan 1994.