2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Gesture-based Robot Control: Design Challenges and Evaluation with Humans

Enrique Coronado, Jessica Villalobos, Barbara Bruno, Fulvio Mastrogiovanni

***Abstract*— In this paper we introduce a gesture-based robot control framework, we discuss the adopted design principles and we report results about its evaluation with humans. Gesture-based control using wearable devices may constitute a novel form of human-robot interaction, but its implications have not been discussed in the literature. We discuss the main challenging issues, possible design guidelines and an open source, freely available implementation using commercially available devices and robots. The overall performance of the architecture, as well as its validation with 27 untrained volunteers, is reported.**

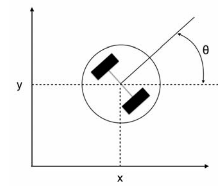

Fig. 1. Left: the robot used in our experiments. Right: the kinematic structure of a unicycle-like robot.

I. Introduction

Human-Robot Interaction (HRI) is a research field dedicated to understanding, designing and evaluating robotic systems when working for or with humans [1]. An essential element of the interaction process is the (possibly multi-modal) form of communication. Each independent channel enabling communication is known as a *modality* [2]. Typical modalities used to *remotely control* robots, such as keyboard, mouse, joystick or teach pendant, are generally considered unnatural methods of communication and require prior training, which can be unpleasant and time consuming [3].

We argue that new robot-mediated services will be enabled by simpler or novel forms of communication, suitable to be used also by non technically skilled people, and requiring a shorter learning curve to gain a full user experience [4]. Such simplification is possible with the adoption of *natural interfaces* [3][5], i.e., interfaces enabling users to interact with computing devices (and robots) in the same way they interact with their everyday environment [6].

In this work, the key hypothesis is that body tracking devices are able to capture a wide range of natural gestures, which can be used to interact with robots. In our case, gestures are used to impose specific functional states (i.e., *idle* or *moving*) and to map user movements to robot control parameters. The study of gesture-based natural interfaces as a viable form of robot control has still open issues. In the literature, this topic has not been explored in detail, and problems such as usability, precision in control, physical fatigue after long periods of operation, as well as the performance of the user interface must be carefully analysed [3].

In an effort towards the development of intuitive and natural communication interfaces, we propose an architecture for the gesture-based control of a mobile robot. The main contributions of this paper are: i) the design and development of an *open source*, freely available control architecture for mobile robots relying on human gestures as communication interface; ii) a proposal for the identification of suitable gestures to associate with motion control commands; iii) the definition of a suitable mapping between each gesture and its corresponding motion command. Gestures are modelled using inertial information originating from a person's wrist. The overall output is the set of driving and angular velocity commands to feed a unicycle- or car-like mobile robot. For the tests, we have employed a Husqvarna Research Platform, see Figure 1(a). The usability of the proposed interface has been tested with 27 volunteers.

This paper is organised as follows. In Section II, we formalise our problem, we discuss related work and we survey general principles for gesture-based robot control. Section III presents the proposed gestures, as well as their rationale. Section IV reports, respectively, the overall performance of the system in terms of response times, as well as a discussion about its usage by human volunteers. Conclusions follow.

II. The Use of Human Gestures for Robot Control

A. Problem Statement

We investigate the problem of developing an approach for the control of a wheeled mobile robot via gestures. The control interface must satisfy two general requirements:

R_1 full control on the robot's *motion*;
R_2 full control on the robot's *functional states*.

Although most of the ideas put forth in this paper can be easily generalised, we restrict our analysis to wheeled mobile robots, and in particular to robots with a unicycle- or car-like kinematic model, Figure 1(b). The model of a unicycle-like robot can be expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta \\ sin\theta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2, \tag{1}$$

All the authors are with the Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa, Via Opera Pia 13, 16145 Genoa, Italy. Corresponding author's email: luis.coronado@uaslp.mx.

978-1-5090-4633-1/17/$31.00 ©2017 IEEE

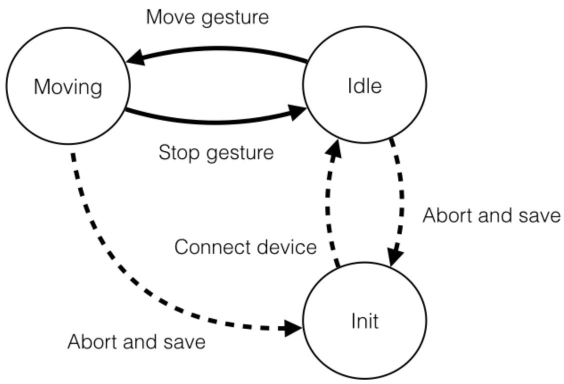

Fig. 2. A finite state machine with functional state transitions.

where x, y are the Cartesian coordinates of the mid location of the axis connecting the two wheels, θ measures the orientation of the robot's body with respect to the x axis, v_1 is the driving velocity, and v_2 is the angular velocity.

The full control of the robot's motion (requirement R_1 above) is enabled by controlling the driving velocity v_1 (control variable C_1) and the angular velocity v_2 (control variable C_2). Meeting R_1 requires us to define a *continuous* mapping between gesture-induced inertial data and the couple v_1 and v_2. For the control of the robot's functional states (requirement R_2 above), we use a simple state machine with three states, namely `init`, `idle` and `moving` (Figure 2). The control of the functional states requires the (discrete) control of the transitions from `idle` to `moving` (control variable C_3) and from `moving` to `idle` (control variable C_4), respectively. After bootstrap, the robot is in the `init` state. The robot has zero speed and no sensing data is acquired. In order to activate navigation (i.e., to switch to the `idle` state), the user must start the communication between the wearable device and the aboard computer. This is done using standard input methods in smartwatches. In the `idle` state, velocities are set to zero and the robot waits until a *move* gesture is performed to switch to the `moving` state. When in the `moving` state, inertial data is continuously acquired and processed to control v_1 and v_2 by means of continuous user motions. In order to switch again to the `idle` state, the robot expects a *stop* gesture. Therefore, requirement R_1 is met by means of continuous gestures, whereas requirement R_2 is met by controlling the transition from `moving` to `idle` and *vice versa*.

B. Related Work

The design of a gesture-based control framework requires to track the movements of a person's arm. Most of the approaches in the literature adopting gestures to interact with robots rely on vision [7]. Unfortunately, their performance is constrained by environmental conditions (including lightning and occlusions) and by the fact that the person must be present in camera view at all times. Wearable sensors are an interesting alternative to vision, since their effectiveness is related only to a minor extent to environmental conditions, and there are no constraints on the mutual spatial relationship between the person herself and the robot. Furthermore, the low requirements in terms of power consumption allow for a prolonged use of the device [8]. Approaches exploring this alternative typically use gloves [9], armbands [10] or other specialised devices [11][12] as input modalities. Although the literature reports solutions based on commercial devices, such as the Leap Motion [13] and the MYO smartband [14], these devices are still far from a widespread use.

Wearable devices with aboard inertial sensors, such as smartwatches, are becoming increasingly popular. According to the International Data Corporation Worldwide Quarterly Wearable Device Tracker, the worldwide wearable devices market is expected to surpass 200 million units shipped in 2019. This organisation also expects a strong smartwatch growth in the next years [15]. Modern smartwatches can easily communicate with their paired smartphone via Bluetooth and have been recently used for human activity recognition [16][17]. On these premises, it makes sense to develop an architecture for the gesture-based control of a robot, which relies on the inertial information provided by a sensor embedded in a commercial smartwatch.

C. Gesture-based Interaction with Robots

In his seminal work on user-centred design, the usability engineer Donald Norman indicates six core principles for a *usable* interface: i) simplifying the structure of tasks; ii) making options visible; iii) getting the mapping right; iv) exploiting the power of constraints; v) designing for error and vi) making affordances explicit [18]. Practical guidelines derived from those principles and allowing for the design of interfaces which maximise usability include:

- G_1 minimising the number of options involved in the interaction, to reduce the user's response times and errors, as suggested also by the Hyck Hyman law [19];
- G_2 avoiding the overload of the user's short-term memory, i.e., limiting the number of significant items (e.g., digits, letters, gestures) that must be remembered [20];
- G_3 providing explicit hints to possible actions, guidelines on how to perform them and direct feedback on user's performance, to enforce learnability [21];
- G_4 referring to actions used in well-known real-world contexts, to reduce the learning curve [22].

Unlike interfaces for human-computer interaction, in which the user pays full attention to the interaction itself, interfaces for the control of mobile machines have to guarantee that the user's main focus is always on the driving task [23]. Research in this field is dominated by the automotive industry: in the case of in-vehicle information and communication systems there exist detailed design guidelines defined by international bodies, such as the European Commission [24] and the Alliance of Automobile Manufacturers [25], which ensure their worldwide adoption. The fundamental guidelines state that:

- G_5 the system should be designed so as to avoid distracting or visually entertaining the driver;

G_6 no part of the system should obstruct the driver's view.

Gesture-based control interfaces making use of inertial sensors, which rely on the user's proprioception for feedback and a single, unobtrusive device located on the user's body, are particularly well suited for the satisfaction of principles G_5 and G_6. Moreover, the use of gestures for the direct control of mobile robots allows for relying on the user's proprioception for the evaluation of the performed action, and the user's vision for the evaluation of the effects of the action on the robot, which are – arguably – the most natural and accurate form of feedback in terms of G_3. Building upon these considerations, our aim is to identify a set of gestures which: i) allow for the control of the control variables C_1, C_2, C_3 and C_4, and ii) maximise the usability of the interface, in accordance with the aforementioned design principles. Although the literature provides examples of gesture-based control interfaces for mobile robots [26], to the best of our knowledge ours is among the few studies explicitly taking into consideration design principles for the selection of the gestures to use.

We propose a one-to-one mapping between gestures and the control variables C_1, C_2, C_3 and C_4. This choice minimises the number of options available for each control variable, in accordance with G_1, without overloading the user's short term memory, in accordance with G_2, thanks to the small number of different gestures to remember. Due to the different nature of the control variables, we propose the use of gestures of two types, namely *continuous* and *discrete*. As we anticipated, continuous gestures are used to control the driving C_1 and angular C_2 velocities of the mobile robot (thus meeting the overall requirement R_1), with a mapping from the continuous multi-dimensional domain associated with the inertial sensor to the continuous variable domain. Discrete gestures are used to switch between functional states, i.e., to control the variables C_3 and C_4 (therefore meeting the overall requirement R_2). In this case, the gesture should define a mapping between the continuous domain of the inertial sensor and the discrete domain of the control variable, which requires specific architectural modules.

The design principle conceptualised in G_4 suggests the use of gestures resembling actions performed in a real-world context. We propose to rely on the real-world context of driving a car, since: i) the shape and kinematic model of wheeled mobile robots typically remind those of a car, thus providing a strong and implicit guideline about robot's behaviour; ii) as stated above, the actions required for driving a car are standardised and universal, which ensures that they are familiar to people regardless of their experience; iii) the control interface of a car, i.e., the set of actions and commands required to drive it, is among the ones with highest usability [18], and we argue that its good properties can hold in our case as well.

III. Gestures Analysis and Selection

A. Driving Gesture, or the Control of the Driving Velocity

In the context of human-machine interfaces for vehicles, interaction tasks are categorised in accordance with their impact on the purpose of driving the car [27]. The classification defines *primary* tasks, which are essential to manoeuvre the car, such as controlling the speed, *secondary* tasks, increasing the safety for the driver, the car and the environment, such as setting turning signals, and *tertiary* tasks, regarding infotainment systems, such as choosing the radio station to listen to. Primary tasks are required to have a one-to-one mapping with the associated functionality and provide immediate feedback. Figure 3(a) shows the gesture we propose, which mimics the action of pushing the acceleration pedal of a car, i.e., the primary modality to control the vehicle's driving velocity. While in car driving the user learns the association between the action and its effects by combining haptic and visual feedback, in our case the association is expressed in terms of proprioception and visual feedback. More specifically, we propose a direct mapping between the angular velocity component ω_x measured about axis x_w of the smartwatch and the driving velocity v_1 of the robot as:

$$v_1 = \begin{cases} 0, & \text{if} \quad \omega_x - \alpha \leq 0, \\ (\omega_x - \alpha)/s_x, & \text{if} \quad \omega_x - \alpha > 0, \end{cases} \tag{2}$$

where α is an offset parameter and s_x is a scaling factor. With respect to Figure 3(a), when the arm is in the red region, i.e., the first condition in (2), the driving velocity is $v_1 = 0$. When the arm is in the blue region, i.e., the second condition in (2), a driving velocity proportional to the value of ω_x is set as a reference. The scaling factor and the offset value, which determine the amplitude of the movements to be executed by the user, can be tuned to decrease the effort and fatigue generated by the use of the interface. Online tuning and adaptation of these parameters is current work.

B. Steering Gesture, or the Control of the Angular Velocity

Figure 3(b) shows the gesture proposed for the control of the robot's angular velocity. Analogously to the rationale adopted for the control of the driving velocity, this gesture mimics the action of moving the steering wheel, i.e., the primary modality to control the steering angle of a vehicle. In both contexts the user learns the association between the action and its effects in terms of proprioception and visual feedback. We propose a direct mapping between the angular velocity component ω_y measured about axis y_w of the smartwatch and the angular velocity v_2 of the robot as:

$$v_2 = \begin{cases} 0 & \text{if} \quad d_1 \geq \omega_y - \beta \geq d_2 \\ (\omega_y - \beta)/s_y & \text{if} \quad d_1 \leq \omega_y - \beta \leq d_2 \end{cases} \tag{3}$$

where β is an offset parameter and s_y is a scaling factor. The red area shown in Figure 3(b), defined by the parameters d_1 and d_2, was experimentally set to ease the control of the robot when moving along a straight line. The scaling factor, the offset and the values of d_1 and d_2, which determine the amplitude of the movements to be done by the user, can be tuned to adapt to the user's preferences.

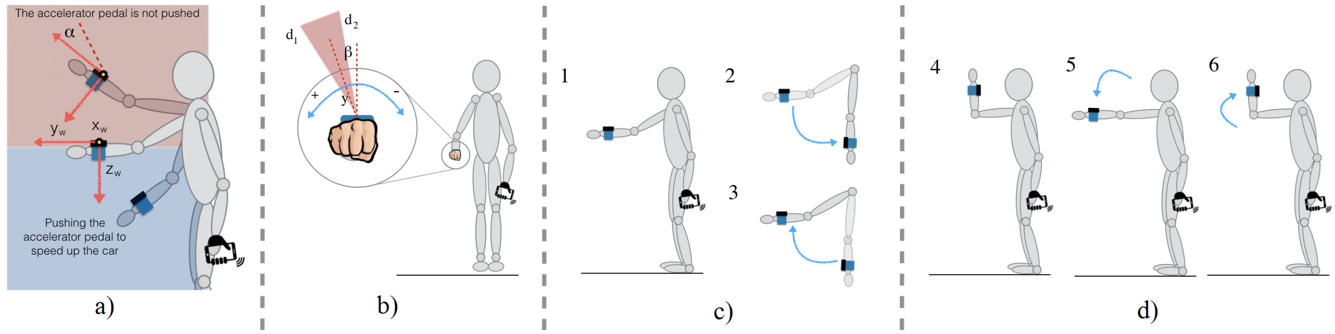

Fig. 3. a) Control of the driving velocity. b) Control of the angular velocity. c) *Move* gesture: 1) start position, 2) first movement, 3) second movement and final position. d) *Stop* gesture: 4) start position, 5) first movement, 6) second movement and final position.

C. Move Gesture, or the Idle-to-Moving Transition

Secondary tasks comprise the ones devoted to minimising the chances of accidentally issuing driving commands, by enabling and disabling them in accordance with the functional state of the vehicle and forcing the driver to perform voluntary actions to move from one state to another. Tasks of this type include turning on/off the engine, typically associated with the action of rotating a key or pressing a button, and releasing the handbrake. Actions associated with secondary tasks are typically more complex than the ones associated with primary tasks and allow for a less immediate feedback. Figure 3(c) shows the discrete gesture proposed to start driving the robot, i.e., for the control of the transition from the `idle` to the `moving` state. This gesture mimics the action of releasing the handbrake, i.e., the secondary command enabling the primary driving commands. In our case, the user learns the association between the action and its effects in terms of proprioception and delayed visual feedback, since a correct execution of the gesture makes the robot move in response to a continuous gesture. This gesture is described in terms of the 3-dimensional acceleration pattern generated by the movement and requires the availability of two modules in the architecture: i) a *recognition* module to analyse the online acceleration data provided by the inertial sensor to check whether they can be classified as an instance of the gesture; ii) a *modelling* module to create a representation of the gesture that maximises the accuracy of the recognition. As soon as the gesture is recognised and properly classified, the system switches the state of the robot to `moving`. We adopt a procedure based on Mahalanobis distance for the recognition step [28], and one relying on Gaussian Mixture Modelling and Gaussian Mixture Regression for the modelling step [29].

D. Stop Gesture, or the Moving-to-Idle Transition

Figure 3(d) shows the gesture used to stop driving the robot, i.e., for the control of the transition from the `moving` to the `idle` state. This gesture mimics the action of pressing the start/stop button of a car, i.e., the secondary command which turns off the engine, and the association between the action and its effects is given in terms of proprioception and visual feedback, i.e., a correct execution of the gesture makes the robot stop. As in the case of the *move* gesture, this gesture is described in terms of the 3-dimensional acceleration pattern generated by the movement. As soon as the gesture is recognised by the recognition module, the system switches the state of the robot to `idle`.

IV. Experimental Evaluation

A. Implementation

The system is currently implemented as a distributed and heterogeneous hardware/software architecture. All the developed software is freely available online[1].

We use an Android Wear compatible Sony Smartwatch 3 paired via Bluetooth with a smartphone Sony Xperia Z1 compact. The smartphone acts as a gateway to a Sony Vaio S14 with an Intel Core i5 processor, 8Gb RAM, connected through WiFi, in which Ubuntu 14.10 and ROS Indigo are installed. The robot is a ROS-enabled Husqvarna Research Platform (HRP), shown in Figure 1(a), which is a car-like robot with rear driving wheels.

Regarding software and modelling aspects, the *recognition* and *modelling* modules are implemented in Python. The modules, as well as the way *discrete* gestures are modelled, represented and recognised at run time, have been described in [28] and [29]. Each gesture is modelled as an expected *average* trajectory (along with its covariance) in acceleration space, starting from a number of human demonstrations. A procedure based on Dynamic Time Warping is used to align and truncate the used demonstrations [30] and a Kalman filter is applied to reduce the effect of noise on trajectories. Specifically, we used inertial data from the smartwatch to model the two discrete gestures *move* and *stop*. For each gesture, we collected a total of 14 human demonstrations from 2 volunteers. Both gestures are composed of 130 acceleration samples, corresponding to a duration of 2.6 *sec*. The *recognition* module applies a sliding window moving on the online data stream provided by the sensor and compares the data within it to each modelled gesture. The output of the module is a measure of *how likely* a given gesture is on the basis of input data [31].

[1]Visit the websites: `https://github.com/EmaroLab/smartwatches_apps`, `https://github.com/EmaroLab/py_socket`, and `https://github.com/EmaroLab/HMPy`.

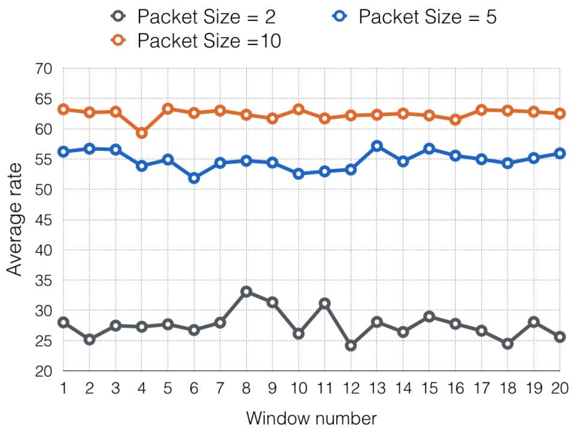

Fig. 4. Characterisation of δ_{ss} for different packet sizes (2, 5 and 10 samples) in terms of communication rates.

TABLE I
CHARACTERISATION OF THE DELAY δ_{ss}.

Size	Avg. delay [sec]	Std. Dev. [sec]	Worst case [sec]
2	0.072	0.033	0.135
5	0.081	0.025	0.135
10	0.159	0.052	0.283

B. Assessment of Computational Performance

Minimising the overall response time of a human-robot interaction interface is a key enabling factor for its usability. As a baseline to ground further studies with humans, we performed a number of functional tests on our architecture to characterise its response times. The overall communication flow from the instant t_a (when a sample data is acquired by the smartwatch) to the instant t_c (when a robot's command can be issued) comprises the delay δ_{ss} introduced by the Bluetooth link between the smartwatch and the smartphone and the delay δ_{sw} introduced by the WiFi connection between the smartphone and the workstation. Furthermore, if the robot is in the `idle` state, we must add the delay δ_i introduced by ROS intercommunication between an *acquisition* module and the *recognition* module on the workstation, and the delay δ_r introduced in the recognition phase. Therefore, the response time Δ^i_{ac} in the *idle* case can be modelled as:

$$\Delta^i_{ac} \equiv t_c - t_a = \delta_{ss} + \delta_{sw} + \delta_i + \delta_r. \quad (4)$$

On the contrary, if the robot is in the `moving` state, it would suffice to add a delay δ_i introduced by ROS intercommunication between the *acquisition* module and a *robot controller* module on the workstation (we use the same δ_i of the `idle` case as a simplification). In this case, the response time Δ^m_{ac} in the *moving* case is modelled as:

$$\Delta^m_{ac} \equiv t_c - t_a = \delta_{ss} + \delta_{sw} + \delta_i. \quad (5)$$

In order to characterise δ_{ss}, we considered the average transmission rates as reported by the mobile app on the smartwatch, when it transmits packets containing 2, 5 and 10 samples, respectively. Each sample includes 6-axis accelerometer (6 $bytes$) and gyro information (6 $bytes$). Figure 4 shows communication rates related to different packet sizes. This estimate is computed using a window size corresponding to 500 samples. Table I shows average delays, their standard deviations and worst cases after taking 1000 samples. In this case, the greater the packet size, the bigger the delay, and the less deterministic the communication.

TABLE II
CHARACTERISATION OF THE DELAY δ_i.

Size	Avg. delay [sec]	Std. Dev. [sec]	Worst case [sec]
2	0.034	0.112	0.72
5	0.018	0.083	0.71
10	0.016	0.078	0.58

The delay δ_{sw} depends on the employed WiFi connection. In our case, we estimate $\delta_{sw} = 0.1$ sec as a worst case.

The ROS command `rostopic hz` has been used to evaluate the intercommunication delay δ_i between the *acquisition* module and the *recognition* or *robot controller* modules, respectively. Table II shows average transmission rates, their standard deviations and worst cases, measured after sending 3000 samples in packets of 2, 5 and 10 sample sizes. In this case, the greater the packet size, the smaller the delay, and the more deterministic the communication.

Although the delay δ_r introduced by the *recognition* module does not affect the `moving` state, as described in (5), it is convenient to keep it bounded to enhance user's experience. An analysis about the interplay between gesture model parameters and classification rates is discussed in [29].

In our case, the delay δ_r originates from two contributions: the basic execution time δ^e_r of the module, and the intrinsic bias δ^b_r introduced by the temporal duration of a gesture, i.e.:

$$\delta_r = \delta^e_r + \delta^b_r. \quad (6)$$

On the one hand, to characterise δ^e_r, we recorded 150 execution time cycles of the *recognition* module. The average computational time is 0.009 sec (roughly corresponding to 112.35 Hz) with a standard deviation of 0.067 sec. On the other hand, δ^b_r is a fixed delay originating by the number of cycles needed for the recognition of a gesture, which is proportional to the number of samples constituting the longest model. For example, if a gesture classification occurs in 100 cycles with an acquisition frequency of 60 Hz, then δ^b_r is roughly 1.66 sec. Since gestures, on average, take at least half of their duration to be recognised, this delay can become an issue for longer gestures. However, for typical gestures, these two estimates indicate that the *recognition* module can be used for run-time recognition.

C. Qualitative Assessment with Humans

To match our design choices with respect to the guidelines introduced in Section II-C, we set-up an experiment in which volunteers must drive the robot along a prescribed path and then park it in a specific area. The experiment involved 27 participants among teachers and students (aged 15 to 60) of the Vocational Education Training school Casa di Carità, Arti e Mestieri of Ovada, Italy. The volunteer must drive

Fig. 5. The circuit used in the experiment with human volunteers.

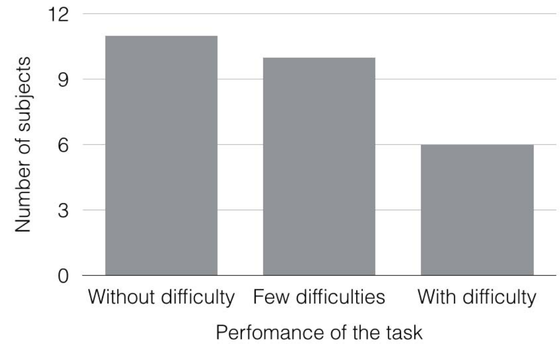

Fig. 7. An experimenter subjective analysis of participants' performance.

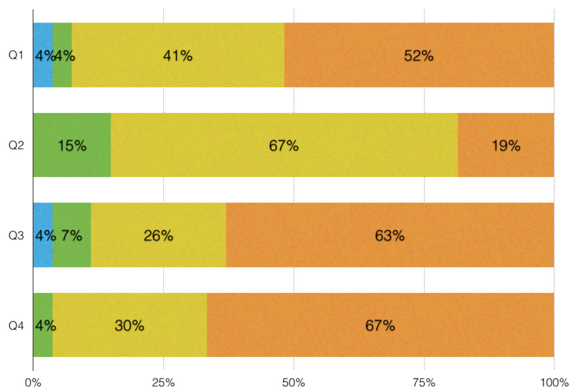

Fig. 6. Results from the post experiment survey.

HRP to complete two laps of the circuit shown in Figure 5 and he is asked to wear the smartwatch on the right wrist. Before each trial, a briefing session occurs in which the experimenter explains to the volunteer the four control gestures and the system's usage. At the end of the trial, the volunteer is requested to fill out a post experiment survey measuring the quality of the proposed gesture-based interface in terms of naturalness of the gestures (2 items), usefulness of the application (1 item), memorability and learnability of the gestures (1 item). Items are rated with a 4-points Likert scale in which 1 corresponds to a strong negative evaluation and 4 corresponds to a strong positive evaluation.

The questions are:

Q_1 Are the gestures comfortable to use?
Q_2 Is the system easy to use?
Q_3 Is the application useful?
Q_4 Is it easy to understand and learn the gestures?

and are aimed at providing an assessment with respect to the guidelines introduced earlier. Figure 6 summarises the participants' responses: *blue* indicates a strong negative evaluation, whereas *orange* stands for a strong positive evaluation. In general 91% of the participants show a positive attitude to Q_1, 85% to Q_2, 89% to Q_3 and 96% to Q_4.

To complement the participants' subjective evaluation, during the test an experimenter noted whether they experienced problems in performing the gestures, or whether they had difficulties in driving the robot along the path. The results of these observations are shown in Figure 7. Observations have been classified in three categories: *without difficulty* comprises all volunteers who performed the test with no evident issues (i.e., neither with the gestures nor with the driving), *few difficulties* includes those who were able to drive the robot along the path, but experienced issues in performing the gestures, and *with difficulty* comprises all subjects who had issues with both the gestures and the driving. Overall, 11 participants had no difficulties, 10 had a few difficulties, and 6 had difficulties. Participants took 97 to 187 sec to complete the two-laps circuit, with an average of 123.25 sec and a standard deviation of 23.53 sec.

D. Discussion

The post experiment survey reveals that most participants found the proposed gestures intuitive and comfortable, as shown by Q_1, with a mean of 3.4. As Figure 6 and Figure 7 show, whilst the experimenter reported that 22% of the participants experienced difficulties in driving the robot with the proposed interface, only 15% rated the interface as difficult to use in Q_3. This difference may be due to a different interpretation of difficulties between the participants and the experimenter, an acquiescence effect in participants' responses, or a severity bias of the experimenter. Question Q_3 allows for measuring the participants' appreciation, in this aspect the proposed interface has a mean of 3.48. The system is considered easy to use by the majority of the participants, as shown by Q_2, with a mean of 3.04. Finally, the results of Q_4 indicate that most of the participants find the interface easy to learn with a mean of 3.62. Given the selection of gestures, these results seem to support our claims in Section II-C.

We did not observe any significant correlation between the time spent to complete the control task and the self-rated difficulty of the participants. Moreover, we noted that most participants chose precision over speed, usually driving the robot at a slower-than-maximum pace to ease control.

V. Conclusions

In this paper, we discuss an approach to gesture-based robot control using wearable devices. The use of simpler or even novel forms of interaction modalities for human-robot interaction scenarios is a challenging research direction, which is still unexplored for the most part. Although a bottom-up approach to select an appropriate set of gestures could be considered, we argue that in order to control a robot in a natural and effective way, it is necessary to adhere to well-defined design principles, which have been defined in related research and industrial fields.

Here, we propose a system to guide a mobile robot via gestures obtained using wearable devices. We motivate our design choices on the basis of practical guidelines inspired by the automotive industry. The framework, available open source, has been tested both with respect to performance aspects and considering qualitative human evaluation. Specifically, 27 previously untrained volunteers shared their experiences in using our framework. It is noteworthy that the system has been showcased at the IEEE ICRA 2016 conference in Stockholm, Sweden, where conference attendees had the opportunity to give it a try at the exhibitions.

References

[1] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[2] F. Karray, M. Alemzadeh, J. A. Saleh, and M. N. Arab, "Human-Computer Interaction: overview on state of the art," *International Journal on Smart Sensing and Intelligent Systems*, vol. 1, no. 1, p. 137159, 2008.

[3] G. M. Bandeira, M. Carmo, B. Ximenes, and J. Kelner, "Using gesture-based interfaces to control robots," in *Human-Computer Interaction: Interaction Technologies*, ser. Lecture Notes in Computer Science, 2015, vol. 9170, pp. 3–12.

[4] F. Boniardi, A. Valada, W. Burgard, and G. D. Tipaldi, "Autonomous indoor robot navigation using a sketch interface for drawing maps and routes," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[5] D. Wigdor and D. Wixon, *Brave NUI World: designing natural user interfaces for touch and gesture*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[6] J. Preece, H. Sharp, and Y. Rogers, *Interaction design: beyond human-computer interaction*, 3rd ed. New York City, USA: John Wiley & Sons, 2015.

[7] K. Qian, J. Niu, and H. Yang, "Developing a gesture based remote human-robot interaction system using Kinect," *International Journal of Smart Home*, vol. 7, no. 4, pp. 203–208, 2013.

[8] X. Wang, P. Tarrío, E. Metola, A. M. Bernardos, and J. R. Casar, "Gesture recognition using mobile phones inertial sensors," in *Distributed Computing and Artificial Intelligence*, ser. Advanced in Intelligent and Soft Computing. Heidelberg, Germany: Springer, 2012, vol. 151, pp. 173–184.

[9] H. K. Yap, B. W. K. Ang, J. H. Lim, C. H. James, and C.-H. Yeow, "A fabric-regulated soft robotic glove with user intent detection using emg and rfid for hand assistive application," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[10] C. Rossa, J. Fong, N. Usmani, R. Sloboda, and M. Tavakoli, "Multi-actuator haptic feedback on the wrist for needle steering guidance in brachytherapy," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 852–859, 2016.

[11] M. Kim, J. Lee, and K. Kim, "Tele-operation system with reliable grasping force estimation to compensate for the time-varying semg feature," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[12] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the JPL BioSleeve," in *Proceedings of the 2013 International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[13] S. Radmard, A. J. Sina, and E. A. Croft, "Interface design and usability analysis for a robotic telepresence platform," in *Proceedings of the 2015 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Kobe, Japan, 2015.

[14] G.-C. Luh, H.-A. Lin, Y.-H. Ma, and C. J. Yen, "Intuitive muscle-gesture based robot navigation control using wearable gesture armband," in *Proceedings of the 2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, Singapore, SG, January 2015.

[15] IDC Research Inc, "Forecasts worldwide shipments of wearables to surpass 200 million in 2019, driven by strong smartwatch growth," Tech. Rep., 2016. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS40846515

[16] A. H. Johnston and G. M. Weiss, "Smartwatch-based biometric gait recognition," in *Proceedings of the 2015 IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Washington DC, USA, September 2015.

[17] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: a machine learning approach," in *Proceedings of the 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Las Vegas, Nev., USA, February 2016.

[18] D. A. Norman, *The design of everyday things*. Cambridge, MA: MIT Press, 1988.

[19] S. C. Seow, "Information theoretic models of HCI: a comparison of the Hick-Hyman law and Fitts law," *Human-Computer Interaction*, vol. 20, no. 3, pp. 315–352, 2005.

[20] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information." *Psychological Review*, vol. 63, no. 2, p. 81, 1956.

[21] D. A. Norman, "Natural user interfaces are not natural," *Interactions*, vol. 17, no. 3, pp. 6–10, 2010.

[22] A. Malizia and A. Bellucci, "The artificiality of natural user interfaces," *Communications of the ACM*, vol. 55, no. 3, pp. 36–38, 2012.

[23] D. Kern and A. Schmidt, "Design space for driver-based automotive user interfaces," in *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications (Automotive UI)*, Essen, Germany, September 2009, pp. 3–10.

[24] European Commission, "Safe and efficient in-vehicle information and communication systems: a European statement of principle on human machine interface," Tech. Rep., 1999. [Online]. Available: http://cordis.europa.eu/pub/telematics/docs/tap_transport/hmien.pdf

[25] Driver Focus-Telematics Working Group, "Statement of principles, criteria and verification procedures on driver interactions with advanced in-vehicle information and communication systems," Tech. Rep., 2006.

[26] X.-H. Wu, M.-C. Su, and P.-C. Wang, "A hand-gesture-based control interface for a car-robot," in *Proceedings of the 2010 IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[27] G. Geiser, "Man machine interaction in vehicles," *ATZ: Automobiltechnische Zeitschrift*, vol. 87, pp. 74–77, 1985.

[28] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[29] ——, "Human motion modelling and recognition: a computational approach," in *Proceedings of the 2012 IEEE International Conference on Automation Science and Engineering (CASE)*, Seoul, Korea, August 2012.

[30] S. Salvador and P. Chan, "Toward accurate Dynamic Time Warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[31] B. Bruno, F. Mastrogiovanni, A. Saffiotti, and A. Sgorbissa, "Using Fuzzy Logic to enhance classification of human motion primitives," in *Proceedings of the 2014 International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Montpellier, France, July 2014.