Human-Centric Assistive Remote Control for Co-located Mobile Robots

Akansel Cosgun
Georgia Tech
Institute for Robotics and Intelligent Machines

Arnold Maliki
Georgia Tech
Mechanical Engineering

Kaya Demir
Georgia Tech
Electrical and Computer Engineering

Henrik Christensen
Georgia Tech
Institute for Robotics and Intelligent Machines

ABSTRACT

Autonomous navigation is an essential capability for domestic service robots, however at times direct remote control may be desired for cases where robot and user are co-located. In this work, we propose a remote control method that allows a user to control the robot with smartphone gestures. The robot moves with respect to the user's coordinate frame and avoids obstacles if a collision is imminent. We think that interpreting the commands from human's perspective would decrease the cognitive load of the user, therefore allowing efficient operation.

Keywords

Assisted teleoperation; Interfaces; Domestic robots

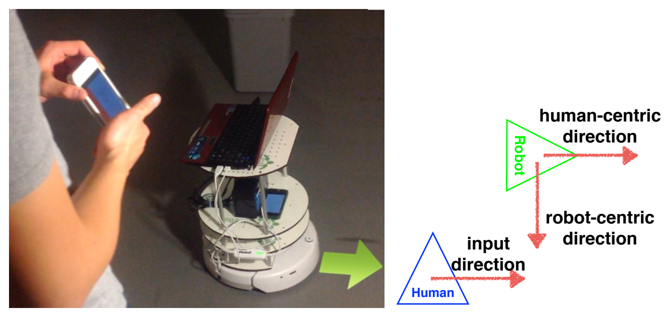
Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

1. INTRODUCTION

In the future, domestic service robots are expected to become a common consumer product for homes. In most applications, the robot is required to navigate among clutter the the home environment. Autonomous navigation for mobile robots is an essential capability, however at times remote control by the user may be desired. For example, if a user wants to position the robot with precision, instead of "go to kitchen", she may want to drive the robot herself. Similarly, a user may want to park the robot to a corner when it is unused or get the robot out of the doorway that the robot is blocking. Positioning can be done in a number of ways, such as with speech or by pointing to the desired location, however we argue that driving the robot directly will give uses more control. The remote control user interface should be intuitive and it should use human feedback for meaningful motion planning. Assistive remote control interfaces has been proposed recently and is shown to reduce the number of collisions [1, 2], however the focus has usually been telepresence robots. Instead, we are focusing on scenarios where the user and the robot are in the same location.

HRI'15 Extended Abstracts, Mar 02-05 2015, Portland, OR, USA
ACM 978-1-4503-3318-4/15/03 .
http://dx.doi.org/10.1145/2701973.2702066



(a) (b)

Figure 1: Remote control of a robot with smartphone gestures is illustrated. The user provides a direction of "right" by tilting the device and the robot moves in the "right" direction with respect to the user's coordinate frame.

In this work, we propose a remote control method that assists the user by interpreting commands from his/her point of view and avoiding obstacles. A laser scanner is used to detect and avoid obstacles. Compass data from the robot and user's phones are used for human-centric control. Our contributions are three-fold:

1. A UI that allows controlling a robot by tilting the device
2. An assistive layer that helps avoid obstacles
3. Intuitive control by moving the robot with respect to user

2. APPROACH

Our assistive remote control approach requires three main components: a user interface, a communication handler and a trajectory planner. The system diagram shown in Figure 2 demonstrates the relationship and data flow between these components.

First, the user provides an input direction using an iPhone app interface. Compass data from both iPhones are fetched to find the direction robot needs to move for human-centric motion. We use UDP protocol over WiFi for communication with the robot's computer. Then, trajectory planner gets the laser data and finds an obstacle-free trajectory close to the input direction. Finally, linear

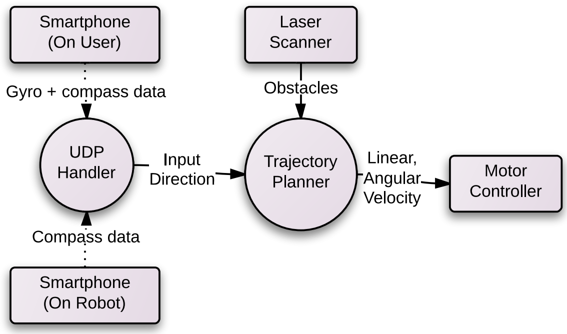


Figure 2: System Diagram.

and angular velocities are applied to the robot by its motor controllers.

2.1 System Overview

Our system consists of a Turtlebot robot, a Hokuyo UTM30LX laser scanner and two iPhone 4s devices. The Turtlebot base is a iRobot Create vacuum cleaner robot. One iPhone is used by the user for gyroscope and compass data and the other is placed on the robot for compass data. We use Robot Operating System (ROS) on the robot computer for message passing between processes.

2.2 Smartphone App

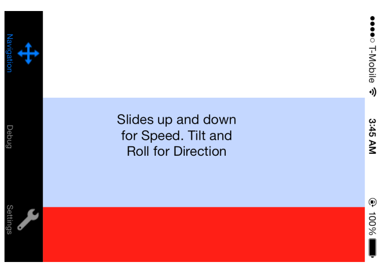


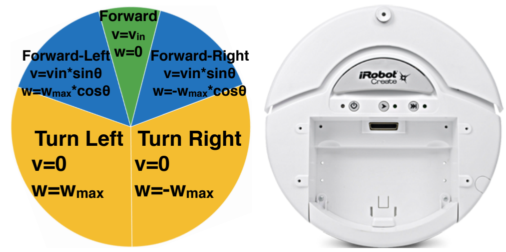
Figure 3: iPhone app is used in landscape mode as shown. Direction is controlled by roll/pitch and speed is set by a slider (upper blue line). Speed is zero in red region and max at the top of the screen.

We designed an application that lets a user set the speed of the robot and give an input direction using the roll/pitch of the device (Figure 3). Roll and pitch readings are acquired at 60Hz by the internal gyroscope of the iPhone device. Given the orientation of the device, we calculate a vector representing the input direction. The phase of the input direction vector $A\angle\theta$ is found by: $\theta = arctan(\beta/\alpha)$, where α is roll and β is yaw of the smartphone. The magnitude is found by: $A = |\alpha| + |\beta|$. The robot is stopped if the magnitude is below some threshold $A < A_{min}$. The compass readings are also transmitted to the robot for human-centric control. We use UDP protocol over TCP because overhead is less and transmission is faster.

2.3 Human-Centric Control

With human-centric control, the input direction from user's phone is interpreted in the human's coordinate frame instead of robot's. Figure 1(b) illustrates how the input direction is corrected for human-centric control. True magnetic heading data from the magnetometer of the smartphone from both the user's and robot's phones are used to find the relative angle between the robot's orientation and human's orientation. We update the angle of the input direction vector accordingly to the difference in orientations of the person and robot. We assume that the app is always used in the landscape mode during remote control and user holds the device parallel to her torso.

After the adjusted input robot direction vector is found, we compute a linear and angular velocity pair (v, w) that would be applied to the robot base. If the input direction is in the "forward" direction of the robot, the linear velocity should dominate. If the input is in the "right" direction with respect to the robot, a clockwise rotation should be applied. Direction to velocity mapping, along with the robot footprint, is illustrated in Figure 4.



(a) (b)

Figure 4: a) Linear and angular velocity pairs (v, ω) applied, given the input direction. b) Robot base.

2.4 Obstacle Avoidance

A linear and angular velocity is computed from the user input and relative orientation of the robot and the human; however we should check if the robot will end up colliding an obstacle upon applying velocities to the robot base. Obstacles are detected using a Hokuyo laser scanner. The robot trajectory is simulated for Δt and if the default velocity pair is projected to result in a collision, we search for a similar trajectory that does not result in a collision. In the current implementation, we linearly sample about 20 angular and 4 linear velocities, centered around the original velocities. We use the motion model shown below for simulating trajectories:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{v}{\omega} + \frac{v}{\omega} sin(\pi/2 + \omega\Delta t) \\ -\frac{v}{\omega} cos(\pi/2 + \omega\Delta t) \end{pmatrix}$$

Collision check is conducted by finding the closest laser hit to the test point and testing if the that distance is smaller or greater than the radius of the robot base. If none of the sampled trajectories is collision free, then the robot stops. Otherwise, the velocity pair (v, ω) that is closest to the original is applied to the robot by the motor controller at 10Hz rate.

3. CONCLUSION AND FUTURE WORK

We presented an assistive remote control method using a smartphone, obstacle avoidance and human-centric control. Our hypothesis that human-centric control would take less cognitive load than robot-centric, will be tested by user studies as future work.

References

[1] D. G. Macharet and D. A. Florencio. A collaborative control system for telepresence robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5105–5111. IEEE, 2012.

[2] L. Takayama, E. Marder-Eppstein, H. Harris, and J. M. Beer. Assisted driving of a mobile remote presence system: System design and controlled user evaluation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1883–1889. IEEE, 2011.