The VoiceBot: A Voice Controlled Robot Arm

Brandi House, Jonathan Malkin, Jeff Bilmes
Department of Electrical Engineering, University of Washington
{bhouse,jsm,bilmes}@ee.washington.edu

ABSTRACT
We present a system whereby the human voice may specify continuous control signals to manipulate a simulated 2D robotic arm and a real 3D robotic arm. Our goal is to move towards making accessible the manipulation of everyday objects to individuals with motor impairments. Using our system, we performed several studies using control style variants for both the 2D and 3D arms. Results show that it is indeed possible for a user to learn to effectively manipulate real-world objects with a robotic arm using only non-verbal voice as a control mechanism. Our results provide strong evidence that the further development of non-verbal voice-controlled robotics and prosthetic limbs will be successful.

Author Keywords
Voice-based interface, speech recognition, motor impairment, robotics

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User interfaces ——*Voice I/O*; K.4.2 Computer and Society: Social Issues—*Assistive technologies for persons with disabilities*

INTRODUCTION
Individuals with motor impairments such as those with paraplegia, spinal cord injuries, war-time injuries, or amputations rely on others to assist them their in daily activities. Advances in assistive technologies have begun to provide an increase in independence for these individuals, but there is great potential for further technological developments to significantly improve their abilities, independence, and overall quality of life. One of the greatest challenges faced in assistive technology, however, is that control options are extremely limited when the target users have little or no use of their limbs. For example, a mouse is useless to someone without arms. Spoken language and automatic speech recognition (ASR) systems are often considered the natural solution to such problems. Unfortunately, natural speech is limited to discrete commands, falling short especially on steering tasks which require smooth continuous control.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CHI 2009, April 4-9, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-246-7/09/04...$5.00.

The Vocal Joystick (VJ) [1] seeks to use the human voice without the constraints of a natural language. Instead, the user relies on continuous sounds that can vary in pitch, vowel quality, or amplitude to provide control of computer applications and ultimately electro-mechanical devices. The VJ system has been designed as reusable infrastructure available to any application. Existing work on a VJ-driven mouse has demonstrated the suitability of the system to 2D control [3]. In particular, the VJ mouse application enables the user to navigate a windows/icons/mouse/pointer (WIMP) interface using only the voice. Beyond mouse control, VoiceDraw [4] is a drawing program which mixes speech recognition and Vocal Joystick control in a voice-controlled drawing environment. The success of that work, especially the case study with a motor impaired individual, shows the flexibility of the Vocal Joystick engine and its promise for moving beyond simple WIMP-based interaction. The use of non-verbal vocalization for interaction is a growing field, as evidenced by the recent publication of a humming or whistling interface to control a robotic car [15] or for hand-held devices [19].

In this work, we take a step towards vocal controlled robotics by considering the potential of a VJ system to control a computer simulated two-dimensional robotic arm, and then also a real 5 degrees-of-freedom (DOF) three-dimensional hobbyist robotic arm. Our goal is to leverage the human vocal tract's multiple degrees of freedom to provide an inexpensive control method for a robotic arm. As our later results show, users can indeed complete simple tasks with our simulated robotic arm and the real robotic arm using continuous vocal control. These results are particularly promising given the underlying low cost of a VJ-based system.

BACKGROUND
There exist several assistive technologies using novel control methods for manipulating robotic arms. These include physical joysticks, speech-based robotic controllers, and brain-computer interfaces (BCI). Each seeks to increase independence of individuals with disabilities, and each involves a trade-off between expense, invasiveness, and accuracy.

The most widely accepted method for control of robotic limbs is currently a physical joystick. Commercially available robotic arms often include a joystick feature that allows the user to position the gripper or tool at the end of the arm[1], called the

[1]An example is the Lynxmotion Lynx 6 arm at `http://www.lynxmotion.com/Category.aspx?CategoryID=25`

end effector. These controls tend to be inexpensive, non-invasive, and fairly accurate. For individuals with motor impairments, however, these options are inaccessible. An alternative approach combines speech recognition with semi-autonomous robotics. For example, a system called FRIEND uses a robotic arm attached to an electric wheelchair [11]. The entire system is controlled by a speech interface using simple commands. Sensors and control logic incorporated into the arm allow some complex movements with little user input. Unfortunately, control systems of this type often require a structured physical environment or a large number of voice commands to achieve high accuracy as non-autonomous commands seem to control only 1 dimension at a time. Moreover, such a system is not able to specify continuous control. Recent work by Choi [2], however, uses either a head-mounted laser or a touch screen to help give high-level commands to an autonomous robot.

Looking at non-verbal vocal control methods, such as those of Igarashi [6] or Mihara [12], one finds approaches designed primarily for 1-D control, perhaps including some sort of intensity control; the latter relies on images displayed on a screen for positioning, making it unsuitable for real-world tasks such as robotic arm control. Similarly, the humming or whistling approaches mentioned earlier ([19, 15]) do not currently allow simultaneous control of multiple independent degrees of freedom.

The BCI approach is extremely popular in the mainstream media and is widely accepted to be the great promise of hands-free robotic or prosthetic control. Indeed, BCI has shown considerable progress in the past decade. Non-invasive methods use signals from electrodes on the scalp surface [14] and have recently shown improved 2D control [9]. Invasive BCI devices produce the highest quality signals from measurements of motor neurons, but also require placing electrodes inside the scalp or at nerve endings. The BrainGate implant for humans can deliver 2D cursor control of both direction and speed, although robustness can still be a challenge [7]. The most pressing concern for invasive BCI technology is the need for brain surgery, as well as its ability to function as a long-term solution due to scar tissue formation, and of course the high cost.

The Vocal Joystick, by contrast, relies on the enormous flexibility of the human vocal tract to produce a range of discrete and continuous sounds, providing a unique voice-controlled interface. No specialized or invasive hardware is required, only a standard microphone, sound card, and computer are used. The core engine is a portable library that can be used by any application to provide control parameters. Therefore, possible applications for the VJ are virtually unlimited. The VJ system extracts several vocalic parameters (currently loudness, pitch and vowel quality) which are then mapped into control signals and ultimately motion. In previous work, it has been demonstrated that novices using the VJ for mouse control can provide results at least as good, measured by time to acquire a target, as existing word-based cursor control models [3]. In addition, it has been shown that the Vocal Joystick is also suitable for steering tasks, situations where

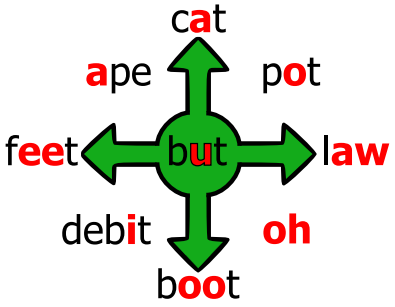

Figure 1. Vowel mapping to cursor movement for mouse control. Words include the nearest vowel in American English.

word-based control models are sorely lacking [8]. While previous work exists for *speech* controlled robotic arms (e.g., in the medical and surgical robotics communities [13]), to our knowledge, our paper demonstrates the first instance of a robot arm being controlled by non-verbal vocalizations.

For mouse control tasks, vowels are mapped into a two-dimensional space, shown in Figure 1, where each vowel determines movement in one direction. Loudness controls cursor velocity with louder sounds corresponding to faster movement. This two-dimensional mapping has been successfully used in drawing applications [4]. Pitch is currently unused in VJ mouse control but is used in the robotic system below. VJ mouse control also uses discrete [ck] and [ch] sounds for buttons — these discrete commands were also used in our robot arm studies, although the underlying VJ software can employ any discrete sounds for this purpose.

INITIAL STUDIES: SIMULATED 2D ARM CONTROL

There are a number of possible control models for a VJ-driven robotic arm. For this first study, we developed a VJ system to simultaneously control three joint angles in two dimensions. We tested three ways of determining joint angles, known as *kinematic models*. In this section, we present the control models, leaving details of the simulated environment for the experimental design section. Note that this section deals with paradigms for control of the arm. A complete exploration of arm control models would also need to examine issues such as what we call "intentional loudness" [10] — estimates of a speaker's intent in an utterance, allowing for discrepancies between intent and the sounds actually produced due to, for instance, physical properties of the vocal tract. Such considerations are beyond the scope of this work, although they may strongly influence results.

Forward Kinematic (FK) Model

Forward kinematics requires the user to control each joint angle explicitly. Such a model is computationally quite simple since the system directly applies the user-supplied input. At the same time, this model may require more cognitive user effort since the user's attention is split between accomplishing the task and the mechanics of realizing that goal. We suspect that an expert user might find this approach appealing while a novice would find it more difficult — lacking true "expert" users (i.e., users who are fluent due to relying on the system as part of their daily routines) means that an answer to this question awaits future research.

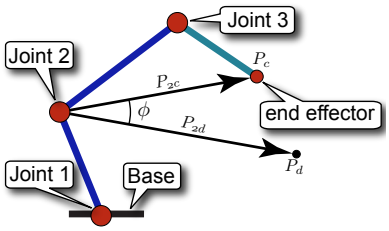

Figure 2. Example of calculation to update joint angle 2 aiming for target point P_d. Joint angles 1 and 3 are fixed for this step, resulting in a 1-D optimization problem.

Inverse Kinematic (IK) Model

In this model, the arm is simply a vehicle used to position the end effector in the appropriate location; the specific joint angles are of little concern to the user as long as the end effector, the typical tool attachment point in robotics, is correctly placed. Through the use of *inverse kinematics*, we can automatically determine joint angles given an end effector's target position. This allows the user to remain focused on the task at hand, but requires additional computation. This approach coincides with the dominant hypothesis for how humans use their own arms for grasping.

There are a number of methods for determining the joint angles, each with advantages and disadvantages. We chose one, cyclic coordinate descent (CCD) [17, 18], due to its relatively simple implementation, rapid convergence, and numerical stability. For this work, we control only joint angles so we can consider only rotational degrees of freedom. CCD works by iteratively minimizing an error function. If we define $\mathbf{P}_c$ as the current end effector position and $\mathbf{P}_d$ as the desired position, the objective is simply to find a joint angle vector θ which minimizes the ℓ_2 norm $E(\theta) = ||\mathbf{P}_d - \mathbf{P}_c||$. For each iteration, the method considers each joint individually, starting from the end and working towards the base. Each joint angle is updated one at a time while the others are held fixed. This means that we need solve only a 1-D optimization problem for each joint.

For joint $i \in \{1, 2, 3\}$ in turn, we rotate the arm at joint i with the goal of moving the end effector P_c closer to the desired position P_d — Figure 2 shows the case where $i = 2$. We rotate joint i by an amount ϕ so that the vector from the joint to the end effector, or P_{ic} becomes parallel with the vector from the joint to the desired location P_{id}. The rotated vector is calculated as $\mathbf{P}'_{ic} = \mathbf{R}(\phi)\mathbf{P}_{ic}$ where $\mathbf{R}(\phi)$ is a rotation matrix parametrized by ϕ. This can be accomplished by maximizing the dot product of the two vectors: $\phi^* = \text{argmax}_\phi \, \mathbf{P}_{id}^T \mathbf{P}'_{ic}$. A new joint angle of $\theta_i^{new} = \theta_i + \eta_i \phi^*$ is then utilized where $\eta_i \in [0, 1]$ is a joint stiffness penalty.

Hybrid Model

Between the FK and IK models is a *hybrid kinematic* system. Since two joints are theoretically sufficient to reach any point in two dimensions, arm segment lengths permitting, the inverse kinematic model is redundant when there are 3 arm segments. We propose a hybrid model in which the first two joints are controlled using inverse kinematics, and the last is controlled directly. This should provide advantages from the other two models, allowing the user to remain more focused on the goal while simultaneously allowing explicit fine-grained adjustments of the last segment.

SIMULATED ARM USER STUDY

Experimental Design

We had two objectives for these experiments. The first was to determine the feasibility of the concept of vocal control of a robotic arm — in attempting these experiments we were making an implicit hypothesis, needing to be tested, that such control is possible. The second objective was to look for evidence supporting our hypothesis that the hybrid model would be preferred over the other two. The hybrid approach, in theory, allows simple arbitrary positioning while retaining an additional degree of freedom for small adjustments.

Each of the three control models were tested in an attempt to determine the fastest method for a simple ball placement task. The testers were required to move a ball along the ground to four sequential locations for each control model. The four locations of the target positions were kept constant for each model, and they were designed such that significant movement of the arm was required to reach each new location. We recorded the completion time for each task. If the ball was pushed out of reach of the arm, it could be reset to a starting location, and we recorded the number of such resets. All models used 3-segment arms with each segment 2/3 the length of the previous one; see Figures 3 and 4 for examples. The ball had to be on the target for at least 1 second to be accepted, ensuring users had to intentionally position the ball accurately. Models were tested in the order presented in the section on Control Models.

Users were allowed to practice as long as they wanted with each control model before the timed trial, but they did not have targets during practice. Four experienced VJ users, all Vocal Joystick project members, were chosen as testers, as these individuals are familiar with the sounds required for Vocal Joystick and could focus their attention on controlling the simulated arm. As a result, we considered this to be a feasibility study that may simply suggest performance trends — our study therefore did not take into account the time required to learn the VJ vowels (see [5] for such work).

For the FK model, we used two vowels to move each of the first and second joint angles, and the time derivative of pitch, determined by a linear frequency scale with pitch changes calculated in Hertz, for the third as shown in Figure 3. We used a simple mapping where 1 pixel of movement from the VJ mouse application corresponded to a 1 degree rotation. Large discontinuities were removed from the pitch derivative before scaling by 0.3, which is the same sensitivity found to be effective for the other movement calculations in this application [10].

All inverse kinematic models specified 2D end effector locations using the VJ mouse control (Figure 1). The hybrid model used pitch to control the last arm segment. Stiffness values for the CCD updates were set to 0.75 for each joint, picked to ensure some movement would propagate to the

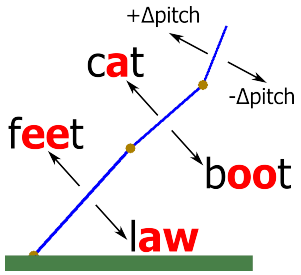

Figure 3. Vowel sounds associated with arm movement for the forward kinematic control model. The underlying image (without labels) is an actual screen-shot from the application.

Model	User 1	User 2	User 3	User 4	User 5*
Forward	602	96	292	138	77
Inverse	216	141	135	230	34
Hybrid	663	175	144	185	85

Table 1. Task completion times (seconds) per user per control model.

base. CCD was allowed to run either until it reached the target, or for a maximum of 50 iterations per step, empirically chosen to allow convergence in almost all cases yet easily achieving real-time performance.

To better distinguish between control models and avoid user confusion, the arm using the forward kinematic model appeared as shown in Figure 3 (without control labels). The inverse kinematic model colored the last arm segment cyan. The hybrid model colored the joint between the last two arm segments cyan – the control point was at the center of the joint. In the latter two models, the user could click a checkbox to show a red dot at the inverse kinematic control point.

Results and Discussion

Time trial results for the experiments appear in Table 1. We first observe that all users were successfully able to complete all tasks, thus demonstrating feasibility of the concept of non-verbal voice-controlled robotics. Users 1, 2 and 3 had experience with the VJ mouse but this application was entirely new to them all. User 4 had experience with an older version of the forward kinematic model, but the inverse and hybrid kinematic models were new. Most users practiced longer with the forward kinematics than with the others, perhaps because they felt they understood 2D positioning well from their VJ mouse experience. This meant they had a less thorough understanding of how the inverse kinematics positioned the arm. User 5 was the primary developer, and results presented here are the best achieved. These times, the results of much practice associated with building and testing the system, are included to show the currently known best case for each method.

User 1 typically used only the last segment to manipulate the ball, which meant taking a large amount of time to position the arm so that the shortest arm segment could reach the ball, even for longer distance moves. As a result, this tester required many more careful position changes than other users. This strategy was unavailable with the inverse kinematics resulting in much faster performance. In all cases, users had

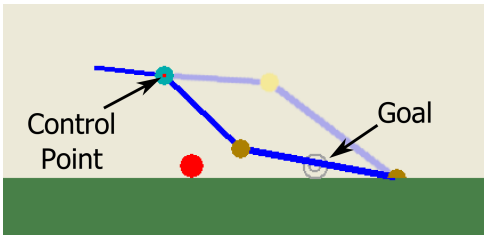

Figure 4. Example of hybrid mode in an undesirable configuration. The faded arm would be a better position since the arm currently prevents the red ball from reaching the goal. Users often had trouble helping CCD reach the faded position while moving the control point via inverse kinematics within the hybrid model.

the most difficulty controlling pitch, likely since the mapping of pitch to movement has received less thorough study than mapping of the other parameters.

The user sample is not unbiased, but preference trends were apparent. Although not always the fastest in these trials, users generally preferred using inverse kinematics (faster for 3/5 of the users, with forward kinematics faster 2/5 of the time). The hybrid mode was strongly the least favorite method for all but user 3, and forward kinematics was in the middle.

The hybrid mode had poor performance because the joint between the first and second arm segments often became "stuck" bending to one side due to the inverse kinematics algorithm. Finding a movement so the algorithm will bend the joint to the other side sometimes took substantial effort. Figure 4 shows how a bad joint angle can prevent the ball from reaching the target. With its extra available joint angle, the pure inverse kinematics model provides a simple way to avoid the problem of the arm bending in an undesirable manner — by briefly moving the arm to the upper portion of its reach, users can help the algorithm find a more suitable position. Forward kinematics avoids the problem entirely.

Since the test subjects were drawn from the VJ development team, we do not make any definitive performance conclusions from these results other than that a 2D voice-controlled robot-arm is feasible to accomplish simple tasks. We are comfortable in declaring, however, a lack of support for our initial hypothesis that the hybrid model would be superior to the other models. In practice, the redundancy of the inverse kinematics mode helped minimize the effect of the CCD algorithm's weaknesses. The practiced results from User 5 suggest that the inverse kinematics model may be faster than the forward kinematics model, but unequal amounts of practice with the two methods and a sample size of one precludes making such a claim with any certainty.

As mentioned above while describing inverse kinematics, we selected CCD for practical concerns — it works reasonably well, and converges fast enough for real-time continual robot arm movement. Other methods (e.g., [18]) may be more akin to the behavior people expect. Overall, we anecdotally found that users, at least at this level of practice, prefer not having to pay attention to all the joint angles explicitly, but movement produced by another algorithm may prove more favorable. Alternatively, it may be possible to re-

fine the joint angle stiffness weights to produce a more desirable effect. Lastly, with much more practice than even User 5, the human user might adapt so that the forward kinematic model becomes second nature, without the user needing to explicitly think about joint angles, similar to how a musician ultimately learns to play the violin without thinking explicitly about each arm position.

ROBOTIC ARM CONTROL: THE VOICEBOT

Having demonstrated the feasibility of the 2D simulated arm, we investigated controlling a simple 3D robotic arm, which we call *The VoiceBot*. This was done by building an interface between the Vocal Joystick engine and the robotic arm controller. We also developed a new inverse kinematic control algorithm to fit the additional degrees of freedom in this arm. To avoid the issues encountered previously with the CCD approach, another method was used to determine the inverse kinematic equations. Via a geometric analysis of the arm, we found deterministic equations for the joint angles when given the 3D gripper position, under a few assumptions. This derivation is discussed in more detail below.

Arm and Hardware Overview

The robotic arm we use in this exploratory research is a small hobbyist device called the Lynx 6 by Lynxmotion. The arm has a total of five degrees of freedom (DOF): shoulder rotation, shoulder bend, elbow bend, wrist rotate, and wrist bend. A simple 2-prong gripper at the end of the arm is used to hold small objects. Figure 5 demonstrates these controllable features. Although the arm has no feedback or sensors, it is still sufficient as a prototype arm for use in this proof-of-concept exploration.

To interface the VJ system to the robot, commands are sent from the computer to the arm through a serial port. A microprocessor controls servomechanisms (servos) at each of the arm joints. Servos can be rotated 180 degrees, specifically 90 degrees in either direction from its center of rotation. A pulse-width modulated signal from the microprocessor controls the rotation of the servo. The microprocessor on the Lynx 6 arm updates the servo position once every 20 ms, so this limits the speed of position updates.

Control Methods

The VoiceBot is currently controlled using two modes: one for gross positioning (called "position mode"), and the other for fine alterations in the orientation of the gripper (called "orientation mode"). The user can switch between these modes by using a [ck] sound. At all times a [ch] sound can be used to open or close the gripper.

Position Mode

In this mode, the entire arm moves according to one of three kinematic algorithms that use pitch and vowels for controlling movement direction, and loudness to control speed.

Forward Kinematics (FK): As mentioned previously, forward kinematics requires the user to explicitly specify each joint angle. Such a model is computationally simple since the joint angle values are set directly by the user, but it may also require considerable cognitive load, since users must mentally calculate the joint angles required to accomplish a specific task. Only three DOF can be controlled at a given time, so in position mode, the two shoulder rotations are each controlled by two vowel sounds, and the elbow joint is rotated by a higher and lower pitch. More specifically, [ae] (as in cat) and [uw] (as in boot) rotate the shoulder, and [iy] (as in feet) and [aw] (as in law) bend the shoulder as seen in Figure 5.

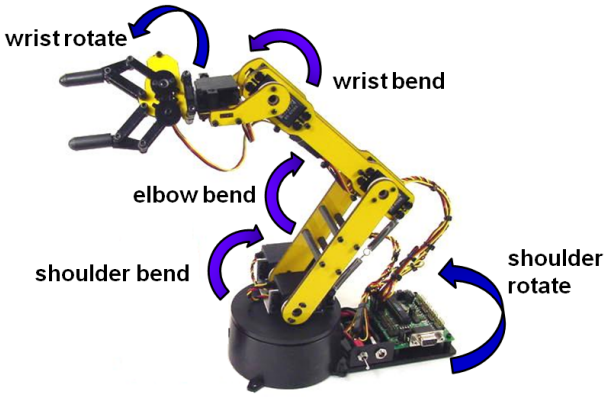

Figure 5. Joint angles on Lynx 6 arm available for control. In forward kinematics, these angles are set explicitly by the user. The inverse kinematic model calculates the angles automatically when the user specifies a gripper position.

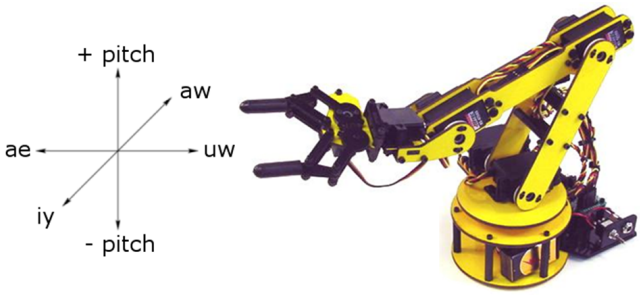

Figure 6. Arm movement under Cartesian space IK control.

Inverse Kinematics - Cartesian (IK): As seen with the 2D simulated arm, inverse kinematics can be useful and possibly more intuitive for some users. Although the inverse kinematic solution is calculated differently for the Lynx arm than the simulated arm, the control from the user's end is similar. This inverse kinematic control method uses closed-form trigonometric equations to automatically calculate the arm's joint angles for each desired gripper position in Cartesian space given by the user. As can be seen in Figure 6, the gripper is positioned by making [uw] and [ae] sounds to move backwards and forwards from the base respectively, and the sounds [iy] and [aw] are used to move left and right. Changes in pitch raise and lower the gripper. The derivation of the equations mapping user specified Cartesian positions to joint angles is given below.

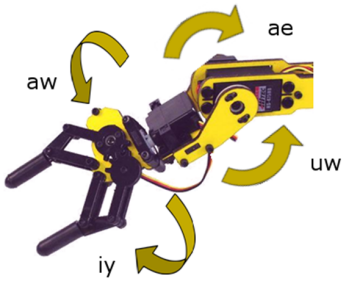

Figure 7. The vowel mapping to control the movement of the gripper in orientation mode.

Inverse Kinematics - Cylindrical: This control method is very similar to the Cartesian IK, and pitch control is identical. However, this Cylindrical method no longer uses the four vowels to move the gripper forward, backward, left and right. Instead, [ae] and [uw] rotate the shoulder, and [iy] and [aw] change the radial distance from the base to the gripper. This method can be seen as a combination of the two previous methods, since it utilizes the shoulder rotation of the FK method and the pitch control of the IK-Cartesian method.

When using either of the inverse kinematic methods, two more options appear for the default movement of the wrist. While the arm is in motion, the wrist can be held at either a constant angle relative to the ground or at a constant angle relative to the arm itself. By keeping the wrist at a constant angle relative to the ground, the user can maintain the orientation of an object while in motion. E.g., a cup of liquid could be held and moved without spilling its contents.

While in position mode, there are also two methods to control how pitch affects the arm, either the delta pitch or the relative pitch method. In either case, our pitch response was again determined by a linear frequency scale with pitch changes calculated in Hertz. With the *delta pitch* method, the amount of arm movement is proportional to the change in the user's pitch during each time frame. For example, with the IK methods, a rising pitch will raise the gripper, and a falling pitch will lower the gripper. The *relative pitch* method relies on three pitch ranges to move the arm: low, medium, and high. After the user has "enrolled" with the Vocal Joystick software [1], an average pitch for the user is determined and used as the center for the medium pitch range. If the pitch of an utterance is fairly close to this medium range, the arm will not change height in IK or rotate the elbow joint in FK. However, if the utterance is significantly higher or lower than the medium range, the arm will raise or lower at a constant rate.

Orientation Mode

The second control mode allows fine control of the gripper instead of producing a gross movement of the arm. This wrist motion is always controlled by the same four vowels, and has far fewer options than the position mode. Figure 7 shows the mapping of the vowels to the wrist movement. The vowels [ae] and [uw] change the wrist bend, and the vowels [iy] and [aw] rotate the wrist. In orientation mode the gripper still opens and closes using the [ch] discrete sound.

The only other option in orientation mode relates to the response of the arm to a change in orientation. The simplest choice is to only have the gripper move by keeping the wrist joint fixed in space while altering the bend and rotation of the gripper (called the "fixed wrist" option). The second option is to keep the tip of the gripper fixed in space and adjust the rest of the arm to accommodate a change in orientation (called the "fixed tip" option).

Geometric Solution for IK Equations of Lynx 6 Arm

Using IK-Cartesian mode, the user specifies the desired target position of the gripper in Cartesian space as (x_d, y_d, z_d) where z_d is the height, and the angle of the gripper relative to ground, α (see Figure 9), is held constant. This constant α allows users to move objects without changing the object's orientation (the holding a cup of liquid scenario). Also, by either keeping α fixed in position mode or keeping the wrist fixed relative to the rest of the arm, the inverse kinematic equations can be solved in closed form as we now show for the case of a fixed α.

The lengths L_1, L_2, L_3 and L_4 correspond to the base height, upper arm length, forearm length and gripper length, respectively, and are constant. The angles $\theta_1, \theta_2, \theta_3$ and θ_4 correspond to shoulder rotation, shoulder bend, elbow bend and wrist bend, respectively. These angles are updated as the specified position in space changes. We solve for the joint angles of the arm, $\theta_{1:4}$ given desired position (x_d, y_d, z_d) and α which are determined from voice control.

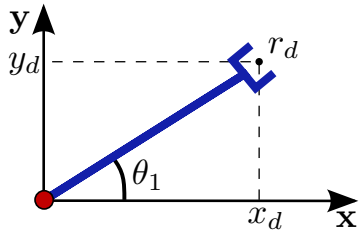

Figure 8. Top view of robot.

From Figure 8, we clearly see that $\theta_1 = \arctan(y_d/x_d)$ and the specified radial distance from the base r_d is related to x_d and y_d by $r_d = \sqrt{x_d^2 + y_d^2}$, $x_d = r_d \cos(\theta_1)$, and $y_d = r_d \sin(\theta_1)$.

Moving now to the planar (i.e., mean sagittal plane of the arm) view in Figure 9, we find a relationship between joint angles θ_2, θ_3 and θ_4 and α as follows:

$$\alpha = \pi - \theta_2 - \theta_3 - \theta_4. \quad (1)$$

Since α is given, we can calculate the radial distance and height of the wrist joint:

$$r_4 = r_d - L_4 \cos(\alpha) \quad (2)$$
$$z_4 = z_d - L_4 \sin(\alpha) \quad (3)$$

We want, finally, to determine θ_2 and θ_3. We first solve for

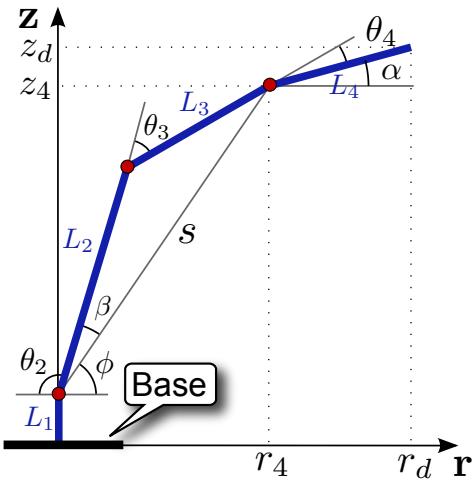

Figure 9. Arm-sagittal planar view of robot. The plane is parallel to the arm segments and includes both the origin and r_d.

β, ϕ, and s (from Figure 9) using the law of cosines as:

$$\beta = \arccos\left(\frac{s^2 + L_2^2 - L_3^2}{2sL_2}\right) \quad (4)$$

$$\phi = \arctan\left(\frac{z_4 - L_1}{r_4}\right) \quad (5)$$

$$s = \sqrt{(z_4 - L_1)^2 + r_4^2} \quad (6)$$

With these intermediate values, we can now find the remaining angle values as:

$$\theta_2 = \pi - \phi - \beta \quad (7)$$

$$\theta_3 = \arccos\left(\frac{s^2 - L_2^2 - L_3^2}{2L_2L_3}\right) \quad (8)$$

$$\theta_4 = \pi - \theta_2 - \theta_3 - \alpha. \quad (9)$$

USER STUDY

Experimental Design

A user study was designed to test two control methods for the arm, FK and IK-Cartesian, with a total of 12 users (8 male, 4 female), all graduate students. Most of these students were familiar with the Vocal Joystick project, although only one had used the VJ previously; none had ever used the VoiceBot. The position mode settings were to use delta pitch for pitch control and a constant angle of the wrist relative to ground with the IK-Cartesian method. In orientation mode, the basic fixed wrist option was selected.

The study used a counterbalanced within-users design in which each user performed two runs of an identical task using each of the control methods. The goal was to move two pieces of wrapped mini-sized candy bars in the order specified to a target as seen in Figures 10 and 11. To simplify vowel learning, we used five vowels: the cardinal directions and the central vowel. The times to complete the tasks were recorded. Before each timed trial, users were also allowed up to 10 minutes for practice, with the option of stopping earlier if they

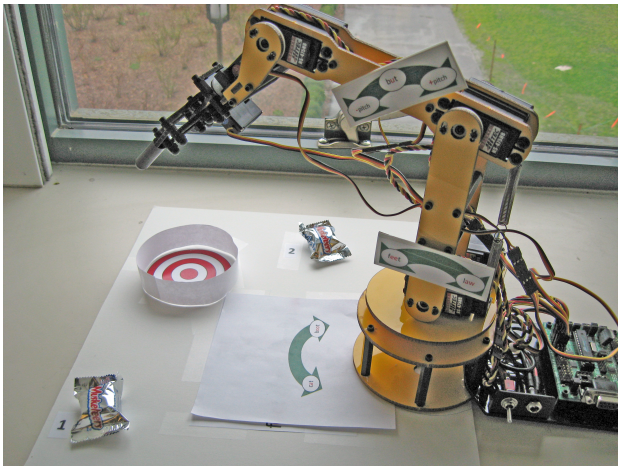

Figure 10. The set-up of the timed trial for the forward kinematic control method. Note the paper diagrams both on the robot and at the base.

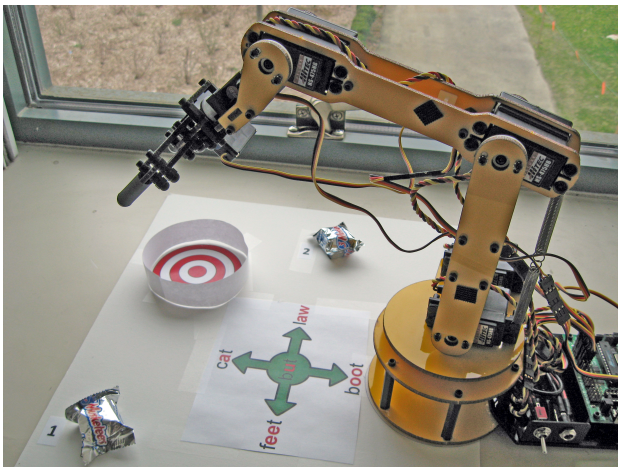

Figure 11. The set-up of the timed trial for the inverse kinematic control method. Note the paper diagram at the base.

desired. The board with the target and candy placements was not visible during the practice sessions. Moreover, we offered coaching as needed during the practice sessions. The protocol was as follows:

Introduction: Users were given a brief introduction to the Vocal Joystick, and the basic goal and methods of the study. The users were given a chance to ask questions following this introduction.

Adaptation: At this point, users were taught the five vowel sounds used to move the arm. Next, using the VJ adaptation algorithm, the VJ system adapted its internal parameters to better match the user by having them produce 2 second segments of each vowel. The discrete sounds [ck] and [ch] were also adapted at this time.

Method Description I: Next, the user was given an explanation of the first control method. Arrows were placed on or near the robot to help remind users of the mappings of the

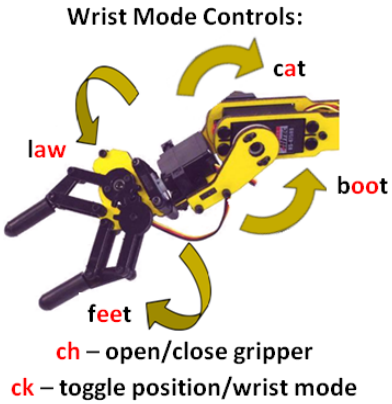

Figure 12. Reference sheet used throughout the user study.

sounds. These arrows remained in place during the timed trials, as seen in Figures 10 and 11. Users were also taught to use a [ch] sound to open and close the gripper and a [ck] sound to change position/orientation mode. A reference sheet seen in Figure 12 depicting the vowel mappings for orientation and a reminder of [ch] and [ck] usage was given to the user to hold for the remainder of the study. Users were warned that talking and laughing may be interpreted as a discrete sound. They were also warned that the robot has hard-coded limits on its motion, so if the arm stops moving downward, it may be at its ground limit.

Practice Session I: Each user was allowed up to 10 minutes to practice, although they could choose to stop when they felt comfortable with the control method. Within this allocated practice time they could ask questions, and we offered coaching if they were struggling with any aspect of the control method. Once they finished practicing, we reset the arm, and placed the board for the user study in front of the robot along with the 2 pieces of candy.

Timed Trial I: Once everything was set up for the timed trial, users were told not to ask questions during the trial since we could not offer any more help. When they were ready, the timer was started. After they placed the second piece of candy on the target, the timer was stopped. The time to complete the task was recorded, as well as the number of any discrete sound detection errors. Users were given a brief period to discuss any comments they had after the first trial.

Method Description II: New arrows were placed on or near the robot for the second control method, and again the users were given an explanation of how the vowels and pitch would move the robot. They were allowed to ask questions before starting the practice session.

Practice Session II: This was also the same as for the first run, and coaching for new difficulties was offered.

Timed Trial II: Again, the users indicated when they were ready to start this trial, and we stopped the time only once both pieces of candy made it to the target. The time to complete each run was recorded, along with the number of discrete sound errors.

Measurement	Mean	St.Dev.
Time, FK (m:ss)	2:31	2:02
Time, IK (m:ss)	2:18	0:55
Prefer IK (% users)	75%	N/A
Difficulty (FK)	3.2	0.9
Difficulty (IK)	2.7	0.9
Difficulty (Pitch)	3.5	1.1
Difficulty (Vowels)	1.8	0.7
Difficulty (ch/ck)	2.8	1.1

Table 2. Aggregate results from user study; top portion values were directly measured, bottom portion values are based on 5-point Likert scale (1 = very easy).

Short Interview: Users were asked to choose their preferred control method and answer other questions on a 5-point Likert scale, and then allowed to qualify any answers with further comments.

Results

Significant Results

Significantly, **all** users were able to use the VoiceBot to accomplish the tasks with both control methods. Table 2 shows results indicating that there were some clear preferences expressed by the users that are worth mentioning.

IK control was preferred to FK by 75% of users (not quite statistically significant) despite the rather small difference in average completion times. Removing one clear outlier in FK completion time lowered the mean to 1:54 and the standard deviation to 0:32. The largest IK time was less of an outlier; removing it lowered the mean and standard deviation to 2:06 and 0:40, respectively. Nevertheless, users felt using IK was slightly more intuitive, even though they also felt it was slower and produced jerkier movements than the FK method. In rating the difficulty of each method, the IK control was rated easier than FK, but the difference is not large. For this reason it may be worth investigating a third control option, perhaps the IK-Cylindrical, or to try increasing the average speed of movement for the IK-Cartesian method.

Next, discrete sound detection was problematic for several users; three users ranked the difficulty of using the discrete sounds difficult or very difficult, and false positives or negatives were notable for some users. Five of the twelve users had at least 6 false negatives for [ch] in a single run. Also, large numbers of false positive detections of the [ck] sound affected three users. For some users there were very few discrete sound errors, but for those who did encounter difficulty, problems were severe: two users had a maximum of 21 and 15 missed [ch] sounds in a single trial. In such a simple task, this was the source of much frustration for users. In one case, we tried to re-adapt the discrete sound [ch] between timed trials, but there was no improvement in performance.

The last major result to glean from the numbers in Table 2 is that pitch was a significant challenge for many users. Seven users rated the difficulty of pitch control difficult or very dif-

ficult, and of the remaining users, only two claimed pitch was easy to control. Some of this difficulty could be due to a lack of practice controlling pitch; one user, for instance, sings regularly and was able to control pitch easily, rating it very easy to control. Since pitch was on a linear Hertz scale, users were unable to get high resolution changes via pitch in the lower register. Consequently, some users had to use higher pitches than would normally be comfortable.

Other User Comments

Aside from the standard questions presented to all users, we also asked for any additional comments they had about the system. Some responses were predictable from the numbers seen above. For example, 42% of users stated that [ch] false negatives or [ck] false positives were frustrating, and 50% of users said that controlling pitch was somewhat tiring. A total of 42% of users noted that their lower register did not work as well for controlling pitch.

Half of the users said that in the FK method, they disliked trying to control the shoulder and elbow bend together to move the arm outwards and inwards. However, most of these users also said that they really liked the base (shoulder) rotation, since it felt intuitive to the arm. This suggests that the IK-Cylindrical method might be preferred to either of the two tested methods. The structure of the arm may have influenced these comments, since the shoulder rotation produces a significant amount of gross positioning. This may not be the case with other robotic arms, especially if the arm is similar to a natural human arm that would not have a base rotation.

One final comment was that 25% of users found pitch control difficult with [uh]. That vowel, a schwa, is used as a carrier vowel to control only pitch movement without the movement effects of any other vowel. Most beginning users prefer to use [uh] initially to practice controlling each DOF independently. However, some users has trouble producing a significant pitch change with that vowel, and resorted to [uw] or [iy] for successfully changing pitch, which caused other unwanted movements. The schwa is theoretically the most phonetically neutral of all the vowels for everyone, but using a different vowel as a neutral carrier may be advantageous. An alternative solution is to adapt the model for a longer period of time thereby effectively adjusting the central value to whatever is more comfortable for each user.

Discussion

The above results clearly demonstrate the feasibility of non-verbal voice-controlled robotics, and portend extremely well for future research. We would be remiss, however, if we did not carefully discuss some limitations of the current VoiceBot system. These issues will, of course, be addressed in future developments.

Pitch and Pitch tracking: As mentioned above, our pitch response was initially based on a linear frequency mapping, with pitch changes calculated using linear frequency. This reduced the effective resolution for pitch control of the arm in the lower frequencies since a perceived change would produce greater amounts of movement in higher ranges than in lower ranges. Log-scale frequency, something much closer to the human perception of pitch, is an obvious improvement. We have since the study updated the system to use a simple log base-2 scale for pitch, and informal results show that control is much easier under this model. Of course, more research is necessary to determine if a simple log-scaling is sufficient, or if a more sophisticated pitch-scaling method that models the human perceptual system (such as the mel-scale [16]) would be superior.

Another pitch option is to improve the pitch tracker itself. Unlike most pitch trackers, ours has the requirement that it needs to be both accurate and fast (no more than about 90ms of latency between when the user first adjusts their pitch and when the response to this change is realized). Of course, there is an accuracy/response time trade-off, and we optimized for response time. Therefore, our pitch tracker experienced some halving and doubling errors. We are currently accounting for this by setting a threshold for the maximum change in pitch between adjacent time frames. This simple fix seems to work well for now, but will likely need to be addressed if pitch is to be used in more intricate ways in the future.

Sensors: The current arm has no basic sensors to protect itself or to perform simple tasks autonomously. A force sensor in the gripper to aid in grabbing objects and proximity sensors on the arm to locate walls or other obstacles will be necessary in a more powerful arm. The arm is currently using preset limits on the range of motion to prevent injury to the arm itself, sometimes preventing what users may think is a reasonable movement. Sensors could also allow an arm to perform certain actions autonomously, although higher-level commands would likely work best if the VJ was set up to function alongside a speech recognition system.

Gravity compensation: As the arm is extended from the base, the strain from gravity increases. Consequently, the arm tip arm is often lower than ideal calculations predict when the arm is fully extended. For now, a simple linear equation is used to predict the position of tip relative to the ground for any radial distance, based on empirical measurements, with an accuracy of about a centimeter. A simple upward bias as the arm is extended may help provide better compensation, although a more sophisticated solution may be necessary.

Calibration: The Lynx 6 arm's servos tend to settle over time, mostly affecting the arm's interaction with the ground. As the gears and motor in the servos wear down, the regression to avoid floor contact should be periodically recalculated to alleviate this problem. To accomplish this, we calibrate by moving the arm to touch the ground at several radial distances, and record the VoiceBot's perceived height of the ground for each radial distance. A plot of this data is very nearly linear, and the best fit line is used as the gravity compensation equation we discussed earlier. Of course, a more powerful and expensive arm would alleviate this issue.

CONCLUSIONS

In this work, we have introduced and evaluated the VoiceBot, a voice controlled robotic arm, implemented using the Vocal Joystick inference engine. We have evaluated such a system in two settings, the first a simple 2D simulated world, and the second a real 3D robotic arm manipulating objects in the environment. We have conducted preliminary studies for both platforms.

As far as we know, the results presented in this paper are the first instance of a *non-verbal* voice-controlled robotic arm, and up until now it was not known if such an approach allowed one to perform even simple tasks such as moving candy from an initial position to a target. Critically, our results demonstrate that the approach is quite feasible. Additionally, our approach can augment existing systems, allowing, for instance, the system described in [2] to function under hands-free user control if desired. It is believed, therefore, that with further research into voice-controlled robotics, including target-population studies on how best to avoid fatigue and reduce learning time, a system could be created to help individuals with motor impairments lead more independent and productive lives.

Lastly, we wish to thank the anonymous reviewers for their useful comments. This material is based on work supported by the National Science Foundation under grant IIS-0326382

REFERENCES

1. BILMES, J., ET AL. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In *Proc. Human Language Tech./Epirical Methods in Natural Language Processing* (Oct. 2005), pp. 995–1002.

2. CHOI, Y. S., ANDERSON, C. D., GLASS, J. D., AND KEMP, C. C. Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired. In *ASSETS* (Oct. 2008), pp. 225–232.

3. HARADA, S., LANDAY, J., MALKIN, J., LI, X., AND BILMES, J. The Vocal Joystick: Evaluaton of voice-based cursor control techniques. In *ASSETS* (Oct. 2006), pp. 27–34.

4. HARADA, S., WOBBROCK, J., AND LANDAY, J. VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. In *ASSETS* (Oct. 2007), pp. 27–34.

5. HARADA, S., WOBBROCK, J. O., MALKIN, J., BILMES, J., AND LANDAY, J. A. Longitudinal study of people learning to use continuous voice-based cursor control. In *CHI* (Apr. 2009).

6. IGARASHI, T., AND HUGHES, J. Voice as sound: using non-verbal voice input for interactive control. In *UIST* (2001), pp. 155–156.

7. KIM, S.-P., SIMERAL, J. D., HOCHBERG, L. R., DONOGHUE, J. P., AND BLACK, M. J. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of Neural Engineering 5*, 4 (2008), 455–476.

8. LI, X., MALKIN, J., BILMES, J., HARADA, S., AND LANDAY, J. An online adaptive filtering algorithm for the Vocal Joystick. In *Interspeech* (Pittsburgh, PA, Sept. 2006).

9. LI, Y., WANG, C., ZHANG, H., AND GUAN, C. An eeg-based bci system for 2d cursor control. *IEEE Int'l Joint Conf. on Neural Networks* (June 2008), 2214–2219.

10. MALKIN, J., LI, X., AND BILMES, J. Energy and loudness for speed control in the Vocal Joystick. In *Automatic Speech Recognition and Understanding* (Dec. 2005), pp. 409–414.

11. MARTENS, C., RUCHEL, N., LANG, O., AND GRASER, A. A FRIEND for assisting handicapped people. *IEEE Robotics and Automation Magazine 8*, 1 (2001), 57–65.

12. MIHARA, Y., SHIBAYAMA, E., AND TAKAHASHI, S. The Migratory Cursor: Accurate speech-based cursor movement by moving multiple ghost-cursors using non-verbal vocalization. In *ASSETS* (Oct. 2005).

13. REICHENSPURNER, H., ET AL. Use of the voice-controlled and computer-assisted surgical system ZEUS for endoscopic coronary artery bypass grafting. *J. Thoracic and Cardiovascular Surgery 118* (1999), 11 – 16.

14. SHENOY, P., KRAULEDAT, M., BLANKHERTZ, B., RAO, R., AND MUELLER, K.-R. Towards adaptive classification for BCI. *J. Neural Eng. 3* (2006), R13–R23.

15. SPORKA, A. J., AND SLAVÍK, P. Vocal control of a radio-controlled car. *ACM SIGACCESS Accessibility and Computing*, 91 (2008), 3–8.

16. STEVENS, S. S., VOLKMANN, J., AND NEWMAN, E. B. A scale for the measurement of the psychological magnitude pitch. *J. of Acoustical Society of America 8*, 3 (1937), 185–190.

17. WANG, L.-C., AND CHEN, C. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Trans. on Robotics and Automation 7*, 4 (1991), 489–499.

18. WELMAN, C. Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Fraser University, Burnaby, BC, 1993.

19. WON, S. Y., LEE, D.-I., AND SMITH, J. Humming control interface for hand-held devices. In *ASSETS* (Oct. 2007), pp. 259–260.