Dumb Robots, Smart Phones: a Case Study of Music Listening Companionship

Guy Hoffman

Abstract—Combining high-performance, sensor-rich mobile devices with simple, low-cost robotic platforms could accelerate the adoption of personal robotics in real-world environments.

We present a case study of this “dumb robot, smart phone” paradigm: a robotic speaker dock and music listening companion. The robot is designed to enhance a human’s listening experience by providing social presence and embodied musical performance. In its initial application, it generates genre-specific, beat-synchronized gestures based on the song’s genre, and maintains eye-contact with the user.

All of the robot’s computation, sensing, and high-level motion control is performed on a smartphone, with the rest of the robot’s parts handling mechanics and actuator bridging.

I. INTRODUCTION

Human-robot interaction (HRI) has advanced significantly over the past decade. Still, most interactive robots are found in laboratories, with personal robots “in the wild”—in people’s homes, offices, and classrooms—not being commonplace.

At the same time, personal computing is shifting towards handheld devices characterized by many features of interest to HRI: (a) high-end reliable sensors previously unavailable to lay users—cameras, microphones, GPS receivers, accelerometers, gyroscopes, magnetometers, light, and touch sensors; (b) high processing power, comparable to recent notebook computers; (c) a growing number of advanced software libraries, including signal processing modules; (d) continuous internet connectivity through wireless and mobile data networks; and (e) high mobility, due to small weight, small size, and battery power. In addition, the two most widespread smartphone operating systems to date specify peripheral data interchange standards to external electronics.

Combining these devices with simple, low-cost robotic platforms could help accelerate the adoption of personal robots in real-world environments, making use of the advanced hardware and software already in the homes, offices, and classrooms of many users. We call this approach “dumb robots, smart phones” (DRSP).

According to this paradigm, all computation, most sensing, and all high-level motion planning and control are performed on the mobile device. The rest of the robot’s parts deal only with mechanics, per-need additional sensors, and low-level actuator control.

The continuous network connectivity of mobile devices opens additional possibilities: (a) remote monitoring of user interaction; (b) remote updating of robot software; and (c) the use of server-based (“cloud”) computation, offloading high-computational demand processes to network computing, a notion already explored in larger service robots [1].

In addition, we suggest that “sharing” a personal object such as a mobile device with a robot could afford emotional bonding. It can also support joint-attention and common-ground interaction between human and robot, focused on the shared device, as well as on the information contained in it.

This paper presents a case study of the DRSP paradigm, in the form of a new robot, Travis, a robotic speaker dock and music listening companion (Fig. 1). Travis is a musical entertainment robot connected to an Android smartphone, and serves both as an amplified speaker dock, and a socially expressive robot. Travis is designed to enhance a human’s music listening experience by providing social presence and audience companionship, as well as by embodying the music played on the device as a performance. We developed Travis as a research platform to examine human-robot interaction as it relates to media consumption, nonverbal behavior, timing, and physical presence. In its proof-of-concept application, the robot performs genre- and segment-specific beat-synchronized gestures to accompany the music played on the device, maintains eye-contact with the user, and uses gesturing for common ground.

II. BACKGROUND

A. Mobile-device based robotics

Despite increasing capabilities in sensing, computation, and connectivity, there has been little use to date of “smart” mobile devices in HRI research. One exception is mebot...
C. Musical Robots and Physical Gestures

Travis also builds on the notion of musical robots. Robotic musicianship extends other kinds of computer music by adding a physical aspect to computer-generated and interactive musical systems [13]. It provides humans with physical cues that are essential to musical interactions. These cues help players anticipate and coordinate their playing. But, importantly, they also create a more engaging experience for the audience by adding a visual element to the sound.

Virtually all robotic musicianship research deals with music production and improvisation [14], [15], with little research on the effect of musical robots for audiences, or the effect or performance in music listening. In human music listening, it has been shown that adding a video channel to a music performance alters audience perception in terms of the affective interpretation of sound features [16]. Musical robots, too, have been shown to positively affect audience appreciation of joint improvisation [15]. This finding, however, was not separated from the other musician’s ability to see the robot’s gestures as it was playing.

Travis is intended to serve as a research platform to isolate and identify the effects of the performative aspect of robotic musicianship on human’s music listening.

III. APPEARANCE DESIGN

The robot’s physical appearance was designed with a number of guidelines in mind: first, the robot’s main application is to deliver music, and to move expressively to the music. Its morphology therefore emphasizes audio amplification, and supports expressive movement to musical content. The speakers feature prominently and explicitly in the robot’s design. Moreover, by positioning the speakers in place of the eyes, the design evokes a connection between the input and output aspects of musical performance and enjoyment. Travis’s head and limb DoFs are placed and shaped for prominent musical gestures.

Second, the robot needs to be capable of basic nonverbal communicative behavior, such as turn-taking, attention, and affect display. The robot’s head, when placed on a desk, is roughly in line with a person’s head when they are seated in front of it.

Finally, the robot’s appearance should evoke social presence and empathy with the human user. Its body is sized and shaped to evoke a pet-like relation, with size comparable to a small animal, and a generally organic, but not humanoid form.

When designing a smartphone-based robot, an inevitable design decision is the integration of the mobile device within the overall morphology of the robot. Past projects have opted to integrate the device as either the head or the face of the robot. Mebot uses the device to display a remote operator’s face on a pan-tilt neck [2]. Other projects [5], [6] have converted the mobile device’s screen into an animated face inside the robot’s head, an approach similar to that taken by the designers of the Tofu robot [17].

In contrast, we have decided to not make the mobile device part of the robot’s body, but instead to create the appearance that the robot “holds” the device, and is connected to it through a headphone cable running to its head. This is intended to create a sense of identification (“like-me”) and...
empathy with the robot, as Travis relates to the device similarly to the way a human would: holding it and listening to the music through its headphone cable. Moreover, this setup allows for the device to serve as an object of common ground \cite{18} and joint attention \cite{19} between the human and the robot, setting the stage for nonverbal dialog. The robot can turn the phone’s front screen towards its head and towards the human discussion partner (Fig. 2(a)). In our current application, for example, we use a gaze gesture (Fig. 1) as a nonverbal grounding acknowledgment that the device was correctly docked.

Overall, we used an iterative industrial / animation / mechanical design process, similar to the one used for the design of our previous robots, AUR \cite{20} and Shimon \cite{15}. This process includes separate design stages that take into account the appearance (industrial design), motion expressivity (animation), and physical constraints (mechanical design) of the robot. Initial concept sketches (Fig. 2) lead to a rough 3D model transferred into an animation program. The animation stage consists of generating numerous test animations with varying DoF placements to explore the robot’s expressivity in terms of its physical structure. This stage sets final DoF number and placement. The result is then resolved in terms of the physical constraints and dynamic properties of the motors used.

### IV. SYSTEM OVERVIEW

The resulting design consists of a five degree-of-freedom robot with one DoF driving the device-holding hand pan, one driving the foot tap, and three degrees of freedom in the neck, set up as a tilt-pan-tilt chain. Each DoF is controlled via direct-drive using a Robotis Dynamixel MX-28 servo motor. The motors are daisy-chained through the servos’ TTL network. The robot has two speakers, acting as a stereo pair, in the sides of its head, and one subwoofer speaker pointing downwards in the base. In addition, the robot contains an ADK/Arduino control board, and a digital amplifier with an audio crossover circuit (Fig. 3).

As per the DRSP paradigm, the robot’s system can be divided into two parts (Fig. 4): all software, including high-level motor control is performed on the smartphone, in the form of a single mobile application. This application communicates over USB using the Android Debug Bridge (ADB) protocol with the ADK board. The device also transmits analog audio to the amplifier in the robot’s body.

### V. EXPRESSIVE MUSICAL GESTURES

In its initial application, Travis plays songs from the mobile device’s music library and responds to the played songs by generating dance moves based on the song’s beat, segment, and genre. We assume the songs have been accurately split into segments (e.g. “intro”, “verse”, “chorus”) and beats, as well as classified into genres (“rock”, “jazz”, “hip-hop”, etc).

The segmentation and classification of songs is beyond the scope of this paper, as there is a large body of work concerned with methods to automatically track beats in musical audio (e.g. \cite{21}, \cite{22}), as well as for splitting musical audio into segments (for a review, see: \cite{23}). More recently, network-based services offer identification and classification of musical audio based on short audio samples. Some of
these services provide beat and segmentation information, as well [24].

We therefore focus on the expressive gesture and animation system given a song’s accurate genre and beat segmentation. Fig. 5 shows an overview of the robot’s system software.

![Gesture + Animation System](image)

**Fig. 5: Travis software diagram.**

The building blocks of the expressive behavior system are genre- and segment-specific Behaviors. These are modeled as movement responses to real-time song beats.

The **Behavior Controller** receives the current song’s metadata—its genre, tempo, and duration—from the device’s media player, and manages the launching and aborting of the robot’s various Behaviors. When no song is playing, a default “breathing” Behavior indicates that the robot is active and awaiting input from the user (see: [25]).

As the song is playing, a **Beat and Segment Tracker** module follows the progress of the song by the Media Player, and triggers callback events to the behavior subsystems of the robot. In case of a segment change, the Tracker calls back the Behavior Controller, causing it to select the next appropriate Behavior based on the genre and segment. For beats, we have currently implemented two kinds of Tracker modules, one fixed-interval module that detects the first beat, and then triggers beats at fixed intervals. This is usually appropriate for electronically generated music files. The second module uses variable intervals read from a beat data file generated by prior beat analysis.

In case of a beat trigger, the Tracker calls the currently running Behavior to execute one of two beat responses: (a) a repetitive beat gesture involving one or more DoFs; or (b) a probabilistic adjustment gesture, adding variability to the repetitive motion. Each motion is then split by DoF and sent to the Trajectory Interpolator associated with the DoF, as described in Section V-B.

**A. Responding to a beat**

Travis responds to a beat by doing a genre-appropriate movement, usually a repetitive back-and-forth gesture (e.g. “head banging”, “foot tapping”, etc). For this gesture to appear on beat, the robot has to perform the direction change very close to the audible occurrence of the beat, as we have found human observers extremely sensitive to the timing of the trajectory reversal. This planning challenge is exacerbated when beats are not at perfectly regular intervals.

We address this challenge with an **overshoot and interrupt** approach, scheduling each segment of the repetitive movement for a longer time period than expected, and ending the motion not with a zero velocity, but with a slow continued trajectory to a point beyond the target. The following beat then interrupts the outgoing trajectory on sync with the returning trajectory command. Since the exact spatial position of the beat event is not crucial, “overshoot-and-interrupt” allows for a continuous and on-beat repetitive gesture. The robot seemingly reaches the end of its motion precisely on beat, simply by reversing course at that moment.\(^1\)

**B. Smoothing the motion trajectory**

Within each gesture segment, we aim to achieve lifelike, expressive motion. Traditional and computer animation uses trajectory edge-damping to achieve less mechanical seeming movement, a technique called ease-in and ease-out [26]. While easily accomplished through acceleration-limited motor control, many lower-end servo motors, such as the ones used in the design of Travis, specify movement only in terms of goal position and velocity. In addition, to optimize bandwidth on the servo’s half duplex architecture, we also rely on dead-reckoning, without polling the motors for their accurate position.

To simulate ease-in/ease-out given these constraints, we use a high-frequency interpolation system, inspired by the animation arbitration system used in [27], and similar to the one used in a previous robot, Shimon [15]. A **Trajectory Interpolator** per DoF receives target positions and maximal velocities from the Behavior layer, and renders the motion through a high-frequency (50Hz) interpolator. The closer the motion is to the edge of the movement, the slower the commanded velocity of the motor. Periodic velocity \(v'\) is expressed as a positive fraction of goal velocity \(v\):

\[
v' = v \times \left(2 \times (1 - \frac{|t - d/2|}{d}) - 1\right)
\]

where \(t\) is the time that passed since the start of the movement and \(d\) is the planned duration of the movement.

An opportune side-effect of this approach is that the duration compensation from the original linear motion trajectory causes the movement to take slightly longer than the single or half beat of the gesture. This enables the use of the overshoot-and-interrupt approach described above, resulting in precise beat timing. The combination of both methods results in continuous, life-like, beat-synchronized gestures.

**VI. EYE-CONTACT**

Gaze behavior is central to interaction both between humans [28] and between humans and robots [29]. Travis makes eye-contact by using the built-in camera of the mobile device to capture the scene in front of it. We then make use of existing face detection software on the phone to track and follow the user’s head.\(^2\)

\(^1\)Thanks to Marek Michaelowski for pointing out this last insight.
Fig. 6: Active perception tracking with the head following the camera-holding hand, compensating for parallax.

Our head tracking follows an active perception approach [30], [31]. Since the phone is mounted on a pan DoF, linear compensation feedback will keep the head centered in the camera view. Given a high enough face detection frame rate, and continuous user motion, we move the device-holding hand according to

\[ p' = p + \lambda(x - \frac{w}{2}) \]

with \( p \) being the current motor position, \( x \) being the face detection center of mass, \( w \) the image width, and \( \lambda \) the tracking factor. A higher value for \( \lambda \) results in more responsive, but also more jittery, tracking.

As the mobile device, and thus the camera, is coupled to the robot’s hand, gaze behavior requires an additional transformation of the hand rotation to the head pan coordinates. Coupling the neck pan DoF angle \( \theta' \) to the active perception result angle \( \theta \), the robot compensates for parallax induced by the disparity \( d \) between the two DoF centers (Fig 6). \( h \) is the estimated frontal distance of the human’s head:

\[ \theta' = \arctan\left(\tan\theta - \frac{d}{h}\right) \]

We are currently able to smoothly track a human head with 40 motion commands and 16 detections per second, using the built-in face tracking of a Samsung Galaxy Nexus smartphone running Android 4.0.2.

VII. USE OF SMARTPHONE INFRASTRUCTURE

The design of a DRSP robot such as Travis could serve as a model for the wider adoption of personal robotics, as smartphones become more prevalent, and increasingly equipped with sensing, computation, and interaction capabilities. In this case study, the functionality of a commercially available mobile device kept the robotic platform constrained to a simple bridge controller and consumer-level servo motors without position feedback. Still, it resulted in expressive robot behavior, comparable to that achieved in the past with specialized motors, hardware, and software libraries.

This section describes our current use, and guidelines for future utilization, of smartphone infrastructure for personal robotics.

A. Current

In the music response application, all computation was processed on the mobile device, relying heavily on existing OS software. In particular, we used the phone’s media player and playhead tracking API, as well as the built-in audio hardware to connect to our speaker system. We also used an existing accessory protocol to command the motors through the phone’s USB port.

The device’s high-resolution micro-camera in combination with the operating system’s fast face detection API enabled active vision tracking using a single pan DoF. This resulted in smooth gaze behavior which, until recently, was reserved for research-grade equipment and software libraries.

In addition, we are currently using the device’s network connection for human subject experiments, and have explored the use of the built-in microphone for both music information retrieval and voice commands. The latter also relies on existing network-processed speech-to-text software increasingly available on commercial smartphones and other mobile devices. These modules have not been included in the application described in this paper.

B. Future

Additional sensors and software libraries on smartphones are applicable to personal robotics. For example, available GPS tracking subsystems with mapping and reverse geocoding could be beneficial to mobile personal robots. Robots could use the device’s accelerometer, gyroscope, and magnetometer to infer their own orientation and acceleration. This could provide for safety related capabilities, such as drop and bump detection. It could also support interaction scenarios in which a robot is held by the human, as has been explored in the realm of child and elder care [32], [33].

A smartphone’s network connectivity allows for communication between robots, and between robots and their users’ personal computers. In addition, as many processing-intensive computational task are transferred to a server-based model (“cloud computing”), robots using smart phones as their computational core could make use of such services to further enhance their processing capabilities [1]. We are currently exploring the use of server-side song detection, beat analysis, and genre classification for our musical robot application.

Smartphones are also highly personalized, and can identify their owners, leading to readily customized robotic hardware. Different users in the same usage space (e.g. home, office, nursing home, classroom) could use a single robot hardware which “remembers” their preferences, history, behavior, and dispositions, simply by running their own version of the robot software. This could aid affective bonding with the robot.

Finally, the possibility to remotely log behavior, update and add software to smartphone devices, enables continuous expandability of the robot’s capabilities. New versions of mobile devices with enhanced sensing and computational capabilities could also upgrade the robot without replacing the mechanical hardware of the machine.
A “dumb robot, smart phone” approach to personal robotics has significant potential to accelerate the adoption of robots in real world environments, such as in homes, offices, and schools. This is for a number of reasons:

First, by making use of sensors and processors available on mobile devices, robotic hardware complexity and cost, for both developers and consumers, can be reduced to a fraction.

Second, advances in mobile OS, third-party, and cloud software greatly reduces development time. In our case study we used camera sampling, face recognition, music playing and tracking, and speech-to-text from existing smartphone libraries. In other work, we use music analysis libraries and the robot’s network connectivity for research studies.

Finally, sharing a personal object such as a smartphone with a robot fosters common-ground based human-robot interaction, potentially increasing affective bonding and empathy. We therefore support keeping the smartphone visible and modeling it as an accessory for the robot.

In this paper we explored a case study of these notions realized in a new research robot. Travis is a robotic speaker dock and listening companion, designed to enhance the human listening experience by providing social presence and embodied musical performance. In this application, the robot moves to the beat, keeps eye contact with the user, and uses gestures for common-ground. Additional research into the relationship between media consumption, timing, nonverbal behavior, and physical embodiment is currently underway.

REFERENCES


