Locomotion and Telepresence in Virtual and Real Worlds

Alessandro Spada, Marco Cognetti, Alessandro de Luca

Locomotion and Telepresence in Virtual and Real Worlds

Alessandro Spada, Marco Cognetti, Alessandro De Luca

Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Sapienza University of Rome, Italy
{spada.aless}@gmail.com, {cognetti,deluca}@diag.uniroma1.it

Abstract. We present a system in which a human master commands in a natural way the locomotion of a humanoid slave agent in a virtual or real world. The system combines a sensorized passive locomotion platform (Cyberith Virtualizer) for the walking human, the V-REP simulation environment, an Aldebaran Nao humanoid robot with on-board vision, and a HMD (Oculus Rift) for visual feedback of the virtual or real scene. Through this bidirectional human-robot communication, the human achieves a telepresence that may be useful in different application domains. Experimental results are presented to illustrate the quality and limits of the achieved immersive experience for the user.

Keywords: telerobotic systems, personal and entertainment robots

1 Introduction

Nowadays technology allows a convergence of different, but related domains such as telerobotics [1], wearable haptics [2] and locomotion interfaces [3], virtual and augmented reality [4, 5], and human-robot cognitive and physical interaction [6]. As a result, real or virtual objects can be tele-manipulated by haptic interfaces with force feedback, visual and tactile human perception can be enhanced by augmented virtual features, remote navigation can be realized by transferring human locomotion, and so on. While the entertainment field remains a major market offering to the public new devices to enrich multimedia experience, new application areas such as search & rescue, medicine, or arts and cultural e-visits are emerging, in which the human user can benefit from interacting with a remote scene by seeing, feeling, manipulating, and moving freely inside of it.

Current research developments aim at providing the best *user immersion* properties, i.e., the perception of being present in a non-physical world, independently from the user senses and stimuli [7]. We focus here on the two complementary aspects of visual feedback to the user from a real scene or from its rendering, and of the transfer of motion commands from a human to a remote (robotic) agent. These aspects represent in fact the two opposite flows of information in a human-in-the-loop control scheme.

Concerning sight in virtual reality, innovative solutions are spreading fast thanks to the capability of building relatively small and portable devices with large computing power and high resolution, capable of deceiving the human eye. One can distinguish between solutions that are developed around the user, which require a physical structure to create the virtual environment, e.g., the Cave Automatic Virtual Environment (CAVE), and those that are in contact/worn by the user, typically Head Mounted Displays. These may show information directly on the lenses, but if the user has to be immersed in a different scenario, the HMD needs to cover the human field of view and/or be inserted in a helmet with internal focusing lenses (and some distortion). The Oculus Rift [8] adopted in our work belongs to this class. Indeed, one can project on the HMD views from a virtual scene or (possibly, stereo) images from a real camera mounted on board of a mobile robot. In our setup, we used V-REP [9] as a 3D simulation environment, capable of reproducing accurately also the presence of virtual vision sensors.

In the other flow direction, the most intuitive and effective way to transfer motion commands to a robot is to replicate (through optimization techniques) the motion of a human user by precise mirroring [10, 11], rather than by using more conventional haptic devices, e.g., a joy stick for the direction/speed of walking and function buttons for stopping and starting [12]. To this purpose, tracking of the whole-body human motion relies on motion capture technology, e.g., the Optotrak in [13] or the Xsens in [14], possibly removing also the need of body markers as in [15].

However, the above works do not deal explicitly with robot navigation. In [16], the authors have used used a free-locomotion interface (a "caddie") for controlling a mobile robot by pushing its virtual representation displayed on a large projection screen. When the target is a humanoid robot that should explore a remote environment, like the Aldebaran Nao adopted in our laboratory, one may wish to replicate the actual locomotion of the human lower limbs. A simple solution is to extract the linear and angular velocity of a human walking on a locomotion platform that keeps the user approximately in place. Among such locomotion platforms, we can distinguish actuated treadmills (e.g., the Cyberwalk omnidirectional platform [3]) and passive, but sensorized supporting platforms, like the Cyberith Virtualizer [17] considered in our work. The main advantages of the latter are indeed the smaller size and its limited cost.

Based on the above analysis, we present here for the first time an original system in which a human master in locomotion on the passive Cyberith platform commands the navigation of a Nao humanoid robot, in a virtual world modeled by V-REP or in a real remote environment, while receiving immersive visual feedback on an Oculus Rift that closes the perception-actuation loop.

The paper is organized as follows. Section 2 introduces the components of the system setup and gives an overview of the adopted control architecture. More details on how telepresence was implemented are provided in Sec. 3. Section 4 reports experimental results during navigation tests, both in real and virtual environments. Conclusions and future work are summarized in Sec. 5.

2 System Setup

This section briefly describes the main hardware components of our system, see Fig. 1, together with their integration and communication.

(a) Cyberith Virtualizer (b) Oculus Rift (c) Nao Robot

Fig. 1: Hardware components of the system.

2.1 Cyberith Virtualizer

Virtualizer is a passive omnidirectional treadmill of small size developed by the austrian company Cyberith. The structure in Fig. 1a consists of a flat base plate connected to three vertical pillars holding a circular structure, which is composed by an inner ring that can rotate inside an outer one and supports an harness. The ring structure can move up and down along the pillars in order to enable crouching. Finally, the platform has a vibration unit on the base plate in order to give haptic feedback to the user.

The working principle of the device is simple and at the same time complete. It combines a low friction principle of the base plate and a set of high-precision optical sensors with a special mechanical construction, resulting in a new form of omni-directional treadmill. The user is supposed to push the hips slightly against the ring; in this way one foot slides backwards, while the other is doing a step. The harness belt compensates the remaining friction, enabling at the same time running, walking, and crouching with user confidence and stability.

The set of API developed by the company is pretty straightforward, and consists of few calls that cover every feature. The platform returns the walker hips height and orientation, together with an average feet velocity and their direction.

2.2 Oculus Rift

Oculus Rift (see Fig. 1b) is a virtual reality headset developed by Oculus VR, composed by two Samsung AMOLED screen with a resolution of 1980×1020 pixels per eye at 90 Hz refresh rate. It has headphones and a microphone, together with a solid-state accelerometer, a gyroscope, and a magnetometer. The visual algorithm for the HMD is used to render both monoscopic and stereoscopic views (see Fig. 2). In the first case, the same camera frame is shown to both eyes, applying a suitable distortion to the image. However, the depth feeling will be lost. In the second case, the frames are taken from two (real or virtual) cameras, allowing the user to appreciate distances to the surrounding objects.

The Oculus Rift works together with a stationary infrared camera placed in the vicinity, which is delivered with the visor and has also the power and transmission unit of the device. On the HMD there is a constellation of leds installed inside the coverage, thus invisible to the human eye but visible to the infrared camera. At every sampling instant, the software takes the data and couples them with inputs from the other internal sensors. In this way, the API can return the position and orientation of the user head, relative to a fixed frame placed on the infrared camera. This sequence of measurements can be usefully mapped to a desired motion for a robot head.

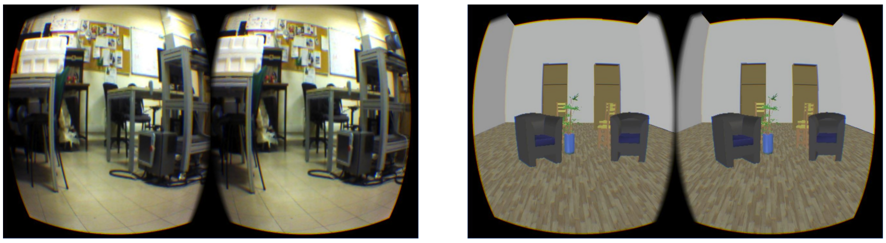


Fig. 2: Mono- (left) and stereoscopic (right) views on the Oculus Rift. The first (real) view comes from the single Nao camera, the second view from V-REP.

2.3 Nao humanoid robot

In this work, the Nao humanoid developed by the french company Aldebaran was considered as the target robot —see Fig. 1c. Nao is 58 cm tall and weighs 4.3 kg. kilograms. With its 25 dofs and the relatively large set of sensors (two cameras —but not as stereo pair, four microphones, sonar, IR, etc.), it is currently one of the most common robots for academic and scientific usage. Its proprietary OS enables the user to program Nao using several options, in particular setting joint position references or sending linear and angular velocity commands to the robot body, while taking care autonomously of its stabilization.

An important feature considered in the development of control modules is that all functions must be non-blocking, i.e., the robot should be able to receive multiple instructions, even interrupting the execution of the running one.

In order to virtualize the Nao robot, we adopted the multi-platform robot simulation environment V-REP [9] developed by the swiss company Coppelia. It is based on a distributed execution architecture, i.e., each object in the scene can be controlled with a script, a ROS node, a plugin and other options. Items can be mounted also on the robot, e.g., a depth sensor or a stereo camera that will produce in real time views of the virtual scene to be fed back to the Oculus.

2.4 Control architecture

The core of the control algorithm consists of two threads working in parallel, as shown in Fig. 3. The first thread is responsible of streaming the visual output. The user has the option to stream to the HMD the images of a stereo camera mounted on the head of the robot, or those from another internal camera if not feeling comfortable with the multidimensional sensation. The programmer may also bypass the Oculus Rift to ease debugging and visualize the stereo/mono output on a screen. The first thread has also the purpose of mapping the movements of the user head to those of the robot head. The second thread is responsible instead for the robot motion, taking the velocity inputs from the Virtualizer and sending them to the robot controller. This module is responsible also for testing collisions and, in the positive case, for triggering a vibration to the platform.

A remarkable feature is that the control laws were designed to work properly both with a simulated robot and a real one, by changing only the associated IP address. In this way, it is easy to switch seamlessly the navigation from a virtual environment to the real world and vice versa.

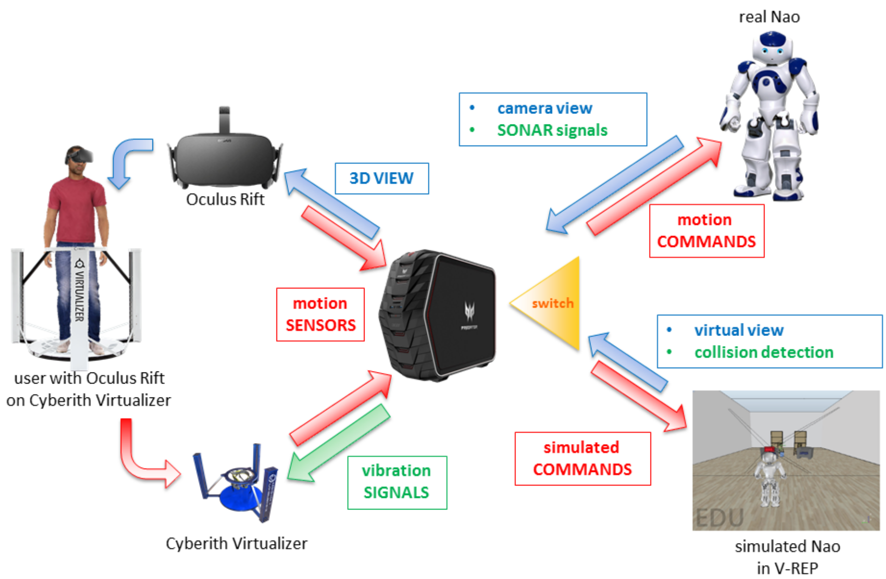


Fig. 3: Overview of the control architecture.

3 Telepresence

The key to telepresence is being able to send human motion commands to the robot in a natural way and, at the same time and with little or no delay, to visualize what the robot onboard camera is currently framing, with full transparency for the user of this bidirectional communication.

As for the motion commands issued by the human, the problem can be decomposed in two subtasks, namely controlling the robot body and the robot head. To this end, when considering the local mobility of a human or of a humanoid robot, we can reduce the control problem as if both were modeled by a simple unicycle. As a result, the motion of the body for each agent can be characterized by two scalar components, a linear velocity $v \in \mathbb{R}$ and an angular velocity $\omega \in \mathbb{R}$. On the other hand, the head pose can be compactly described by a triple (α, β, γ) of roll, pitch, and yaw angles for each agent.

It is worth to remark that the velocity commands extracted from the human locomotion on the Cyberith platform cannot be directly sent to the Nao robot, because a rescaling is needed in general. This can be done either by adjusting the dimensions of the virtual world on the fly, or by introducing a scaling factor $k_{scale} > 0$ for the transformation of commands to the robot. This scaling factor can be computed, e.g., as a simple proportion between the human and the robot maximum feasible velocities.

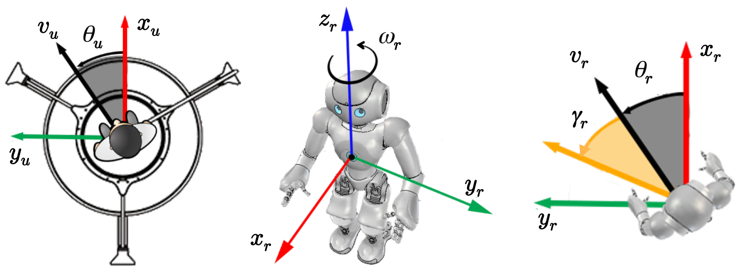


Fig. 4: Definition of axes and variables for the human and the Nao humanoid.

3.1 Body motion

With reference to Fig. 4, let the human orientation around a vertical axis be θ_u, and let v_u and ω_u be the linear and angular velocities of the human. Similarly, for the robot we have θ_r, as well as v_r and ω_r. We define the following mapping

$$v_r = k_{scale}\, v_u, \qquad k_{scale} > 0, \tag{1}$$

as the commanded linear velocity to the humanoid robot. For the angular velocity, we consider the proportional error feedback law

$$\omega_r = k_{body}\,(\theta_u - \theta_r)\,, \qquad k_{gain} > 0. \tag{2}$$

The above gain values cannot be assigned a priori, and their optimal value will depend on the specific user-robot pair. In particular, k_{scale} will take into account the maximum feasible velocity of the robot and the relative size of the virtual environment with respect to the simulated robot. In rough terms, it represents how much the robot size should be enhanced in order to approach the one of the human user. On the other hand, we have limited the angular gain to $k_{body} = 0.25$, in order to prevent overshooting during transients due to the execution delays.

Moreover, larger angular gains would easily drive the robot command ω_r into saturation, since the angular rotation of the user is significantly faster than that achievable by the robot.

3.2 Head motion

Another important aspect that affects the sensation of immersion of the user is how the human head movements are mapped to those of the robot. Indeed, the human is always able to reorient his head in any direction, while moving the body. Let γ_u be the (absolute) yaw angle of the user and γ_r the yaw angle of the robot (relative to its sagittal plane). We have considered again a proportional control law, in order to correct at each instant any relative orientation error of the robot head:

$$\omega_{head} = k_{head} \left((\gamma_u - \theta_u) - \gamma_r \right), \qquad k_{head} > 0. \tag{3}$$

When considering a discrete-time implementation of (3), we obtain

$$\gamma_{r,k+1} = \gamma_{r,k} + \omega_{head,k} \, T_c = \gamma_{r,k} + k_{head} \, T_c \left((\gamma_{u,k} - \theta_{u,k}) - \gamma_{r,k} \right) = \gamma_{u,k} - \theta_{u,k}, \tag{4}$$

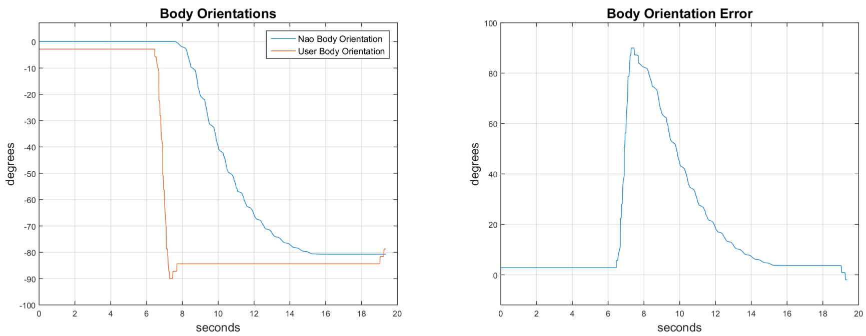
where $T_c > 0$ is the sampling time, and we have chosen $k_{head} = 1/T_c$. In our experiments, the Nao robot controller accepts commands every $T_c = 10$ ms.

4 Experimental Results

Several operative conditions were tested in experiments, with the purpose of validating the strength and reliability of the control and communication code. Some tests were chosen so as to apply the proposed control laws for the body and the head separately. The following results refer to three different situations: a simple rotation, a rotation combined with a walk, and finally a complete navigation task. Each test was performed both in simulation and with experiments on a real robot, sending the output of the camera to the HMD.

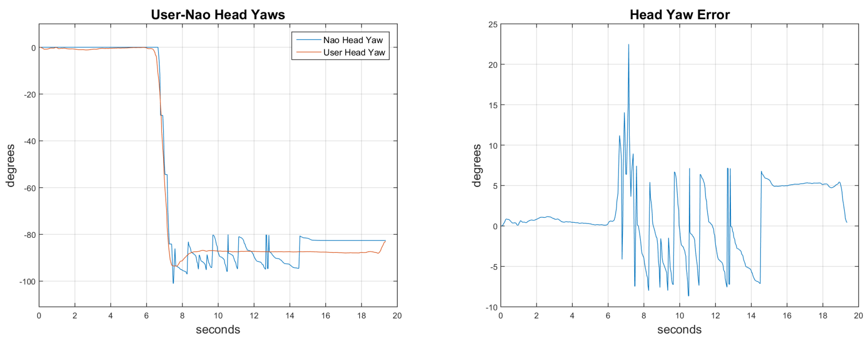
4.1 Rotation

The first test is a simple rotation of the user body. As shown in Fig. 5, after a rotation of the user by 90°, the orientation error has a peak and then is exponentially corrected to zero. The robot orientation shows an initial finite delay of slightly less than 2 s and a settling time of almost 7 s. Because of this slow response, which is due to an intrinsic limitation of the mechanical locomotion of the humanoid robot, the user is supposed to wait until the robot completes its rotation. Thus, she/he needs to become confident with this temporization, adapting the own commands to the robot responsiveness. In Fig. 6a, the yaw angle of the robot head shows significant fluctuations due to the motion, whereas the signal coming from the HMD is smooth. This is translated into the noisy error signal of the relative human-robot yaw error shown in Fig. 6b.



(a) User-Nao Body Orientations (b) Body Orientation Error

Fig. 5: Body orientations and error during a simple rotation with simulated Nao.



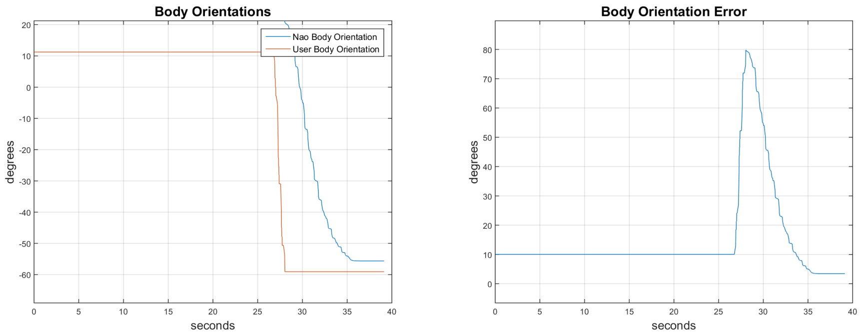
(a) User-Nao Head Yaws (b) Head Yaw Error

Fig. 6: Head yaws and error during a simple rotation with simulated Nao.

In the experiment with the real robot we can observe a similar situation, see Figs. 7–8. The results from the teleoperation of the real robot are indeed perturbed by noise and affected by different delays. Although it is difficult to provide an accurate numerical estimation of the several non-idealities that influence the complete control and communication framework, these appear in general as an overall finite delay roughly in the order of 2 to 3 s at the body motion level, and of less than 0.5 s at the head motion level.

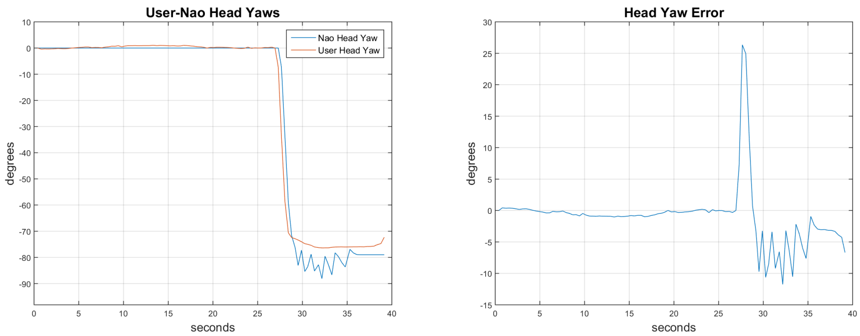
4.2 Rotation and walk

The second test consists again in a rotation, combined with a walk: the user rotates the body while sliding feet on the platform. As before, the behavior of the body error in Fig. 9 is regular, with slight oscillations due to the movements of



(a) User-Nao Body Orientations (b) Body Orientation Error

Fig. 7: Body orientations and error during a simple rotation with real Nao.



(a) User-Nao Head Yaws (b) Head Yaw Error

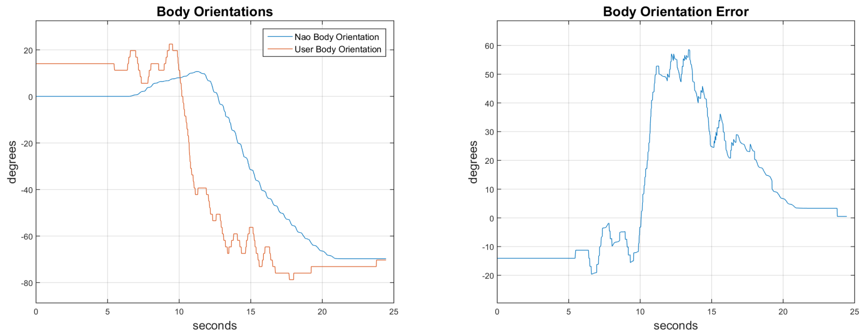
Fig. 8: Head yaws and error during a simple rotation with real Nao.

the hips during the walk. Similar observations regarding the head can be drawn from Fig. 10, as in the previous case of a simple rotation. The experimental results on body and yaw orientations and on their errors with the real Nao are shown in Figs. 11–12.

4.3 Navigation

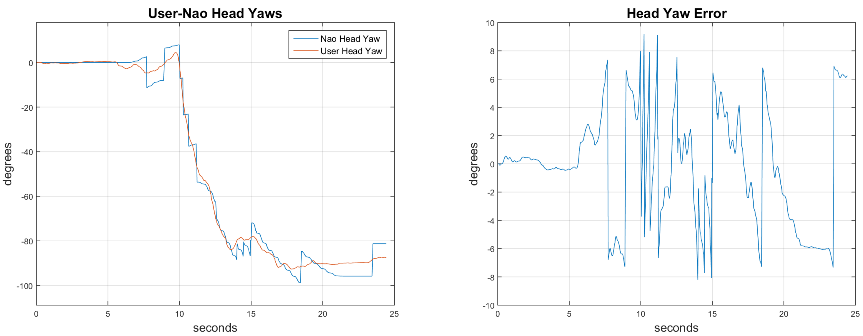
The last experiment is a complete navigation, which can be better appreciated in the accompanying video available as `https://youtu.be/RqMfbvWymFo` on the YouTube channel of our laboratory.

For the simulated robot, we built a virtual room in V-REP emulating a living room. Since the robot odometry is estimated by the simulator, we observed better results as compared to those obtained with a real Nao. Also, the frame rate of the



(a) User-Nao Body Orientations (b) Body Orientation Error

Fig. 9: Body orientations and error during a walk + rotation with simulated Nao.

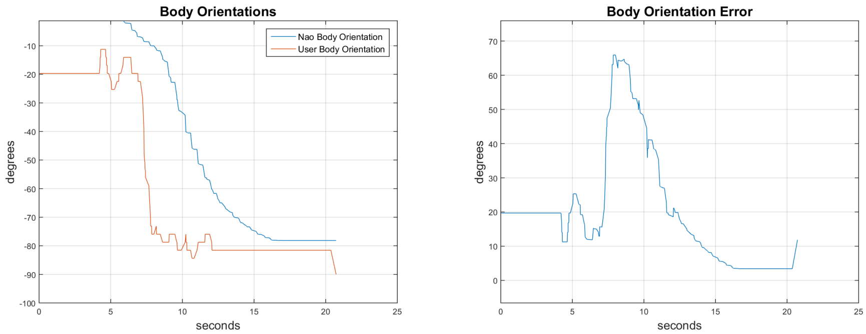


(a) User-Nao Head Yaws (b) Head Yaw Error

Fig. 10: Head yaws and error during a walk + rotation with simulated Nao.

virtual camera is obviously faster and its resolution higher; thus, as expected. we observed systematically a better performance in simulated environments. Figure 13 shows some snapshots from the experiment with the simulated and with the real robot.

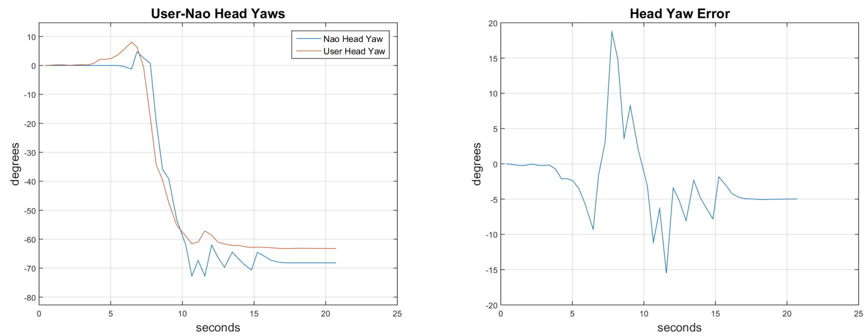
The results of the experiments are perturbed by several non-idealities, some independent from the robot, such as noise and delay in the communication channel, other related to the Nao itself. For instance, it is well known that this humanoid robot is affected by a significant drift in the odometry, as discussed in [18]. As time goes by, the localization error would become significant, impairing the proper execution of the programmed task. On the other hand, the success of a remotely driven real navigation will depend not only on the reliability of the control laws, but also on the quality of the visual feedback provided to the user. Thus, during longer motion tests, there will be a higher cognitive burden placed



(a) User-Nao Body Orientations

(b) Body Orientation Error

Fig. 11: Body orientations and error during a walk + rotation with real Nao.



(a) User-Nao Head Yaws

(b) Head Yaw Error

Fig. 12: Head yaws and error during a walk + rotation with real Nao.

on the user in order to correct the position of the Nao and follow a trajectory to the desired goal.

5 Conclusions

We have presented a combined application of remote locomotion and human telepresence using a humanoid robot, both in a simulated virtual reality environment and in the real world. To the authors' knowledge, this is an original result in terms of realizing a complete loop between a low-cost locomotion interface, a remote humanoid in navigation, and a visual feedback to the walking user from real or virtual cameras.

The obtained results on user immersion are preliminary but encouraging. One major bottleneck was the relatively large delays present in the control loop.

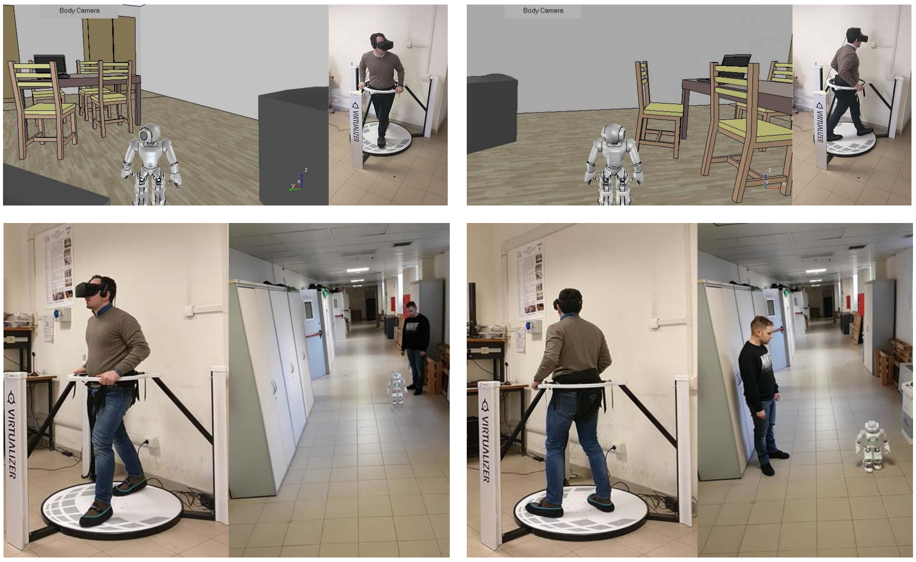


Fig. 13: Navigation in virtual [top] and real [bottom] environments.

Indeed, the Nao robot is not equipped with a powerful processor, and due to the heavy data that the real Nao has to manage through its communication channel, the reactivity of robot body control suffers, and the frame rate of the on-board camera providing visual feedback to the user decreases significantly.

In the next future, the presented work has a large range of possible developments. First, improve the on-board vision module of Nao, by mounting on the robot a stereo camera and bring stereoscopy in the picture, as done in the simulated world. The main drawback for this robot is that the USB port on its head does not allow a good and efficient frame sending. A possible solution would be to direct the streaming onto an external router, so that the internal robot communication will only be responsible for exchanging motion data.

Second, improve the velocity mapping between the human commands and the robot by introducing a user-dependent rather than a constant scaling factor. A possible way is to learn the most appropriate value from the physiognomy of each person.

Third, remote user immersion may involve more haptics, introducing also grasping of objects. This may be implemented using a hand-tracking algorithm when the robot is not moving, e.g., through a motion sensing camera or the Oculus Touch. The first device could also be used for tracking feet, in order to climb stairs remotely.

References

1. Niemeyer, G., Preusche, C., Stramigioli, S., Lee, D.: Telerobotics. In B. Siciliano, O. Khatib (Eds.), *Springer Handbook of Robotics*, Springer, 1085–1108 (2016)
2. Scheggi, S., Meli, L., Pacchierotti, C., Prattichizzo, D.: Touch the virtual reality: Using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering. *Proc. ACM SIGGRAPH* (2015)
3. Souman, J., Robuffo Giordano, P., *et al.*: CyberWalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Trans. on Applied Perception*, 8(4), 24:1-24:22 (2011)
4. LaValle, S.: Virtual Reality, `http://vr.cs.uiuc.edu`
5. Maisto, M., Pacchierotti, C., Chinello, F., Salvietti, G., De Luca, A., Prattichizzo, D.: Evaluation of wearable haptic systems for the fingers in Augmented Reality applications. *IEEE Trans. on Haptics* (2017)
6. Haddadin, S., Croft, E.: Physical human-robot interaction. In B. Siciliano, O. Khatib (Eds.), *Springer Handbook of Robotics*, Springer, 1835–1874 (2016)
7. Biocca, F., Delaney, B.: Immersive virtual reality technology. In F. Biocca, M. Levy (Eds.), *Communication in the Age of Virtual Reality*, Hillsdale, NJ: Lawrence Erlbaum, 15–32 (1995)
8. Rendering to the Oculus Rift, `https://goo.gl/WFAu1D`
9. Rohmer, E., Singh, S. PN, Freese, M.: V-REP: A versatile and scalable robot simulation framework. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1321–1326 (2013)
10. Suleiman, W., Yoshida, E., Kanehiro, F., Laumond, J.P., Monin, A.: On human motion imitation by humanoid robot. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2697–2704 (2008)
11. Do, M., Azad, P., Asfour, T., Dillmann, R.: Imitation of human motion on a humanoid robot using non-linear optimization. *Proc. 8th IEEE Int. Conf. on Humanoid Robots*, 545–552 (2008)
12. Kim, T. and Kim, E., Kim, J.W.: Development of a humanoid walking command system using a wireless haptic controller. *Proc. Int. Conf. on Control, Automation and Systems*, 1178–1183 (2008)
13. Naksuk, N., Lee, CS G., Rietdyk, S.: Whole-body human-to-humanoid motion transfer. *Proc. 5th IEEE Int. Conf. on Humanoid Robots*, 104–109 (2005)
14. Koenemann, J., Burget, F., Bennewitz, M.: Real-time imitation of human whole-body motions by humanoids. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2806–2812 (2014)
15. Dariush, B., Gienger, M., *et al.*: Online transfer of human motion to humanoids. *Int. J. of Humanoid Robotics*, 6(2), 265–289 (2009)
16. Lemoine, P., Thalmann, D., Gutiérrez, M., Vexo, F.: The “Caddie Paradigm”: A free-locomotion interface for teleoperation. *Work. on Modelling and Motion Capture Techniques for Virtual Environments (CAPTECH)*, 20–25 (2004)
17. Cakmak, T., Hager, H.: Cyberith Virtualizer - A locomotion device for virtual reality. *ACM SIGGRAPH* (2014)
18. Ferro, M., Paolillo, A., Cherubini, A., Vendittelli, M.: Omnidirectional humanoid navigation in cluttered environments based on optical flow information. *Proc. 16th IEEE Int. Conf. on Humanoid Robots*, 75–80 (2016)
19. Project Avatar: A Gesture-Controlled Fully Immersive Telepresence Robotics System with NAO*, `https://goo.gl/oIQT8t`
20. V-REP remote API, `https://goo.gl/VZ3b0L`