Chapter 5
Integrating Smart Mobile Devices for Immersive Interaction and Control of Physical Systems: A Cyber-Physical Approach

Jared A. Frank and Vikram Kapila

Abstract The embedded technologies integrated into smart mobile devices are becoming increasingly more powerful and being applied to solve disparate societal problems in unprecedented new ways. Billions of smartphones and tablet computers have already reshaped the daily lives of users, and efforts are currently underway to introduce mobile devices to some of the most remote and impoverished areas of the world. With an ever-expanding list of sensors and features, smartphones and tablet computers are now more capable than ever of enhancing not only our interactions with software and with each other, but with the physical world as well. To utilize smart mobile devices at the center of rich human-in-the-loop cyber-physical systems, their sensing, storage, computation, and communication (SSCC) capabilities must be examined from a mechatronics perspective rather than the contexts in which they are conventionally treated (e.g., messaging, surfing the web, playing games, navigation, and social networking). In this chapter, we discuss how state-of-the-art mobile technologies may be integrated into human-in-the-loop cyber-physical systems and exploited to provide natural mappings for remote interactions with such systems. A demonstrative example is used to show how an intuitive metaphor is uncovered for performing a balancing task through the teleoperation of a ball and beam test-bed.

Keywords Ball and beam • Control • Cyber-physical system • Device • Immersive • Interaction • Interface • Mobile • Smartphone • Tablet • Teleoperation • User

1 Introduction

Our modern technological age abounds with user interfaces to interact with a myriad of technologies that we encounter in our daily lives (e.g., kitchen appliances, consumer electronics, office machines, automobiles, etc.). When a user interface

---

J.A. Frank • V. Kapila (✉)
NYU Tandon School of Engineering, Brooklyn, NY 11201, USA
e-mail: jared.alan@nyu.edu; vkapila@nyu.edu

D. Zhang, B. Wei (eds.), *Advanced Mechatronics and MEMS Devices II*,
Microsystems and Nanosystems, DOI 10.1007/978-3-319-32180-6_5

is intuitive, users are able to draw from their prior experiences and familiarity to immediately use the interface effectively. This allows the interface to become transparent to the user and the user's attention to be directed towards performing the task rather than trying to learn how the interface is used. That is, when interfaces are made intuitive, users can perform tasks with ease, comfort, and delight.

Besides the traditional button- and knob-based interfaces that we encounter on a daily basis, people now interact with graphical interfaces that exist on mobile devices. Not only are these interfaces used everyday by the public to accomplish the tasks of the internet age (e.g., accessing email, searching the internet, social networking, listening to music, and checking the weather), the inherent mobility of smartphones and tablets has led to a growing list of novel interactive activities designed to socially connect people (e.g., tweeting, snapchatting, location-based dating, etc.). However, the overwhelming number of these interactions occurs in cyberspace, without any physical element. That is, such interfaces typically enable people to virtually interact either with each other or with digital information rather than with physical systems like machines.

Society is experiencing an escalation in the complexity of engineering systems. Machines that were once simple are growing ever more complex and occupying new application domains. In particular, a host of robotic technologies that has already penetrated industry is expected to appear both in the home and at the workplace in the near future. Although sophisticated machines like robots have been remotely controlled, or teleoperated, by trained technicians for many years, their painless adoption by the general public will require nontechnical and inexperienced consumers to be provided with the most intuitive user interfaces possible. With the steady advancement of embedded sensing, storage, computation, and communication (SSCC) capabilities, smart mobile devices like smartphones and tablets are more capable than ever to serve as platforms for providing immersive interactions with such systems. Moreover, because of the already significant popularity and familiarity of smart mobile devices, such interfaces may be accessible to users with little to no additional cost or training.

Human-in-the-loop cyber-physical systems, in which user input, computation, communication, and control of physical dynamics are more intimately interconnected than in traditional teleoperation systems, have the potential to augment the user's interaction with the physical world [32]. However, the incorporation of a mobile device as a component in such a system is not a trivial undertaking. It involves the synergistic integration of the mobile device hardware and software towards a high-level objective that goes beyond the conventional purposes for which the embedded technologies have been designed and used. New challenges include how the SSCC capacity of mobile devices can be used to (1) capture and map user behavior to desired behavior of the physical system to support the creation of intuitive metaphors, and (2) capture and display the state of the physical system to the user to support situational awareness. These considerations are essential in achieving a level of immersion that enables the user to effectively transfer acquired skills in interacting with mobile interfaces to successfully operate the physical system. Note that to accomplish such rich interactions between human users and

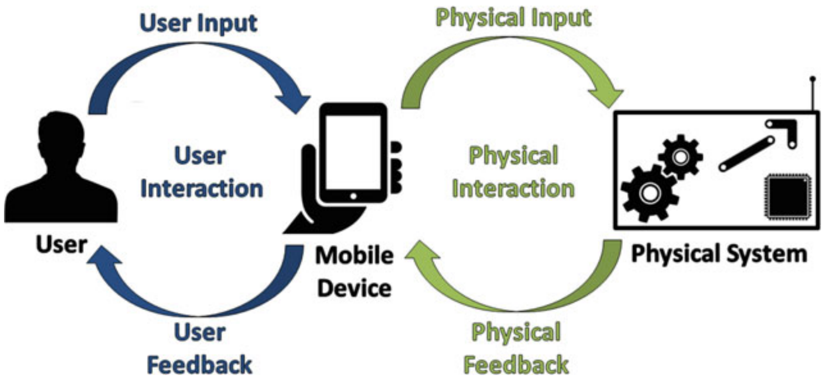

Fig. 5.1 Architecture of a cyber-physical system that enhances user interaction with a physical system using a mobile device

physical systems, the mobile device occupies the center of two interdependent cyber-physical loops (see Fig. 5.1). In this architecture, the properties of the physical system (e.g., stability and performance of the physical process) and conditions of the user experience (e.g., responsiveness and usability of the user interface) can become intimately coupled and dependent on the speed and reliability with which the mobile device processes, stores, and communicates information between the user and the physical system. Therefore, both the physical and user interaction loops impose demands on the hardware and software of the mobile device as it is used to close the two loops.

After the physical process is constructed, the mobile application is developed, and the functionality of the components is tested, a successful interaction is contingent upon the usability of the system. With a human-in-the-loop system, the outcome is difficult, if not impossible, to predict *exactly* by means of theory, computation, or simulation. Therefore, the performance of the closed-loop system must be expressed in terms of quantitative metrics and studies must be conducted in which users interact with the system. In this chapter, the development of a ball and beam system for studying rich immersive interactions using a smartphone is described. The work presented in this chapter is based on [14].

The chapter is organized as follows. In Sect. 2, the motivation behind the integration of smart mobile devices to build human-in-the-loop cyber-physical systems is discussed. Section 3 describes the hardware technologies of mobile devices that make them suitable for interaction with and control of physical systems. Then, Sect. 4 discusses the software aspects, including mobile operating systems and mobile application development, that allow for the immersive interaction with and control of physical systems using smart mobile devices. Next, Sect. 5 discusses the development of a human-in-the-loop cyber-physical system in which a mobile application is used to uncover an intuitive interaction metaphor for performing a remote balancing task through the direct control of a ball and beam test-bed. Finally, Sect. 6 offers concluding remarks and future directions.

2 Motivation

The release of the first smartphones with diverse sets of embedded sensors in 2007 was soon accompanied with the development of mobile interfaces that utilize those sensors to monitor users' physical activities, heart rate, and driving style [5]. These mobile applications have brought attention to the use of mobile devices as handheld measurement systems and have come to monitor much more than just the user of the device, impacting areas such as healthcare, transportation, and environmental monitoring [22, 25].

Almost immediately after exploring the measurement applications of smart mobile devices, development began of an even more compelling class of mobile applications that allow users to operate physical systems. Of these systems, the overwhelming majority are classified as robots or unmanned vehicles and used to extend the user's physical influence in a remote environment [1, 3, 8, 16, 23, 34]. These efforts are in many ways a continuation of telerobotics work with earlier generations of mobile technologies, such as personal digital assistants [11, 38]. Regardless of the interface hardware or software, users traditionally interacted with physical systems through a direct control approach, in which the physical system exhibits little to no autonomy and thus all the monitoring, decision making, and planning are performed by the user. Although this scheme was quick and easy to implement, direct control presented several problems in time-critical applications with significantly delayed communications that were resolved with the introduction of supervisory control schemes [10, 33]. Supervisory control, in which the physical system remains semi-autonomous and thus relatively stable in the face of communication delays, permits the user to intermittently monitor and intervene with the control of the physical system, shifting the focus to remotely commanding systems at a high level of abstraction.

Advances in the SSCC capabilities of mobile devices like smartphones and tablets have led to interest in their use as platforms for cyber-physical systems [2, 37]. Unlike traditional embedded systems, these systems give rise to a number of conceivable architectures in which the roles of sensing and computation are distributed amongst a network of heterogeneous interacting elements [21]. Mobile interfaces designed to operate as part of human-in-the-loop cyber-physical systems may enable immersive interactive experiences with physical systems using any desired control scheme. Regardless of the chosen control scheme, low-level interrelationships will exist between the SSCC capabilities of the mobile device, characteristics of the physical dynamics of the engineering system, and aspects of the user experience. These interrelationships are of particular interest in direct control architectures, since although the direct control scheme is well suited for applications that require real-time human decision making and that can support low-delay communications, their effects can be destabilizing. By addressing these challenges on mobile platforms, the performance of more challenging teleoperation tasks with relatively unstable systems may be possible, such as the balancing task demonstrated in this chapter.

Moving some of the computational and communicative load onto the mobile device can potentially reduce demands such as the size, cost, complexity, and energy usage of the engineering system being controlled. However, this comes at the expense of increasing the sensitivity of the system to user actions and the network performance, and adding an additional layer of complexity to the user interface design problem. In particular, mobile interfaces must provide users with services such as intuitive interaction metaphors, a variety of feedback modalities, and a bidirectional non-blocking communication routine for exchanging information with the physical system, while remaining responsive and meeting timing constraints to maintain the stability of the physical process. To achieve this goal, the hardware and software components of the devices must be assessed with respect to the roles they are expected to serve in these systems.

3 Smart Device Hardware

Although the performance and user experience associated with utilizing smart mobile devices in the interaction and control of physical systems depend on many factors, the SSCC capabilities of the mobile device are essential to the successful operation of the system. The requirements of the hardware will vary depending on the physical process to be controlled and in some cases may be beyond the capabilities of the current generation of devices. For this reason, developers should stay up-to-date on the capabilities and limitations of the device hardware, as they continue to advance at an accelerated rate.

3.1 Sensing

To utilize a smart mobile device to interact and control physical systems, the ability to accurately capture information from the user or from the physical system is of paramount importance. As is the trend in electronics, sensors continue to be made smaller, more powerful, and affordable. This is good news, since sensors have become an integral part of shaping the mobile experience. Many phones now contain sensors to measure everything from the location, motion, and orientation of the device; ambient temperature, light, pressure, and humidity; as well as the proximity, fingerprint, footsteps, voice, and heart rate of the user. Although not all sensors are embedded in all devices, most if not all of the relevant technologies for extracting users' gestures are now standard on modern devices.

Touchscreen When it comes to user interaction, the device screen plays a pivotal role. Not only must it be used to provide the main source of feedback to the user to support situational awareness during the interaction, but the screen also serves as one of the most important sources of user input. Unlike most touchscreens, the touchscreens of modern smartphones and tablets are based on capacitive sensing technology. This means that touchscreens have not only become more sensitive, responsive, and accurate than previous generations of resistive screens, but their displays have become much sharper as well. This has resulted in the creation of rich touch and multi-touch interactions with mobile interfaces.

Inertial Sensors People have been using forms of gestures to communicate non-verbal messages to each other since the beginning of history. By using inertial sensors such as 3-axis accelerometers, gyroscopes, and magnetometers, mobile device manufacturers have made it possible to accurately capture the movements of devices (e.g., tilts, shakes, rotations, and swings) and to recognize gestures from movement data. Moreover, the ability to recognize gestures from movement data empowers mobile devices to act as extensions of the user and is beginning to play an important role in creating a range of new promising interaction possibilities. With the integration of dedicated motion coprocessors, devices can now collect, process, and store motion data without burdening the application processor.

3.2 Storage

The need to collect and process data from the user and from engineering systems on smart mobile devices necessitates the ability to store and work with large amounts of data. With mobile devices that contain up to 128 GB in flash memory and as much as 4 GB of RAM, this is not a problem on current platforms.

3.3 Computation

Many aspects of the performance and user experience associated with interacting and controlling a physical system from a smart mobile device are closely linked to the computational power of the device. During the interaction, the device is expected to collect and store data from a multitude of sensors, to execute a multitude of operations on the data, to render 2D or even 3D graphics to display relevant data to the user, and to communicate information with the physical process, oftentimes simultaneously and many times per second. In this way, the processor plays a central role in satisfying both functionality and usability requirements of the system. Manufacturers are now producing systems-on-a-chip (SoCs), in which dual- or even quad-core processors, running at rates several times faster than their first released models, are integrated with graphics processors, coprocessors, GPS, cellular modems, and memory on a single board. The SoC not only allows the size of the device to be reduced, but also dramatically improves speed and power usage. In fact, the computational power of mobile devices has quickly caught up to the level of desktop PCs and has been responsible for a shift towards a mobile computing era in which smartphones and tablets are the central computing devices in people's lives.

3.4 Communication

If the interactions discussed in this chapter were just between the user and the mobile interface, then simply considering the sensing, storage, and computation of the mobile device would have been sufficient. However, wireless connectivity is necessary to communicate data between the mobile device and the engineering system. Support for the latest generations of mobile, Wi-Fi 802.11, bluetooth, infrared, and near-field communications enables mobile devices to access and share information with other devices at incredible speeds and over a variety of ranges. This communication will introduce small delays to the closed loop, which must be minimized as they negatively affect the stability and performance of the physical system as well as the responsiveness of the interface.

4 Software

The purpose of mobile device software is to harness the embedded hardware in providing rich intuitive user experiences. This includes the mobile operating system, which provides the hardware drivers, the libraries, frameworks, and application programming interfaces (API) so that developed applications have access to the sensors, features, and data available on the device. It also includes the mobile applications themselves, which make use of the available sensors, features, and data to provide specialized functionality to the user. However, the design of interfaces that enable users to interact with physical systems presents an additional layer of complexity: the user must directly interact with the application on the mobile device to indirectly interact with the physical system. In other words, mobile interfaces must provide natural mappings from the user's actions on the interface to the commands that will be communicated to influence the state of the physical system. In addition, the interface must provide clear feedback that can be used to interpret the state of the physical system. It is natural to expect that the choice of mappings and feedback will be dictated by the nature of the physical system. For assistance, developers must leverage the research and heuristics that have been gathered on the design of teleoperation systems [33] as well as on the design of mobile interfaces in general [31, 36].

A well-designed interface must reinforce users' assumptions and expectations of how the interface is to be used and their mental models of how the physical system will behave in response to their actions. To avoid overwhelming users, the low-level details of the system ought to be hidden from the user and replaced with high-level metaphors that may be familiar to the user [30]. The choice of interaction metaphors may be inspired from the physical nature of the system to give the user the most natural experience when using a mobile interface (e.g., exploiting a spatial analogy wherein a physical object moves up or down when the user moves an interactive object on the interface up or down, respectively). These interaction metaphors can be realized with gestures extracted from the touchscreen or from inertial sensors. Such metaphors have been shown to be effective in designing interactions using the same sensors and features on tangible user interfaces for human–robot interaction [18], in gaming technology to build natural interfaces that enrich the user interaction with desktop computers [12], and game controllers for introducing the elderly to video games that keep them healthy through physical activity [17]. These applications demonstrate that user interfaces with intuitive metaphors have the potential to make remote interaction with physical systems more natural.

The design of metaphors for interacting with a particular system poses an open-ended, creative design challenge that lacks a unique solution. Thus, it is important to rigorously examine each of the metaphors that may provide a natural mapping. This requires a comparative study to determine the interaction metaphor that not only feels most natural to users but that also yields acceptable performance of the physical process.

5 Putting It All Together: Control and Interaction with a Ball and Beam Test-Bed

We will now discuss the development of a human-in-the-loop cyber-physical system that implements a smartphone to enable immersive interaction with a ball and beam test-bed. During the interaction, the control of the angular orientation of the motorized beam is maintained by a laboratory station built around a PC-based data acquisition and control board (DACB). The goal is to use the mobile interface on the smartphone to monitor and command the orientation of the beam such that the ball remains balanced at the center of the beam. In other words, while the stability of the beam dynamics is ensured by the laboratory station, the ball dynamics are intimately related with user behavior, computation, and communication in the system. To determine the most effective metaphor for completing this task, a user study is performed wherein participants are asked to perform the task using several designed metaphors. A survey is conducted with the participants to determine the user satisfaction with each metaphor. The experimental data shows that the preferred metaphor of the participants is the one in which the smartphone is tilted to mimic the desired tilting of the beam, and is the same one which yields significantly better task performance.

5.1 System Description

The ball and beam test-bed consists of a DC-motor, a 0.5-m long lexan plastic beam, and a smooth 1 in. (0.0254 m) diameter steel ball. The output shaft of the DC-motor is attached to a gearbox to produce sufficient torque to drive the beam mounted to the output shaft. Attached to the shafts of the gearbox are a potentiometer and an absolute encoder for measuring angular displacement and a tachometer for measuring

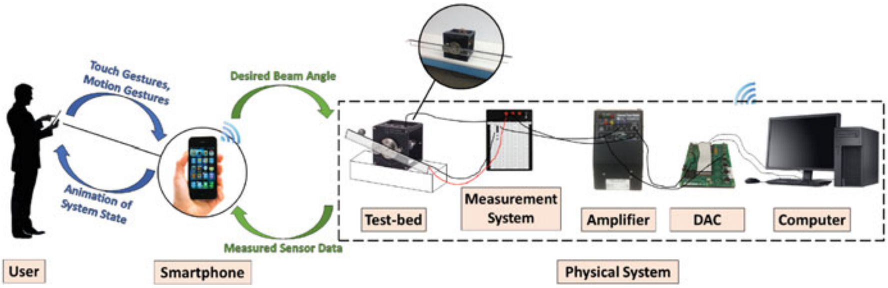

Fig. 5.2 The human-in-the-loop cyber-physical system for interacting with the ball and beam test-bed using a smartphone

angular velocity. A setup with conductive and resistive strips attached to a track on the beam, and associated electronics, is used to measure the position of the ball along the beam. A computer running MATLAB/Simulink wirelessly communicates with the mobile device and stabilizes the DC-motor to the desired orientation using an optimal control algorithm (see Appendix), a PC-based DACB, and a power amplifier. Figure 5.2 shows the complete human-in-the-loop cyber-physical system, its components, and the data communicated between its major constituents.

5.1.1 Measurement System

To control the angle of the beam attached to the output shaft, measurements of the angular position and velocity of the output shaft are, respectively, acquired using the encoder and a tachometer mounted to the test-bed. For the user to be provided with an animated display of the state of the system on the mobile interface for situational awareness, and to evaluate the performance of the balancing task, a measurement system is designed and integrated with the test-bed to supply measurements of the position of the ball to the desktop computer. The circuit shown in Fig. 5.3 is connected to the desktop computer via the DAC board. More information about the setup of the DAC board is found in [20]. A carbon-based resistive strip and a copper-based conductive strip are used to create a linear potentiometer on the beam with the ball acting as the wiper. Since the analog channels of the DAC board have a range of ±10V, an operational amplifier is used to bring the output of the linear potentiometer to this range. Two 10 kW potentiometers are used in the operational amplifier circuit to trim the gain and offset of the output voltage.

5.2 User Interface Design

The smartphone-based user interface for interacting with the ball and beam test-bed is developed to run on an Apple iPhone 5, with iOS as the mobile operating system. When the application is started, the user first encounters the main menu

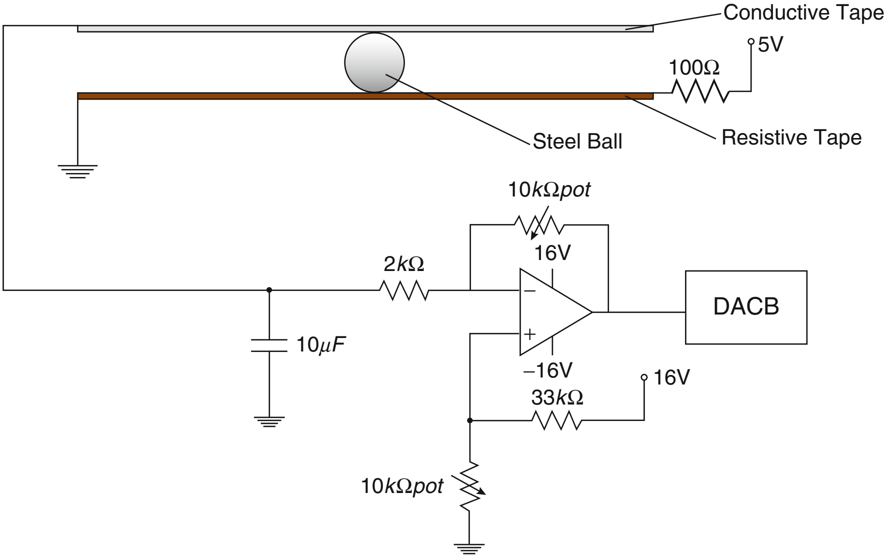

Fig. 5.3 The system for measuring the position of the ball along the length of the beam

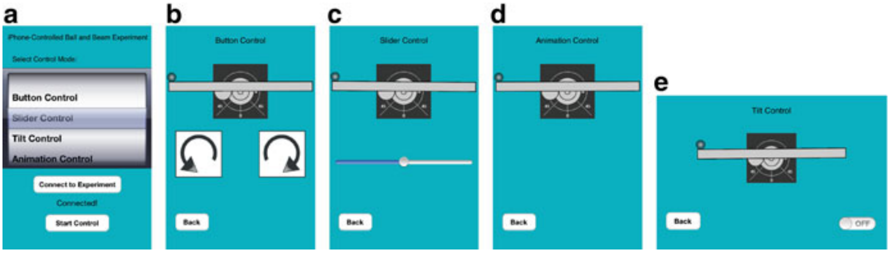

Fig. 5.4 Screenshots of the smartphone application, including the (**a**) main menu and the interface modes for controlling the ball and beam test-bed using (**b**) buttons, (**c**) slider, (**d**) direct manipulation of the animation, and (**e**) tilt

screen (see Fig. 5.4a), where she may select an interface mode to interact with the test-bed and press a button to establish a wireless connection with the test-bed using Wi-Fi 802.11. To allow for ease of navigation, the layout of the interface is inspired by standard utility applications that users are familiar with, while the design of the interactive interface is modeled after the look and feel of game applications, the most widely downloaded class of mobile applications [29]. A rich two-dimensional graphical animation of the ball and beam test-bed is included in the interface, and updates in real time as the application receives sensor data from the test-bed. Inputs from the touchscreen and from the inertial sensors on the device are used in designing the interface modes. Specifically, four different interface modes are made available to interact with the ball and beam test-bed, including tapping virtual buttons, dragging a virtual slider, directly manipulating the animation by tapping and dragging on the virtual beam, and tilting the device as if it were the beam.

These interface modes resulted from (1) a narrow selection process in which more than twice the number of interactive alternatives was evaluated, followed by (2) a series of refinements in which the final values were chosen for parameters governing the mappings. These steps helped to discover usability problems associated with the interface, make predictions concerning tendencies in user behavior, and establish a set of context-specific heuristics to provide a consistent level of usability for future applications [24]. To conduct a study in which the design choices can be validated and the user interface modes can be compared, each mode was used independently and provided in separate views (Fig. 5.4b–e).

Buttons With the first user-selectable interface mode (Fig. 5.4b), two buttons are pressed to drive the beam clockwise and counterclockwise. Several alternative mappings from the button presses to test-bed commands were considered in the design of this mode. Inspiration for the mappings was drawn from the developers' experience with teleoperation systems and video games. Some of the mappings were based on distinct taps on the buttons and others required holding the buttons down. Finite angular displacement commands, velocity commands, and acceleration commands to the beam in the directions indicated by the symbols on the buttons were considered. Ultimately two of the mappings were combined to enable the user to tap the buttons to rotate the beam in increments of 1 ° at a time or hold the buttons to drive the beam at a quick but constant angular velocity (80 °/sec). This gave the user the ability to make fine adjustments to the beam angle as well as liberally drive the beam depending on the situation.

Slider In the second user-selectable interface mode (Fig. 5.4c), a slider is dragged across the screen to command the orientation of the beam. In this case, the major design decisions focus on the orientation of the slider (horizontal versus vertical), and the sensitivity of the mapping, which is dictated by the length of the slider and the range of values covered by the slider. Preliminary evaluations revealed that to balance the ball, the beam needs to remain in a region ±10 ° from its horizontal orientation. Also, users tend to make larger sweeps of the slider as opposed to smaller ones. Thus, a slider that spans the entire width of the screen was used to command the beam to within the appropriate angular region (±10 °), providing users the needed sensitivity to make quick, fine adjustments to beam orientation.

Animation In the third user-selectable interface mode (Fig. 5.4d), the user can interact with the beam by directly interacting with a one-ninth scale animation of the beam using simple gestures on the touchscreen. Several alternative mappings were considered when designing this direct manipulation mode, including tapping on the screen in order for the animated beam to align itself with the user's finger and dragging the animated beam to the desired orientation for reorienting the beam. The main issues encountered during the design of this interface mode included whether users would choose to tap or drag the animation and where on the screen they would perform these actions. Due to the resolutions of the screen and the beam animation, it is more difficult to command the orientation of the beam when interacting with the animation near the center of the beam. Thus, the distance in pixels between the user's finger and the center of the beam is used to reduce the sensitivity as the user interacted closer to the center of the beam. Interestingly this solution of the usability problem produces a virtual moment arm effect. In fact, the results of preliminary evaluations reveal that many users' mental models already include this moment arm effect, as most users tend towards interacting with the animation at the ends of the beam. This interaction provides the user with the sensation that she is grabbing the actual beam by one end and tipping it upward or downward. Moreover, preliminary evaluations predicted that, in general, users tend to make large sweeping movements across the screen when required to manipulate the beam quickly to complete the exercise. Therefore the sensitivity of the interaction was lowered by reducing the ratio between the output (angle of the beam with respect to the horizontal) and the input (vertical finger motion in pixels) to $0.06\,^{\circ}/\text{pixel}$. This allowed users to make fine adjustments with large drags of the fingers on the screen. Although it meant that users were not tapping exactly at the angle on the animation where they would like the actual beam to be oriented, this did not have a significant impact on users' ability to control the beam, since they mostly relied on visual feedback from the test-bed to orient the beam.

Tilt The fourth user-selectable interface mode (Fig. 5.4e) uses the device's built-in accelerometer sensor to command the orientation of the beam. This is based on the underlying assumption that the only acceleration experienced by the device accelerometer is due to gravity [9]. By tilting the mobile device, the user is provided with an interaction metaphor that gives her the sensation that she is actually tilting the beam itself. The tilt-based mode includes a toggle switch to turn the control of the beam on or off, to prevent commands from being sent to the test-bed prematurely until the user has correctly oriented the smartphone. The main design parameter of this interaction metaphor is the ratio between output (the angle of the beam with respect to the horizontal) and input (the angle of the mobile device). This ratio significantly impacts the sensitivity of the interaction. Originally, the ratio was designed to be 1:1; however, the results of preliminary evaluations indicated that users tend to exert large rotations on the device when required to manipulate the beam quickly to complete the exercise. Therefore, the sensitivity of the interaction is reduced to a ratio of 1:3 to improve performance. This allowed users to still make fine adjustments to beam orientation with generous motions of the device.

The mappings of the different interface modes for human–machine interactions presented above are not unique. It is certainly possible to explore alternative mappings and metaphors to control the system. For example, prior research has investigated additional modes, such as knobs and joysticks, for the control of mobile robots [26]. In addition, prior research has also investigated the use of vision as a modality to foster intuitive user interactions with educational laboratory test-beds [15]. However, the focus of this work is to illustrate how the hardware and software elements embedded in mobile platforms can be used in human-in-the-loop cyber-physical interactions and to lay a foundation for further research into designing such interactions. A longitudinal study, in which each of the modalities is iteratively improved upon, will be considered in future work.

5.3 User Study

To investigate the usability of the designed smartphone interface, validate the potential of utilizing smartphone hardware and software in systems such as the one described in this chapter, and determine the most effective interaction metaphor for completing the balancing task, a study is conducted with 21 participants (20 male, 1 female), all of similar age and technological experience (engineering students between 19 and 26 years old). After completing a preliminary survey, each participant uses each of the four interface modes three times (for a total of 12 trials per participant and 252 total trials) to control the orientation of the beam to balance the ball as close to the center of the beam as possible for 20 s. During the interaction, participants stand directly in front of the test-bed and have the choice to either look at the animation of the test-bed on the smartphone screen or to look at the actual test-bed. The order in which the participants use the interface modes is randomized to prevent any order effect on the results of the study. Specifically, the four interface modes yield 24 permutations for the order in which they can be used. Thus, no two of the 21 participants are assigned the same order. At the beginning of each trial, the ball begins at the left end of the beam. When the user is ready, she may use the interface to bring the ball towards the center of the beam, at which point the twenty-second timer is started. The task is completed once either the twenty seconds have ended or the ball has fallen from either end of the beam. As participants complete the exercise, the position of the ball is measured and recorded by the desktop computer in MATLAB. The outcome of the task (whether the ball remains balanced or not) is also recorded. The performance metrics are the score (1 if the ball remains balanced after twenty seconds and 0 otherwise) and the root mean squared deviation (RMSD) of the ball from the center of the beam for those trials that are successful. Of the users who successfully keep the ball balanced, those who have a more difficult time balancing the ball yield larger values for the RMSD and those who have an easier time keeping the ball balanced near the center yield smaller values for the RMSD. After the participants complete the task with each of the interface modes, they complete a final survey in which they indicate their level of disagreement/agreement (on a scale of 1–5) with statements concerning the ease of use and satisfaction with the interface. Participants are encouraged to provide written comments and suggestions, which are used to determine which interface mode each user preferred. The experiment is conducted with each participant according to the following protocol:

1. The participant completes a preliminary survey.
2. The participant is introduced to the smartphone application.
3. The order in which the participant will use the interface modes is generated.
4. The participant has three twenty-second attempts to balance the ball on the beam using the first interface mode.
5. Step 4 is repeated for the remaining three modes.
6. The participant is given the final survey which asks about participant's experience in using each of the modes.

As discussed in [27], user input to a machine can be in a continuous or a discrete format. Since the slider, animation, and tilt interaction modes yield continuous format data and the button mode yields discrete format data, sensitivities of different modes producing data of different formats are not amenable to comparison. Moreover, due to differences in nature between modalities (i.e., tilting the device or dragging a finger on the screen), sensitivities of different modes producing data of same formats are also not amenable to comparison. Thus, in this work, the sensitivity of each mode has been treated as an adjustable parameter that is critical to the performance and acceptance of a particular interaction metaphor, but that needs to be studied independently through user testing.

5.3.1 Quantitative Results

Each of the 21 participants in this study is successful in balancing the ball on the beam at least one time. However, eight of the participants balance the ball successfully by using only one of the interface modes, and seven of these eight participants are successful with only the tilt mode. Figure 5.5a shows the percentage of participants who are successful with each of the interface modes. It is important to note that none of the 21 participants successfully use the button mode even once to complete the exercise. Regarding the remaining three modes, percentages of successful participants are as follows: 28.6 % for the slider mode, 19 % for the animation mode, and 71.4 % for the tilt mode. Therefore, the tilt mode is undoubtedly superior to the other three mode in terms of the success rate of participants. Of the 252 trials, 75 are successful. Figure 5.5b shows the percentage of these 75 successful trials that each of the four interface modes are responsible for. Note that 60 % of the successful trials result from the use of the tilt mode, which is higher than the combined number of successful trials resulting from the animation and slider interface modes.

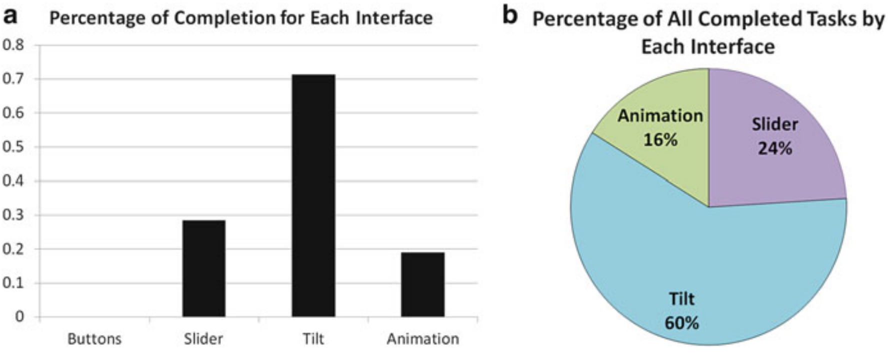

Fig. 5.5 Success results, including (**a**) percentage of successful participants for each mode and (**b**) percentage each mode is responsible for the total number of completed trials

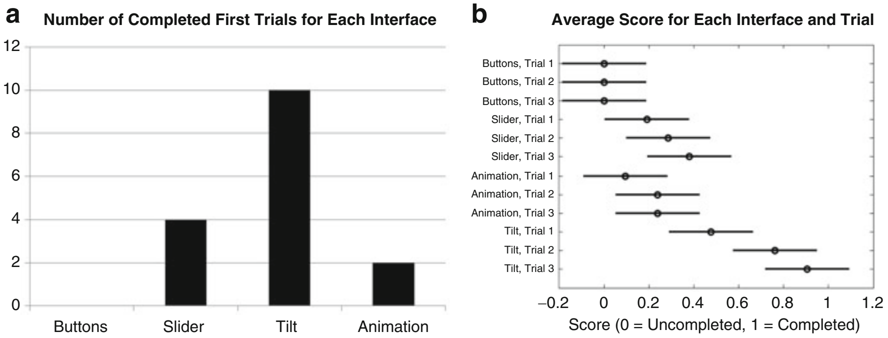

Fig. 5.6 Intuitiveness and learnability results, including (**a**) number of successful first trials for each mode and (**b**) multiple comparison test results for the effects of trial number and interface mode on score

To investigate the intuitiveness of each of the interaction metaphors, the number of times that participants successfully complete the balancing task in the first trial is recorded for each interface mode. Figure 5.6a shows that the tilt mode allows approximately half the participants to successfully balance the ball on the beam with no prior experience using the interface mode. An analysis of variance (ANOVA) test on the collected data reveals that the effect of the interface mode used on the average scores of the participants has a p-value of $p << 10^{-5}$, much less than the critical value of 0.05 for a 95 % confidence level. However, the effect of the trial number (first trial, second trial, or third trial) also has a significant effect on the average scores of the participants, with a p-value of 3.3×10^{-3}. No interaction between these two effects is found. To further investigate how the interface modes and trial numbers yield significant differences in score, a multiple comparison test is performed. The results of the multiple comparison test are shown in Fig. 5.6b. Note that the bars displayed in the figure are not error bars calculated from the variance in the data, but rather equal width intervals calculated using a modified Tukey–Kramer procedure [19] and used by MATLAB for purposes of statistical comparisons. If any two intervals are disjoint, then there is a significant difference between the two associated data points. Although participants perform better on average after successive trials using each interface mode (except for the button mode), the tilt mode is the only one of the four interface modes for which participants exhibit a significant increase in skill between their first and last trials. Participants are more likely to complete the task successfully on average with the tilt mode on their first attempt than with any trial using any other interface mode. Finally, the number of participants who complete the balancing task on their second trial with the tilt interface is statistically greater than the number of successful participants in any trial with any other interface.

To further investigate the performance of the participants, the RMSD was measured for each successfully completed trial. The results of an ANOVA test indicate that the interface mode used has a significant effect on the average RMSD

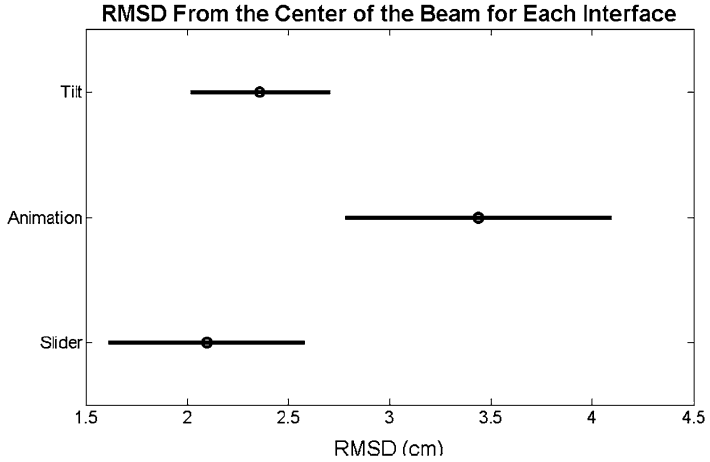

Fig. 5.7 Root mean squared deviation (RMSD) from the center of the beam for each interface

value associated with a successful trial ($p = 1.73 \times 10^{-2}$). A multiple comparison test performed to investigate this effect further yields the results shown in Fig. 5.7. These results indicate that participants using the tilt mode and slider mode are able to balance the ball significantly closer to the center of the beam than the participants using the animation mode.

5.3.2 Qualitative Results

A survey with statements motivated by [35] is given to participants after they complete the interaction exercise. The survey asks participants to agree or disagree with the statements for each of the interface modes according to a 5-point Likert scale, with 1 indicating strong disagreement and 5 indicating strong agreement. The statements consist of the following:

1. Understanding how the interface works is difficult.
2. Remembering how the interface works is difficult.
3. Using the interface is difficult.
4. The interface always performs the intended action.
5. Overall, using the interface is very satisfying.
6. I will be interested to use this interface to control other physical systems.

The participant responses are tallied and used with their suggestions and comments to compare the interaction metaphors in terms of user experience. The participants report having a positive experience using the smartphone interface. Nearly all participants respond that it is neither difficult to understand nor difficult to remember how to use any of the four interface modes. However, most participants agree that the button mode is more difficult to understand at first and the most difficult to use, that it does not always perform the participant's intended action, that it is not very satisfying, and that they would not be interested in using this interface mode again. These responses are expected, since no participant successfully completed the balancing task with the button interface. This is because the binary nature of buttons does not present a clear intuitive mapping for operating a rotating motor whose speed and direction must continuously change. This problem is solved in the gaming industry with the introduction of interface elements like knobs and joysticks. Although a better button-based mapping than the one explored in this study may exist, the pressing of buttons is severely limiting and does not bear any spatial relationship to the problem, and so it can be safely concluded that buttons cannot provide a suitable interaction metaphor for this system. In contrast to the button mode, most participants report that the tilt mode is the easiest and most intuitive to use. This is because the tilt mode effectively exploits the accelerometer onboard the smartphone to provide the user with an intuitive metaphor based on a spatial analogy which is immediately easy to understand.

Participants are also asked to select their preferred interface mode for conducting the experiment. Figure 5.8 shows the results of the responses from the participants, with 82 % of the participants preferring to use the tilt mode. Thus, in combination with the quantitative results, it is seen that the tilt mode provides the most effective, intuitive, and preferred metaphor for interacting with the ball and beam test-bed to perform the balancing task.

5.3.3 Discussion of Cyber-Physical Effects

During the interaction, participants can choose to look at the animation of the test-bed on the smartphone screen or the actual test-bed. Because of the reaction time required by the balancing task, it is generally difficult for participants to switch

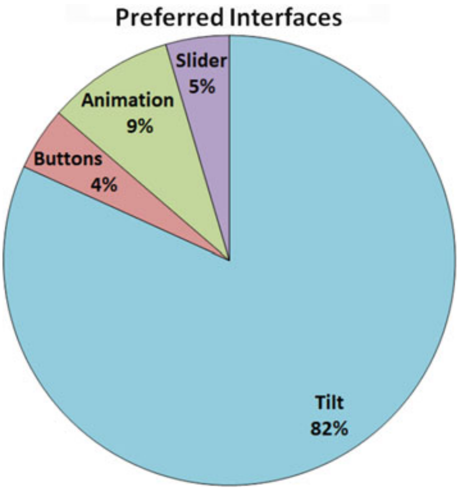

Fig. 5.8 The percentage of times each interface is chosen as the preferred interface by participants

their attention between the two. Participants report that they can sense a small time delay between the response of the animated system and the physical test-bed. There will always be delays in a system such as the one presented in this chapter due to latency inherent in wireless network communication; however, under the influence of gravity, the dynamics of the ball and beam system are especially fast in comparison to the communication delay and update time of the mobile interface. This is not a significant issue in many practical applications involving systems with relatively slower dynamics and tasks which do not impose strict timing constraints [13]. In fact, noticeable delays in the animation disappear as users bring the ball to a more stable state where it can slowly move near the center of the beam. Nonetheless, most participants opt to look at the actual test-bed and use their own visual feedback as opposed to the animation. Since studies with robots show that systems that are directly teleoperated have the potential for increased performance when conducting complex tasks [4], future research must investigate how complex tasks may be performed at large distances, in which direct teleoperation is no longer accompanied by direct visibility and users must rely on only the interface. While using their own visual feedback from the test-bed, some participants report that the slider and animation interfaces, which both involve dragging fingers on the touchscreen, give essentially the same interactive experience. The tilt interface is the only interface that is unaffected by this issue, since in this case participants solely focus their attention on the actual test-bed while naturally tilting the smartphone to command the beam orientation.

Since network latency can degrade the performance and usability of the system, during the user trials efforts were made to keep latency to a minimum. Specifically, all 252 trials were conducted on the same day on a local network that was exclusively dedicated to conduct this study. Next, the size of the network packets was kept to a minimum, the distance between the mobile device and the router was held relatively constant, and algorithms [28] that degrade real-time response of the system were disabled. These strategies allowed the fluctuations in network latency to be neglected across trials and users did not experience any effects of latency fluctuations during trials.

6 Conclusions

Smart mobile devices, such as smartphones and tablets, have the potential to serve as components that may be integrated into human-in-the-loop cyber-physical systems. In these systems, they can provide users with access to immersive and intuitive interactions with complex physical processes. This chapter demonstrated that to achieve effective interactions, in terms of stability, performance, and quality of user experience, harsh demands are imposed on the mobile device that test the limits of its SSCC power. An example system was presented that illustrated how the hardware and software of mobile devices can be blended in the design of intuitive interaction metaphors that are oftentimes based on some physical analogy between user behavior and the nature of the system. Although the integration of mobile devices using the architecture described in this chapter could serve to enhance people's interactions with physical systems, future work must further explore how to resolve issues that arise from cyber-physical effects associated with the interplay of computation, communication, and dynamics of the physical process.

Acknowledgements This work is supported in part by the National Science Foundation awards RET Site EEC-1132482, GK-12 Fellows DGE: 0741714, and DRK-12 DRL: 1417769, and NY Space Grant Consortium grant 48240-7887.

Appendix: Beam Position Control Modeling and Design

To control the angular position of the beam from the laboratory station, the test-bed is modeled as a combination of an electrical and a mechanical subsystem. Figure 5.9 shows a schematic representation of the test-bed, which includes an armature-controlled DC-motor, a gearbox, and the beam [7, 20]. The governing equations for the rotation of the beam under the influence of an applied voltage to the DC-motor can be approximated as a first-order transfer function from the armature voltage $V_{\mathrm{a}}(s)$ to the angular velocity of the beam, $\omega_\ell(s)$, as shown below

$$\frac{\omega_\ell(s)}{V_{\mathrm{a}}(s)} = \frac{K}{\tau s + 1}, \tag{5.1}$$

where K is the dc-gain and τ is the mechanical time-constant of the test-bed dynamics. Next, the parameters K and τ are experimentally identified from the step response of the system [20] and found to be $K = 1.58\,\mathrm{rad/s/V}$ and $\tau = 0.068\,\mathrm{s}$. The first-order transfer function is combined with an integrator to obtain the angular position and converted to the following state-space model:

$$\begin{gathered} \dot{x}(t) = Ax(t) + Bu(t), \quad u(t) = V_{\mathrm{a}}(t), \\ A \triangleq \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 0 \\ \frac{K}{\tau} \end{bmatrix}, \quad x \triangleq \begin{bmatrix} \theta_\ell \\ \omega_\ell \end{bmatrix}. \end{gathered} \tag{5.2}$$

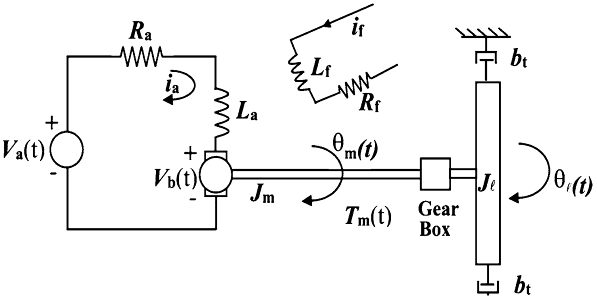

Fig. 5.9 The model of the motorized beam

Using the second-order state-space model of the motorized beam test-bed (5.2), a linear quadratic regulator [6] is designed by selecting the control gain K_c such that the full-state feedback control law $u(t) = V_a(t) = -K_c x(t)$ minimizes the quadratic performance

$$\begin{aligned} J(K_c) &= \int_0^\infty \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt, \\ Q &= \begin{bmatrix} 500 & 0 \\ 0 & 5 \end{bmatrix}, \quad R = 15. \end{aligned} \tag{5.3}$$

The solution for the control gain is found to be $K_c = \begin{bmatrix} 5.7735 & 0.4765 \end{bmatrix}$. Note that a Simulink model is designed to implement the feedback controller on the desktop computer. Specifically, the control signal $u(t)$ is computed using sensor measurements and user commands received from the smartphone. To avoid the wires on the test-bed from getting entangled, in all cases, the feedback controller implementation constrains the commanded position of the beam angle to be within ±20 °.

References

1. Abaid N et al (2013) Controlling a robotic fish with a smart phone. Mechatronics 23(5):491–496
2. Conti M et al (2012) Looking ahead in pervasive computing: challenges and opportunities in the era of cyber–physical convergence. Pervasive Mob Comput 8(1):2–21
3. Correa A et al (2010) Multimodal interaction with an autonomous forklift. In Proceedings of ACM/IEEE conference on human-robot interaction, pp 243–250
4. Crandall JW, Goodrich MA (2002) Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation. Proc IEEE/RSJ international conference on intelligent robots and systems 2:1290–1295
5. Daponte P et al (2013) State of the art and future developments of measurement applications on smartphones. Measurement 46(9):3291–3307
6. Dorato P, Abdallah C, Cerone, V (1995) Linear quadratic control: an introduction. Prentice Hall, Upper Saddle River, NJ
7. Dorf RC, Bishop RH (2008) Modern control systems. Prentice Hall, Upper Saddle River, NJ
8. Evans AW et al (2012) Control solutions for robots using android and iOS devices. In: Proceedings of SPIE 8387, unmanned systems technology XIV
9. Fisher CJ (2010) Using an accelerometer for inclination sensing. AN-1057, Application note, Analog Devices
10. Fong T (2001) Collaborative control: a robot-centric model for vehicle teleoperation. The Robotics Institute, Carnegie Mellon University
11. Fong T, Thorpe C, Glass B (2003) PdaDriver: a handheld system for remote driving. In Proceedings of IEEE international conference on advances in robotics, Paper No. LSRO2-CONF-2003-004
12. Francese R, Passero I, Tortora G (2012) Wiimote and Kinect: gestural user interfaces add a natural third dimension to HCI. In: Proceedings of international working conference on advanced visual interfaces, pp 116–123
13. Frank JA, Kapila V (2014) Development of mobile interfaces to interact with automatic control experiments [Focus on Education]. IEEE Control Syst 34(5):78–98
14. Frank JA, Kapila V (2014) Performing difficult teleoperation tasks using dominant metaphors of interaction. In: Proceedings of ASME biennial conference on engineering systems design and analysis, Paper No. ESDA2014-20133, p V003T15A007

15. Frank JA, Gomez JAD, Kapila V (2015) Using tablets in the vision-based control of a ball and beam test-bed. In: Proceedings of international conference on informatics in control, automation, and robotics, pp 92–102
16. Frank J, Sahasrabudhe Y, Kapila V (2015) An augmented reality approach for reliable autonomous path navigation of mobile robots. In: Proceedings of Indian control conference, Paper No. WeBT2.2
17. Gerling KM, Schulte FP, Masuch M (2011) Designing and evaluating digital games for frail elderly persons. In: Proceedings of international conferences on advances in computer entertainment technology, p 62
18. Guo C, Sharlin E (2008) Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. In Proceedings of SIGCHI conference on human factors in computing systems, pp 121–130
19. Hochberg Y, Tamhane AC (1987) Multiple comparison procedures. Wiley, New York, pp 96–98
20. Kapila V, Lee S (2008) Automatic control laboratory experiment manual, 2nd edn. Polytechnic Institute of New York University
21. Khaitan SK, McCalley JD (2014) Design techniques and applications of cyberphysical systems: a survey. IEEE Syst J 9(2):350–365
22. Khan WZ et al (2013) Mobile phone sensing systems: a survey. IEEE Commun Surv Tutorials 15(1):402–427
23. Krofitsch C et al (2013) Smartphone driven control of robots for education and research. In: Proceedings of IEEE international conference on robotics, biomimetics, and intelligent computational systems, pp 148–154
24. Kuparinen L, Silvennoinen J, Isomäki H (2013) Introducing usability heuristics for mobile map applications. In: Proceedings of international cartographic conference
25. Lane N et al (2010) A survey of mobile phone sensing. IEEE Commun Mag 48(9):140–150
26. Lopez DA, Frank JA, Kapila V (2014) Comparing interface elements on a tablet for intuitive teleoperation of a mobile manipulator. In: Proceedings of ASME biennial conference on engineering systems design and analysis, Paper No. ESDA2014-20126, p V003T15A005
27. Milgram P (1993) Applications of augmented reality for human-robot communication. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems, pp 1467–1472
28. Mogul JC, Minshall G (2001) Rethinking the TCP Nagle algorithm. ACM SIGCOMM Comput Commun Rev 31(1):6–20
29. Nielsen (2010) The state of mobile apps. The Nielsen Company. http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2010-Reports/Nielsen-State-of-Mobile-Apps-Whitepaper.pdf. Cited 30 Jan 2014
30. Nielsen CW, Bruemmer DJ (2007) Hiding the system from the user: moving from complex mental models to elegant metaphors. In: Proceedings of IEEE international symposium robot and human interactive communication. Engineering systems design and analysis, pp 756–761
31. Nielsen J, Budiu R (2012) Mobile Usability. New Riders, Berkeley
32. Schirner G et al (2013) The future of human-in-the-loop cyber-physical systems. Computer 46(1):36–45.
33. Sheridan TB (1992) Telerobotics, automation, and human supervisory control. MIT Press, Cambridge
34. Walker AM, Miller DP (2012) Tele-operated robot control using attitude aware smartphones. In: Proceedings of ACM/IEEE conference on human-robot interaction, pp 269–270
35. Walters ML et al (2005) Practical and methodological challenges in designing and conducting human-robot interaction studies. Proc AISB 5:110–119
36. Wasserman AI (2010) Software engineering issues for mobile application development. In: Proceedings of FSE/SDP workshop future of software engineering research, pp 397–400
37. White J et al (2010) R&D challenges and solutions for mobile cyber-physical applications and supporting internet services. J Internet Serv Appl 1(1):45–56
38. Williams DH (2003) PDA robotics: using your personal digital assistant to control your robot. McGraw-Hill, New York