On the development of a robotic system for telepresence

Paulo A. F. Rezeck and Marcos A. M. Vieira and Luiz Chaimowicz and Mario F. M. Campos
Computer Science Department,
Universidade Federal de Minas Gerais, Brazil
Email: {rezeck,mmvieira,chaimo,mario}@dcc.ufmg.br

***Abstract*—In a large number of tasks, such as exploration, inspection and surgery, robots are being used to replace or represent humans in remote environments. To investigate the challenges of this emerging research field, we have developed a robotic system for telepresence. It is composed of two main modules: a robotic avatar, which is able to represent a human in a remote environment and a user interface that allows a remote operator to task and control the avatar. Through the use of cameras, microphones and other sensors on the avatar, the operator can assess the environment and interact with other humans in the remote area. The robot can execute commands given by the operator and also act autonomously in certain tasks such as navigation and obstacle avoidance. In this paper, we describe the main components of the system in terms of hardware and software and present some proof-of-concept experiments to demonstrate its capabilities.**

I. Introduction

There are several tasks that require a human operator to be virtually present in a remote environment. These tasks range from maintenance operation in nuclear power plants to tele-surgery and tele-medicine. In general, these tasks are performed by a semi-autonomous robot remotely controlled by a human operator. Depending on the task, the robot needs to emulate the human, presenting audio and video streams, performing actions in the remote site and giving sensory feedback to the operator. In this case, they are generally called robotic avatars.

In the last few years, we have been developing a cyber-physical system for semi-autonomous telepresence applications. This system is composed of two main modules. The first consists of a client operated locally by the user, which transmits audio and video captured from the user's computer to the robotic avatar. The second is the robotic avatar, a mobile indoor robot that provides the user the opportunity to participate and interact in a remote environment. The robotic avatar, called DOKBot, can be remotely controlled by a user and sends and receives video and audio to provide the user the perception of being immersed in the environment where the robot is. The robot is semi-autonomous, being able to navigate to waypoints and avoid obstacles in its trajectory. This allows greater user interaction with the physical location where the robotic avatar is located. A picture of the avatar is shown in Figure **??**.

It is important to mention that there has been a great development in the use of robots for telepresence in recent years. One example is *InTouch health* [**?**], a commercial system for telemedicine with remote presence devices. In general, these semi-autonomous systems are important tools to reduce the cognitive load of the user [**?**] and facilitate navigation in environments with dense distribution of obstacles [**?**]. Following this objective, *Willow Garage* made *Texai* [**?**], a tele-immersive system that implements semi-autonomous navigation through laser sensors. Another example is the company *Anybots*, which sells since 2010 the QB robot [**?**]. It is able to move in the form of an inverted pendulum and has a laser that allows the user to pinpoint objects in the scene. Other projects in this area perform tele-presence without sending the user video, such as the mobile webcam *Rovio* and the avatar *TEROOS* [**?**].

In this paper, we describe the main components of our system in terms of hardware and software and present some proof-of-concept experiments to demonstrate its capabilities. Our main contribution is to demonstrate the feasibility of developing such a complex system integrating several off-the-shelf components and using an open software platform, which goes in a different direction from the aforementioned related works.

Fig. 1. DOKBot, a semi-autonomous avatar for telepresence applications.

This paper is organized as follows: the next section presents an overview of the system. Then, sections **??** and **??** detail the hardware and software systems used in the user side and in the avatar respectively. In Section **??** we present results from some experiments performed in a real scenario. Finally, Section **??** brings the conclusion and directions for future work.

II. System Overview

Our system is divided into two parts:

1) User side: controlled locally by a user. It transmits audio and video captured on her computer to the robot. It also sends commands for driving the robot and servo motors that move the cameras.
2) Robot side: is the avatar that sends video and audio from both Avatar cameras, shows video and audio from the user side and performs the semi-autonomous navigation.

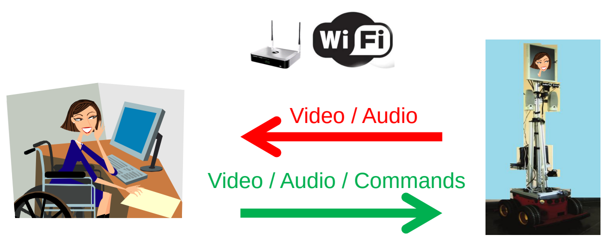

Fig. 2. System Overview.

Figure **??** depicts the system overview. We have a user side and a robot side. The user tele-operates the robot using a computer. The user sends audio, video and commands to the robot. The robot sends audio and video to the user. The interaction between the user and its avatar is made through a wireless network.

We implement the software components using the Robot Operating System (ROS) [**?**]. ROS enables the client (user) to interact with the server (avatar). The communication system of *ROS* is composed of an execution process graph, where each node is connected by a point-to-point network. Nodes are combined together into a graph and communicate with one another using streaming topics. Topics are named buses over which nodes exchange messages. They have anonymous publish/subscribe semantics, which decouple the production of information from its consumption. Topics are unidirectional, streaming communication.

ROS provides libraries and tools to help software developers creating robot applications. In ROS framework, a program that communicates with another program is called a node. Thus, a node is a process that performs computation. *roscore* is a kernel that allows ROS nodes to communicate. *roslaunch* is a tool for easily launching multiple ROS nodes locally and remotely via SSH. It includes options to automatically respawn processes that have already died. *roslaunch* initializes ROS nodes through an XML configuration files that specify the parameters to set and nodes to launch.

We use the transport message, encapsulated by TCP/IP software stack, which is the ROS standard.

The user side software and robot communicate between themselves using an 802.11 wireless enterprise network composed of many access points.

III. User side

In this section, we describe the hardware and software components of the user side in our system.

A. Hardware Components

At the user side, we have several components: video camera, microphone, speaker, monitor, mouse, and joystick attached to a desktop computer. User utilizes an WingMan Attack 2 joystick to drive the robot and control the robotic head.

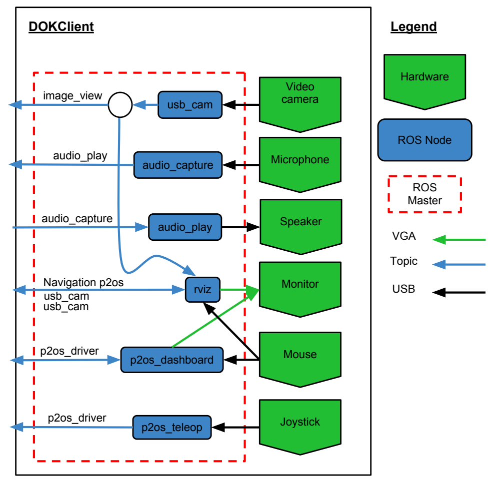

Fig. 3. Diagram of hardware components and ROS nodes running at the client.

B. Software Components

This module is executed at the user computer. It is composed of an audio and a video stream ROS node. There is a node to control the graphical user interface. To transmit the video, it uses the theora format and the resolution is adjustable by the user accordingly with the wireless data transmission capacity. The audio system utilizes the Gstreamer library to transmit with 128 bitrate. The node that controls the robot movement uses the pioneer library. We modified this node to also be able to control the cameras pan-tilt motors. This node works with the joystick device, receiving the Cartesian position from the joystick and sending it to the robot. The node that provides the graphical user interface is called RVIZ. It is a ROS standard tool used as graphical interface. Figure **??** depicts a diagram with the hardware components and software nodes that are executed at the user side.

Figure **??** shows the RVIZ screen which is executed at the DOKclient. These nodes interact with the user so that the user can view all videos that are present in the system. The user can control the avatar movements through this system.

The RVIZ system consists of a map provided in advance by the robot. In the Map, user views landmarks, possible robot's obstacles and robot location. The user can send the avatar to anywhere in the map with just one click.

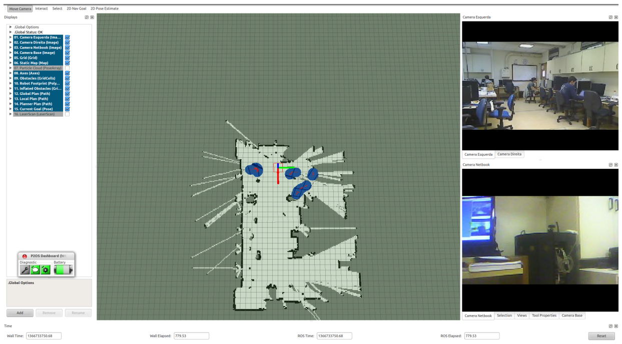

Fig. 4. Rviz containing the environment map and the video streams from avatar and user cameras.

IV. Avatar

In this section, we describe the hardware and software components of the robot side in our system.

A. Hardware Components

The DOKserver is composed of a robotic base and some peripherals. The avatar base is a robot pioneer P3AT robot. We choose this robot because of its robustness and load autonomy. We added a system called robotic head. It consists of a pair of Logitech Webcam Pro 9000 cameras with a microphone and a pair of Dynamixel AX-12 servo motors. The movements of this robotic head is controlled by servo motors through of the joystick. We integrated two speakers to transmit the user audio. We incorporated a 15 monitor SyncMaster 510N model for transmission of the user's face. There is a Hokuyo URG-04lx-UG01 laser sensor for the localization and navigation systems and an Asus EEEPC netbook for processing.

B. Software Components

This package consists of nodes that represent the pioneer driver, the navigation system, the robotics head driver, and audio and video components. The pioneer node serves to control the robot motors and provides information about odometry, battery and touch sensors. The video camera node, similar to the user side node, sends the images from the two Avatar camera to the user, who is able to see them through the node RVIZ. For the special case of visualizing the user image into the robot, we used another node called image view, since this is a more objective approach. The sound system is the same as the one described in the user side. We develop a ROS node to control the robotic head node. We describe the navigation system in Section **??**. Figure **??** depicts a diagram with the hardware components and software ROS nodes that are executed by the avatar.

1) Semi-Autonomous Navigation: Here, we describe the navigation system. The user tele-operates the robotic avatar indicating where it should go. The robotic avatar interacts with humans and can be used in environments that have obstacles. The navigation system uses a set of nodes that takes into consideration odometry and laser sensor readings. Moreover,

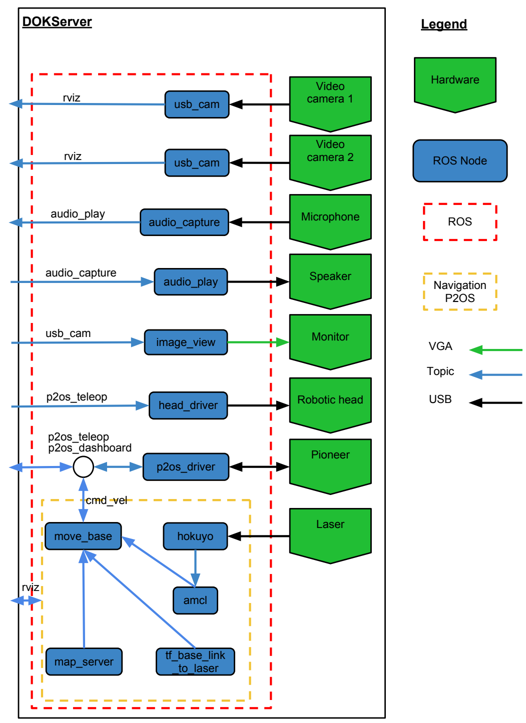

Fig. 5. Diagram of hardware components and ROS nodes running at the server (Avatar).

it has a package that lets the user keep track of multiple coordinate frames over time (called *tf*).

The tele-immersive system must be safe. The navigation system includes a collision avoidance algorithm called Dynamic Window Approach (DWA) [**?**]. It performs forward simulations from the robot's current state to predict what would happen if the sampled velocity were applied for some (short) period of time, picks a safe trajectory, and sends the associated velocity to the mobile base.

Figure **??** shows the navigation components. It illustrates the ROS nodes and their communication through topics. The navigation system uses the following ROS nodes:

- *Hokuyo*: provides laser sensor readings. For each angle, the laser scans it and returns the distance to the closest object that reflected the laser.
- *p2os_driver*: provides a driver to control the pioneer robot.
- *base_link_to_laser*: maps the laser readings to the mobile base coordinate frame.
- *move_base*: allows to, given a goal in the world, attempt to reach it with a mobile base. The *move_base* utilizes a global planner to accomplish its global navigation task. It also includes the Dynamic Window Approach collision avoidance algorithm.
- *AMCL*: is a probabilistic localization system for a robot moving in 2D. It implements the adaptive Monte Carlo localization approach which uses a particle filter to track the pose of a robot against a known map.

- *map_server*: offers a map data as a ROS Service. It also allows dynamically generated maps to be saved to file.

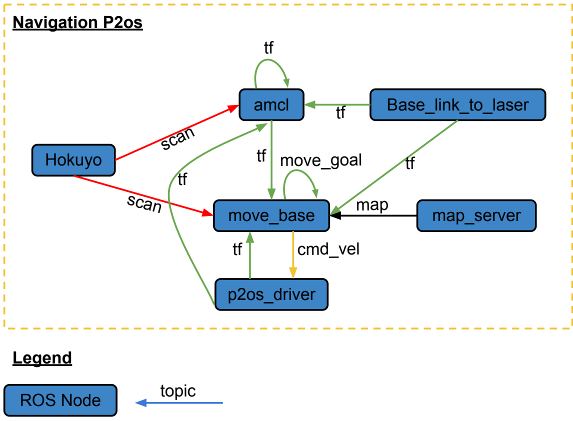

Fig. 6. Diagram of a Semi-Autonomous Navigation task.

V. Evaluation

In this section, we describe a proof-of-concept experiment of the avatar in a real scenario using a wireless enterprise network and demonstrate its capabilities. The mission consisted in navigating in some corridors transmitting audio and video streams in both directions. We evaluate both sides of the system during the experiment. The avatar transmits audio and video to the user interface. The user interface transmits audio, video and commands to the robot.

Figure **??** shows the mission map. The white background indicates open space where it is possible to navigate. The gray background indicates area which navigation is not possible due to walls. Each wifi access point is represented by a wifi symbol in the figure. The user location is indicated by a blue dot. The initial location of the avatar is represented by a red dot. The mission's objective is to make the robot reach the green dot. The green line represents the estimated avatar trajectory during the proof-of-concept experiment. During the experiment, we collected pictures of the user interface and avatar. We labeled on the map six numbered balls indicating places where we took images from the experiment. The balls are ordered by the mission execution time.

Initially, Avatar and user communicates between themselves using the wireless access points. The user marks on the map the initial estimated position of the robot to assist the navigation system and mitigate any possible localization error.

User starts the experiment by setting the avatar's goal position with its interface. The avatar navigation system guarantees that the avatar moves to the goal.

Figure **??** shows the beginning of the mission. The avatar is at position 1 on the map (Figure **??**). Figure **??** also shows the user interface (using RVIZ) and the robot at the initial position. On the left side of the user interface we can observe its settings. The map is shown at the center of the interface. The user can observe obstacles (blue points), laser readings (red points), and the current position of the robot. We can also send the avatar to a particular location by just clicking on the map. On the right side of the user interface are the video streams. The upper video screen shows avatar's views and the lower video screen shows the user's local view.

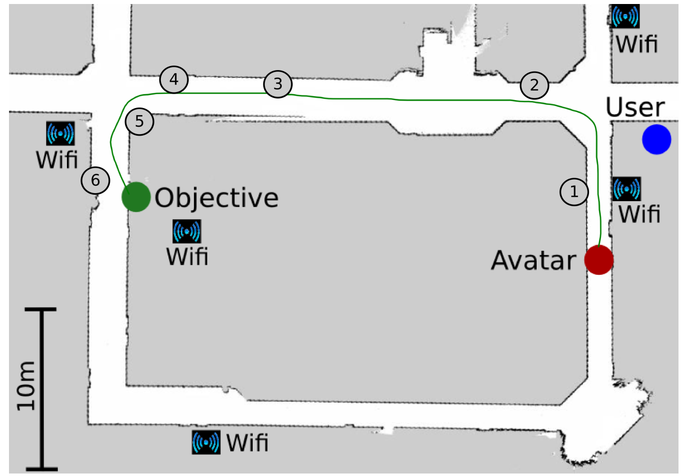

Fig. 7. Mission Map.

Figure **??** depicts the avatar entering the horizontal corridor of the map (position 2). Figure **??** presents the user interface when the Avatar was moving at the big corridor (position 3). The user interface illustrates the position of the robot. The green line indicates the computed path the robot still needs to go until it reaches the final position (purple vector) marked by the user. In the upper video on the right screen, we can see the avatar going to the location represented by Figure **??** (position 4 on the map at Figure **??**).

Figure **??** again depicts the user interface. The avatar is at position 5 on the map. The green line indicates the trajectory to reach the goal. Figure **??** shows the avatar reaching the final position of the mission (position 6).

The avatar's behavior throughout the mission was stable and secure. Even with the transition between wifi access points, the system worked continuously. A video of this experiment in accompanying the paper and can also be seen at: http://www.youtube.com/watch?v=3Y-oFBkJpG8.

VI. Conclusion

In this paper we presented the development of a robotic avatar for telepresence applications. We described the hardware/software architecture of the user side and the avatar (robot) and the communication framework used to connect them. We evaluated the system in a real scenario using a wireless enterprise network, demonstrating its capabilities. We show it is possible to develop a complex system integrating using several off-the-shelf components and also including an open software platform. As a future work, we intend to perform more tests stressing the semi-supervised control and investigate new protocols for guaranteeing quality of service for the audio and video transmission.

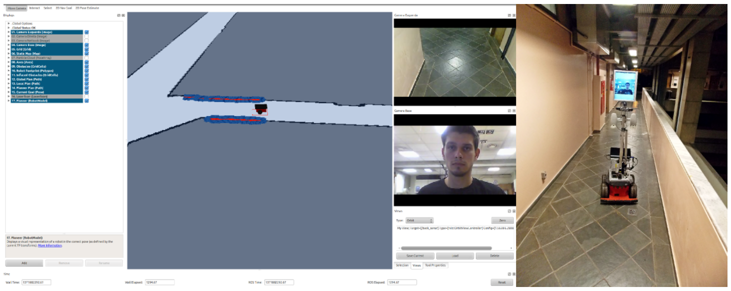

Fig. 8. User interface and Avatar at the beginning of the mission.

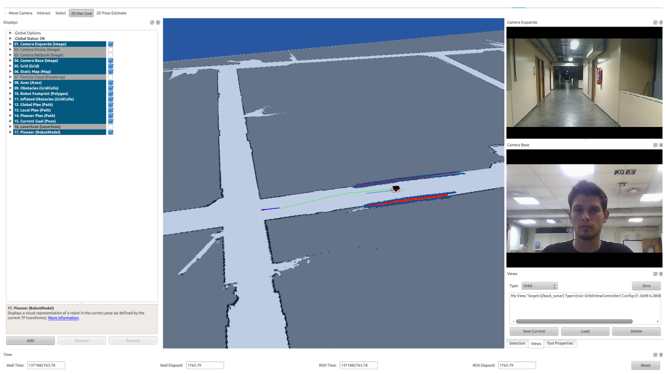

Fig. 10. User interface showing the Avatar at the big corridor.

Fig. 12. User interface showing Avatar ending the mission.

Fig. 9. Avatar entering the big corridor.

Fig. 11. Avatar moving to the end of the corridor.

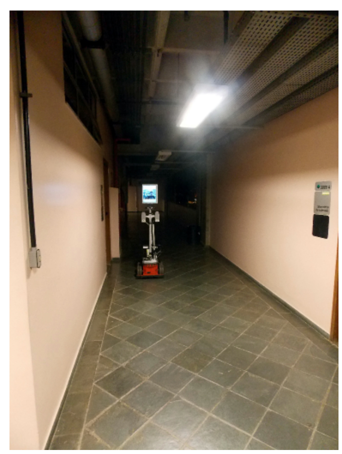

Fig. 13. Avatar reaching the final objective position.

ACKNOWLEDGMENTS

This work is partially supported by CNPq and Fapemig.