Control of Simulated Arm with the Vocal Joystick

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

ABSTRACT

This work explores the use of the Vocal Joystick (VJ) for robotic limb control using a simulated environment. We demonstrate that continuous vocal control is suitable for real-time manipulation of a 3 joint robotic limb in 2-D. We explore several kinematic models by which joint angles are specified, introducing a hybrid system combining forward and inverse kinematic models. Beyond demonstrating the feasibility of control, our results also suggest that the standard kinematic models are preferable to the new model.

Author Keywords

Voice-based interface, speech recognition, assistive device, continuous input

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, multiple sclerosis, or spinal cord injuries rely on others to assist them their in daily activities. Advances in assistive technologies have begun to provide an increase in independence for these persons, but there is great potential for further improvement in their overall quality of life.

One of the greatest challenges faced in assistive technology is that control options are extremely limited when the target user has little or no use of their limbs. Spoken language and automatic speech recognition (ASR) systems are often considered a natural solution to the problem. Unfortunately, speech is limited to discrete commands, falling short especially on steering tasks, which require continuous control.

The Vocal Joystick (VJ) [3] seeks to use the human voice without the constraints of spoken language. Instead, the user relies on continuous sounds that can vary in pitch, vowel quality, or amplitude to provide control of computer applications or possibly even electro-mechanical devices. The VJ engine is designed to be reusable infrastructure available to any application. Existing work on a VJ-driven mouse has demonstrated the suitability of the system to 2-D control [4, 11, 8]. The current VJ mouse application enables the user to navigate a windows/icons/mouse/pointer (WIMP) interface using only vocal parameters. This innovative design permits continuous real-time control of the mouse using only a standard headset microphone.

In this work, we take a step towards vocal controlled robotics by looking at the potential of the VJ for an interface to a simple simulated 2-D robotic arm. Current control methods for assistive robotic manipulators tend to be expensive, difficult for motor impaired persons to use, or highly invasive. 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 using continuous vocal control.

BACKGROUND RESEARCH

Related Work

There already exist several assistive technologies using novel control methods for manipulating a robotic arm. These technologies include physical joysticks, speech-based robotic controllers, and brain-computer interfaces (BCI). Each seeks to increase independence of persons with disabilities with varying degrees of expense, invasiveness, and accuracy.

The most widely accepted method for control of robotic limbs is currently the physical joystick. Commercially available robotic arms often include a joystick feature [7] that allows the user to position the gripper or tool at the end of the arm, called the end effector. These controls tend to be inexpensive, non-invasive, and fairly accurate. Limits in hand mobility, however, eliminate this option for many individuals.

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 [1]. The entire system is controlled by a speech interface with 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.

BCI control of robotic limbs has shown considerable promise in the past decade. Non-invasive methods use signals from electrodes on the scalp surface [9], while so-called partially invasive techniques place electrodes under the skull on the brain surface. The partially invasive method shows better control but is still susceptible to noise, and control of more than one dimension has proven difficult [6]. Invasive BCI devices produce the highest quality signals from measurements of motor neurons, but also require placing electrodes directly in brain tissue.

Most recently, Hochberg et al. [2] produced the BrainGate

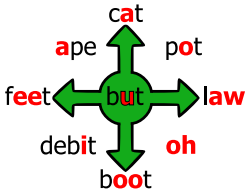

Figure 1. Vowel mapping to cursor movement for mouse control. Words include the nearest vowel in American English.

implant for humans. A pilot version of this implant was tested in a tetraplegic man, who was then able to train himself to navigate a computer and control a rudimentary robotic arm. This research demonstrates that brain signals may be a possible form of real-time continuous control in assistive devices. The most pressing concern for this technology is its inability to function as a long-term solution. Scar tissue develops in the brain near the electrodes causing interference. This method is also still highly invasive and expensive.

Vocal Joystick

The Vocal Joystick relies on the great flexibility of the human vocal tract to produce a range of continuous sounds, providing a unique voice-controlled interface. No specialized or invasive hardware is required to use the VJ; a standard microphone and sound card are sufficient. The core engine is a library that can be used by any application, under Windows or Linux, to provide control parameters, so possible applications for the VJ are virtually unlimited.

The VJ engine extracts several vocal parameters which can then be mapped into motion [3]. The current system uses loudness, pitch and vowel quality; the latter is the specific vowel sound produced. In previous work, we have 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 [8]. In addition, we have shown that the Vocal Joystick is also suitable for steering tasks, situations where word-based control models are sorely lacking [11].

For mouse control tasks, we currently map vowels into two-dimensional space as seen in Figure 1, where each vowel controls one direction. Arbitrary directions are possible via adaptive filtering [11]. Loudness controls cursor velocity with louder sounds corresponding to faster movement. Pitch is currently unused in VJ mouse control.

CONTROL MODELS FOR ARM SIMULATION

There are a number of possible control models for a VJ-driven robotic arm. For this preliminary work, we extend the system to simultaneously control three joint angles, but limit the simulated environment to only two dimensions. We test three ways of determining joint angles for our simulated robot, 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 akin to intentional loudness [4]. Such considerations are beyond the scope of this work, although they may strongly influence results.

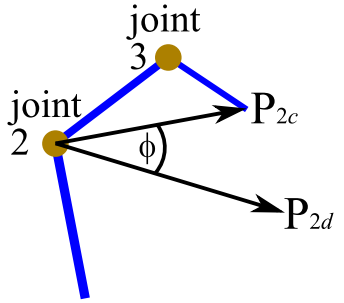

Figure 2. Example of calculation to update joint angle 2 aiming for target point P_{id}. Joint angle 3 is fixed for this calculation, resulting in a 1-D optimization problem.

Forward Kinematic Model

Forward kinematics requires the user to explicitly set each joint angle. 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 user effort since the user's attention is split between accomplishing the task and the mechanics of how to realize that goal.

Inverse Kinematic 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 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 computational power. 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 have chosen one, cyclic coordinate descent (CCD) [5, 10], 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 independently, starting from the end and working towards the base. Each joint angle is updated before proceeding to the next one, and once adjusted is considered fixed for the rest of that iteration. This means that we need solve only a 1-D optimization problem for each joint. Figure 2 depicts the situation graphically for a three joint arm.

For joint $i \in \{1, 2, 3\}$, we can rotate the vector from joint i to the end effector, $\mathbf{P}_{ic}$, by an amount ϕ to line it up with the vector from the joint to the desired location, $\mathbf{P}_{id}$. We can treat ϕ as a free parameter and calculate a rotated vector $\mathbf{P}'_{ic} = \mathbf{R}(\phi)\mathbf{P}_{ic}$ where $\mathbf{R}(\phi)$ is a rotation matrix. For each

step, we want to find the angle θ_i^{new} which places the tip of $\mathbf{P}'_{ic}$ as close as possible to $\mathbf{P}_{id}$. This point will be where the vectors are parallel, so our goal is equivalent to maximizing the dot product of the two vectors: $\phi^* = \text{argmax}_\phi \mathbf{P}_{ic} \cdot \mathbf{P}_{id}$. This gives a new joint angle of $\theta_i^{new} = \theta_i + \eta_i \phi^*$, where $\eta_i \in [0, 1]$ is an optional penalty or "stiffness" of the joint.

Hybrid Model

Between the forward and inverse kinematics 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 CCD-based inverse kinematics, and the last joint angle is controlled directly. This model should provide the advantages of the other models, allowing the user to remain more focused on the goal while explicit use of the last arm segment enables smaller adjustments.

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 that such control is possible. The second was to look for preliminary evidence supporting our hypothesis that the hybrid model would be preferred over the other two. The hybrid, in theory, allows simple arbitrary positioning while retaining an additional degree of freedom for small adjustments. A video showing the test application can be found at `http://ssli.ee.washington.edu/vj/submissions/vocal-input-07.avi`

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 into four sequential locations for each control model. The four locations of the target positions were kept constant for each model since the use of different control models minimizes possible memory effects, and they were designed such that significant movement of the arm was required to reach each new location. A timer in the program 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 the program recorded the number of such resets. All models used 3-segment arms with each segment 2/3 the length of the previous one; see Figure 3 for an example. 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 can focus their attention on controlling the simulated arm. As a result, we consider this to be a feasibility study that may simply suggest performance trends.

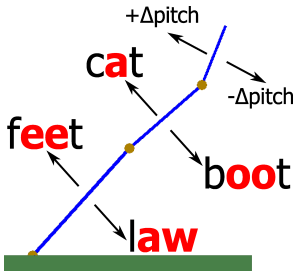

Figure 3. Vowel sounds associated with arm movement for the forward kinematic control model. The underlying image (without labels) is an arm from the test application.

Model	User 1	User 2	User 3	User 4	User 5*
Forward	602	96	292	138	77
Inverse	216	141	135	230	59
Hybrid	663	175	144	185	107

Table 1. Task completion times in seconds for each user under each control model.

For the forward kinematic model, we use the x and y axes from mouse control to move the first and second joint angles, respectively, and the derivative of pitch for the third joint angle, as shown in Figure 3. We used a simple mapping where 1 pixel of movement from mouse control corresponded to a 1 degree rotation. Large discontinuities were removed from the pitch derivative before scaling by 0.3, the same sensitivity used for the other movement calculations [4].

All inverse kinematic models specify end effector locations using the same mapping as for VJ mouse control, as shown in 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 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 achieve 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. The first observation is that all users were successfully able to complete all tasks, thus proving the feasibility of the concept. Users 1, 2 and 3 had experience with the VJ mouse but this application was new to both of them. 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 2-D positioning well from their VJ mouse experience. This meant they had a less thorough understanding of how the

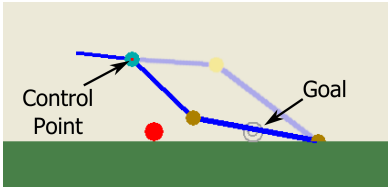

Figure 4. Example of hybrid mode in an undesirable configuration. The faded arm would be a better position since the arm currently prevents the red ball from reaching the goal. Users often had trouble helping CCD reach the faded position while moving the control point via inverse kinematics in this model.

inverse kinematics positioned the arm.

User 5 was the primary developer, and results presented here are the best times under the same testing configuration as the other users. Those times, the results of much practice associated with building and testing the system, are included to show the currently known lower bound 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 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, pitch was a more difficult vocal parameter for users to control, 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. The hybrid mode was strongly the least favorite method for all but user 3, and forward kinematics was in the middle.

The reason for poor performance with the hybrid mode is that the joint between the first and second arm segments often becomes "stuck" bending to one side due to the inverse kinematics algorithm. Trying to find a movement so the algorithm will bend the joint to the other side can take substantial effort. Figure 4 shows how a bad joint angle can prevent the ball from reaching the target.

By adding an extra joint angle, the 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, of course, avoids the problem entirely.

Since the test subjects were drawn from the VJ development team, we have not attempted to draw any performance conclusions from these results. Despite that, we are comfortable in declaring 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 CCD's weaknesses. The practiced results from User 5 suggest that the inverse kinematics model may be slightly 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.

CONCLUSIONS AND FUTURE WORK

The results of this proof-of-concept study are quite exciting. It has been shown that the Vocal Joystick is indeed suitable for control of a robotic arm in two dimensions. While we do not claim statistical significance for the time trial results, they will prove valuable in considering control options in for future development in three dimensions.

As mentioned while describing inverse kinematics, we selected CCD for a few practical concerns. In practice, CCD usually works well, but the method is described as similar to pulling on the end link of a chain. Some other methods [10] are described as more like bending a flexible piece of plastic which, at a high descriptive level, seems to be more akin to the behavior people expect. Overall, users prefer not having to pay attention to all the joint angles, but movement produced by another algorithm may prove more favorable. Alternatively, it may be possible to refine the joint angle stiffness weights to produce a more desirable effect.

The hybrid method suffered from the stuck joint problem, a situation created because CCD uses local optimization which does not take into consideration obstacles such as the ground. Using pitch to control the joint at the base of the arm may help avoid the problem by creating more configurations where the ground does not interfere with the inverse kinematics.

Finally, we would like to thank the members of the Vocal Joystick team who volunteered for this study: Kelley Kilanski, Susumu Harada, and Xiao Li.

REFERENCES

1. C.Martens, N.Ruchel, O.Lang, and A.Graser. A FRIEND for assisting handicapped people. *IEEE Robotics and Automation Magazine*, 8(1):57–65, 2001.
2. L. R. Hochberg et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, July 2006.
3. J.Bilmes et al. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In *Human Lang. Tech. Conf./Conf. on Empirical Methods in Nat'l Lang. Proc.*, 2005.
4. J.Malkin, X.Li, and J.Bilmes. Energy and loudness for speed control in the Vocal Joystick. In *Automatic Speech Recognition and Understanding*, San Juan, PR, Dec. 2005.
5. L.-C.Wang and C.C.Chen. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Trans. on Robotics and Automation*, 7(4):489–499, 1991.
6. E. C. Leuthardt et al. A brain-computer interface using electrocorticographic signals in humans. *J. Neural Eng*, 1:63–71, 2004.
7. Lynxmotion. Lynx 6 robot arm. `http://www.lynxmotion.com/Category.aspx?CategoryID=25`.
8. S.Harada et al. The Vocal Joystick: Evaluaton of voice-based cursor control techniques. In *SIGACCESS Conf. on Computers and Accessibility*, Portland, OR, Oct. 2006.
9. P. Shenoy et al. Towards adaptive classification for BCI. *J. Neur. Eng.*, 3:R13–R23, 2006.
10. C. Welman. Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Fraser University, Burnaby, BC, 1993.
11. X.Li, J.Malkin, et al. An online adaptive filtering algorithm for the Vocal Joystick. In *Interspeech*, Pittsburgh, PA, Sept. 2006.