Communication through Movement: An Alternative Method of Interaction for HRI

Xiang Zhi Tan  Aaron Steinfeld  M. Bernardine Dias
Robotics Institute  Carnegie Mellon University
{zhi.tan@ri., steinfeld@, mbdias@ri.}cmu.edu

Abstract—Human-robot interaction (HRI) often relies on audio and visual channels for communication between robots and human users. While these methods have proven to be successful, the reliance on these two channels limits their effectiveness in extreme environments and for people with certain disabilities. In this paper, we present an alternative communication method in which users manipulate the robot’s arm as a form of input while also using the arm for output. This haptic interaction enables users to give inputs while maintaining contact with the arm, and in turn helps visually impaired users by avoiding the need to search for other interaction elements of the robot. As the user moves the robot’s arm around, resistance is generated to guide user movements. The system learns user-defined gestures using SVM and is able to classify subsequent gestures and their associated user intentions. We demonstrate that the system can learn and classify different gestures by conducting a preliminary exploration with four users. Different configurations and gestures are further discussed to demonstrate the scalability of the system to different robotic platforms. We also discuss scenarios that benefit from simultaneous physical input and output across a wide range of communities, thus illustrating a step towards a universal design method for physical HRI.

I. INTRODUCTION

As the field of human-robot interaction (HRI) continues to mature, many methods of direct communication that exchange information between user and robot have been explored. Prior work in social HRI used dialog [2], visual or touch displays [18], and visual cues [32] as methods of communication between the robot and users. These interaction strategies are often drawn from social norms in human-human social interactions. While these methods have proven to be effective, there are situations where these forms of communication might be undesirable. Environmental issues such as ambient noise and light could affect the reliability of these methods. Also, if the user is receiving assistance through the robot’s arm, using input devices like touch displays would cause inconvenience to the users by requiring them to move their hand from one robot component to another. Furthermore, these methods’ usability can also be severely affected by task-related restrictions like underwater activities where speech is restricted or hazardous conditions that are noisy or have severely limited visibility. Users with disabilities like speech disorders or visual disabilities also face difficulty when using traditional methods. We believe there is a need to explore other modalities for interacting with robots that exist beyond the common human-human social interaction paradigm. While there are possible alternatives like adding pressure sensors or buttons, new users or visually impaired users could encounter problems locating and using these interfaces. This was evident in our initial exploration where blind users encountered trouble with our robot when alternating between holding the robot’s manipulator and pressing buttons due to not being able to see where they were reaching. The design and functionality for specialized buttons and interfaces are also platform-specific and harder to generalize across platforms, compared to visual and audio channels. We aim to find an alternative, intuitive method of interaction that can generalize across platforms and utilize a physical communication channel.

As a first step in our exploration, we demonstrate a haptic user interface that repurposes the manipulator arm as the input device. Our system uses machine learning techniques to recognize different hand gestures by the user when they hold onto the manipulator arm. The movements are recorded by the arm’s internal sensors and classified into different static gestures that are learned from the users beforehand. The system will then executes different callback function defined in the interaction. Our system addresses some of the issues described in the previous paragraph. For instance, the system relies on haptic modality that is underutilized in current social HRI research and also reuse existing hardware on many robot platform. We demonstrate that the system would work across different platforms with minimal changes.
The use of the manipulator arm as the input device will allow users receiving assistance from the manipulator arm to seamlessly transition from receiving services to giving input without letting go. For instance, a user receiving physical assistance from a manipulator arm can apply a stop gesture to the manipulator arm to order it to stop. This method is also more discreet and less likely to be affected by environmental factors.

Gestures can be categorized into two categories, static, where the hand is in a fixed configuration or pose, and dynamic, a continuous series of different poses. In this paper, we mainly focus on static gestures created through hand poses. These hand poses are captured and classified by the learning algorithm. Our system also provides some temporal information like the duration in the current pose. A possible application is a volume control interface with internal states and gestures shown in figure 1.

In this paper, we first describe an exploratory study with a group of people who are blind or low vision to determine feasibility of this method. We then describe the underlying system, the machine learning component and the haptic feedback. We follow with a demonstration of the system through a preliminary exploration with four users without disabilities, and explore how the system would generalize across robotic platforms and applications. We end by exploring how this system could be integrated into existing systems and the universal design implications on HRI.

II. RELATED WORK

Prior work and historical examples for HRI systems similar to the one proposed here are sparse. Hence, we identified four areas of research that informed our approach.

A. Communication methods in Human-Robot Interactions

Common mediums of communication in HRI involves gestures, natural language, physical interactions with haptics and visual displays. Physical interactions are defined as tasks where robots carrying out physical tasks to help users, such as rescue robots lifting casualties. However these are either an output or simple touch detection on the robot, which did not explore possible interaction paradigms that physical interaction enables. The main mediums discussed by Goodrich are still actively being used in HRI research today, for instance, visual displays, speech, gestures and haptics.

B. Hand Gesture Recognition Methods and Applications

Interaction through gesturing while uncommon in HRI, have long been used in other field. Early attempts in gesture recognition were done through wearable glove devices. These systems detect users’ hand positions and orientation through multiple sensors embedded in the glove. Recent work demonstrated wearable smart watch that can recognize hand and finger gestures through sensors in the device. Researchers have used these devices to control different devices such as home automation systems.

Besides wearable devices, developers have long looked into vision-based gesture recognition. In 1995, Freeman and Roth demonstrated a computer vision system that recognizes hand gestures and use them to control a graphical interface. Since then, researchers have proposed multiple different methods of detecting hand gesture such as HMMs, Haar-features, and convolution neural networks.

With the ability to do vision gesture recognition, researchers have explored how to use gestures in robotic systems. Fong et al. showed a system that recognized static gestures to teleoperate a mobile robot. Monajjemi et al. created a system that allows users to select and control multiple quadcopters through gestures. Besides teleoperation, Xiao et al. created a robot that can understand the user’s body language and gestures like “praise” and “stop” and reacts to them with socially appropriate responses.

C. Use of Robot’s Arms

In HRI, robotic arms often serve as tools to pass or receive items from users or to perform physical tasks for the users. Researchers have also used robotic gestures to improve communication. Robot arms are also used in physical rehabilitation like assisting in rehabilitation for stroke or traumatic brain injury patients.

Besides using the robot’s arm as an output device, researchers also looked into teaching a robot by moving the robot’s arm using Learning from Demonstration (LfD). Researchers have also used haptic interfaces to teach motions to the robot and refine them. While both our system and LfD learn movement through the robot’s arm, LfD focuses on learning skills and redoing independently afterwards whereas our system uses it as a method of communication between the user and the robot.

D. Haptic Interfaces and Manipulandum

Haptic Interfaces are input devices that provide feedbacks through touch (force, vibration, etc.). One type of haptic interface is Manipulandum, a haptic interface built like a joystick which detects user movements and generates resistance when needed. Manipulandums are used to study motor learning, the study of how humans apply force on different objects and the neurological processes involved. It can also be used as a computer interfaces. For instance Brooks Jr et al. used a manipulandum-like haptic interface for manipulating virtual proteins displayed on a screen. Millman et al. created a manipulandum to interact with virtual worlds. These manipulanda have also been used as controllers for teleoperation. Lee et al. used haptic feedback on the manipulandum to convey distance to users when teleoperating.

Our system is different from existing haptic interfaces because we reused an existing hardware and does not limit the arm’s original movement and manipulation capability in the process. Another benefit of having a system that is both a manipulator and an input device is that when the robot is assisting a user with the robot’s arm, the system could alternate
between input and manipulation task fluidly, thus creating a better user experience.

III. EXPLORATORY STUDY

To explore how our target user group will react to and interact with this system, we invited four participants (all males, two blind and two with low vision) to help us explore the design space. During this exploration, the robot only ran the PD controller described in the following system design section. The participants were asked to complete three activities. First, they were instructed by the researcher to move the robot's arm in certain ways (e.g. move up, rotate clockwise). Afterwards, participants were given the imaginary scenario of navigating a list interface and asked to come up with gestures for actions like moving up the list and selecting items from it. At the end, a semi-structured interview was conducted to understand the participant’s experience.

While nearly all participants were neutral or positive about the interaction method, they were confused and had trouble perceiving how it would work in a complete system. We surmise this was partly due to the novelty of the interaction system and the stand-alone nature of the experience. We think the utility of the approach will be more apparent in the context of a complete implementation in more realistic application. This will better demonstrate our vision of assistance and communication with continuous hand contact. We also observed that participants interpreted the same verbal instructions like “forward” differently and movements varied in magnitude. When discussing the scenarios, participants appeared to have different mental models of the interface. Both of the participants who are blind saw the arm as a pointer and moved as they navigated the list, while the others had a mental model of the arm as a joystick where the list advanced once for each gesture. These findings motivated us to make our system customizable based on user preference and emphasized the importance of testing with a complete system.

IV. SYSTEM DESIGN

A. High-Level Overview

Our system is constructed from three components: the feedback controller, the learning system, and the classification system. A system flowchart is shown in figure 2. The feedback controllers allows the user to move the robot arm near the origin position and simulates resistance as the user moves the arm further away. At the beginning of each interaction, the learning system acquires the user’s preferences and movement patterns by instructing the user to move through all desired gestures and picking a pose for each gesture. The learning system then passes the trained classifier to the classification system which can uses it to classify subsequent movements into different gestures. An internal state machine keeps track of the current gesture as a state and executes callback during state changes. While the system can be deployed onto different robotic platforms, the system must fulfill one of the following two conditions: (1) the robot’s manipulator arm must be backdrivable and be able to detect the changes in joint angles, or (2) the robot must be able to detect external forces applied to the arm and react to it. We implemented the system on a Baxter platform which has 2 backdrivable 7 DOF arms.

B. Feedback Controller

As the feedback controller helps guide the movements and improves user experience, it is optional if the robot’s actuators are backdrivable. For our system, we used a proportional derivative (PD) controller as our feedback controller. At the start of each session, the arm moved to an initial position that becomes the origin. At each cycle, the controller calculates the applied torque on each joint as

\[ F(t) = K_p e(t) + K_d \frac{de(t)}{dt} \]

where \( F(t) \) is the commanded torques at each joint, \( e(t) \) is the difference between current and origin’s joint position, and \( \frac{de(t)}{dt} \) is the change in velocity. \( K_p \) and \( K_d \) is the coefficient for the proportional and derivative term that describe how much fast and how much torque the system applies.

In our system, \( K_p \) and \( K_d \) were selected through manual tuning so that the arm returns to the origin as quickly as possible without oscillation. We also increased the gain for joints that are further away from the user to increase their resisting force. Through those restrictions, we can encourage users to move in certain directions and appropriate amounts. These coefficients can be modified to suit the system needs and should be different if the robot is in a different configuration.

C. Learning and Recognition System

1) Features: Through testing, we determined that the features which best described the arm’s movements are the end effector’s location related to the robot’s body frame. This feature is generalizable as it is relatively easy to calculate using forward kinematics on other robots and morphologies. The different gestures are linearly separable in this feature space and can be classified using linear classifiers.

In our implementation, the end effector coordinates are measured at a rate of 100Hz. We then apply an Exponentially Weighted Moving Average filter (EWMA) to filter out the sensory noise in our measured features. EWMA uses the following equation:

\[ f_t = \alpha \cdot I_t + (1 - \alpha) \cdot f_{t-1} \]

where \( f_t \) is the filtered value at each time step, \( I_t \) is the raw input and \( \alpha \) is the weighting coefficient between 0 to 1 that determines how much weight prior values has on the new value. Through manual tuning, the \( \alpha \) value was set to 0.35.

2) Rule-based System: As the feature space can be easily visualized, one of the simplest methods of classification is a rule-based system, where system designers set the threshold for each gestures. For instance, they can define a gesture to the left as an offset of 0.005m to the left of the origin. One of the benefits of this method is that it is fairly simple and can be written by programmers without a background in machine

\[ \text{http://www.rethinkrobotics.com/baxter/} \]
learning or controls. However, there are a few issues with using a rule-based system. The rules are determined by the programmer and might not be suitable for the actual users. The programmer decides on the different gestures and it is hard for users to customize based on their preferences and needs. As the number of possible actions increases, the rules become complex and hard to maintain. Furthermore, the system cannot be generalized to other configurations easily as rules would differ based on arm configuration.

3) Machine Learning Techniques: Instead of manual coding, machine learning techniques could be used to learn rules across multiple platforms and allow individuals to customize their own set of gestures. Machine learning techniques do not rely on the programmer to know the user preferences in advance. In this paper, all the machine learning techniques were implemented using functions from scikit-learn [24], an open-source Python machine learning library. We picked three widely used machine learning techniques and evaluated them against a rule-based system. These were:

- **K-NN Classifier (KNN)** – This classifier finds the closest \( k \) examples of training data and chooses the label based on the majority for the output. Through manual testing on a training/test set by a member of the research team, we picked \( k = 25 \) which gave the best result.
- **Logistic Regression Classifier (LR)** – A logistic regression classifier creates a probabilistic model based on the training data that returns a probability of each label given the features. We use the one-vs-rest method to classify more than two states (multiclass).
- **Support Vector Machines Classifier (SVM)** – This classifier separates the feature space with a plane created by support vectors that best separates the labels. The classifier uses a linear kernel and one-vs-rest method for the multiclass classification.

**D. System Response**

Our controller and classifier ran at 100Hz, so the system received about 100 classifications in a second. Through preliminary testing, we noticed that the system was picking up transitional positions that the user moved through before reaching their desired gesture position. Instead of using every single classification, we use a windowed polling system in which at each time step, the newest label was added to a polling queue of size 10 and the majority label was reported.

The reported label was fed to the internal state machine that tracks the current gesture as a state. Each state can be attached with two callback methods, one for when the user transitioned into that state and the other callback was called every 0.75 seconds the user remained in that state.

**E. Test and Training Data**

To demonstrate the feasibility of our system, we collected training and test data from four people (three males and one female) without disabilities. We chose to recruit participants without disabilities to explore the feasibility for users with or without disabilities and a step towards designing a universal design system. Six labels, **origin**, **up**, **down**, **left**, **right** and **forward**, were picked as the training and testing gestures. Each participant was first asked by the system to move through all the gestures based on their interpretation of the name. The participants were only told that the **origin** position was the position from where they began. After each gesture, the participant was asked to go back to the **origin**. These 5 movements were recorded and used as the training data. After training, the robot was restarted and the participant was asked to perform a random sequence of the five specified gestures. Afterwards, the participants were asked again to do a random sequence of the specified gestures, with the addition of audio feedback (discussed in a later section of the paper). The last part was added to simulate an input/output system where feedback was given when the system recognized an input, such as software generated audio clicks on a rotary control interface. After the test, a member of the research team coded the ground truth for each sequence. This ground truth was used in evaluating the performance of our system.

**V. RESULTS**

A. **Classification Methods**

To determine the suitable machine learning technique to be used in the system, we compared the performance of all the machine learning techniques along with the rule-based system. Our first two measures are the accuracy calculated using both micro-averaging and macro-averaging as described in [28]. We then calculated the macro-averaging of the Precision, Recall and F1-score of each method. The results are shown in table I. Each value is the average from all the participants.

We observed that nearly all the machine learning techniques performed at a similar, if not better, level as the rule-based
system for most measures. We did notice that the rule-based method was outperformed on the precision and recall measurements but still had a similar micro-averaging accuracy score as the machine learning techniques. This might be caused by the origin position being overrepresented in the test data and the rule-based method had higher accuracy in that state. This is supported by the confusion matrix where the rule-based method’s true positive for the origin state is 0.797 while the machine learning techniques average 0.44. Since the user needed to go through the origin state to reach other states, techniques that do better at detecting the origin state will have better results in micro-averaging methods since the origin state has the label highest count.

Another measure we employed is the representation of classification through confusion matrices. The combined confusion matrix can be seen in figure 3. As seen in the confusion matrix, SVM and K-NN have no problem classifying most states, with the exception of the origin position. This state is often misinterpreted as an up gesture. Logistic regression also struggles with forward gesture in addition to the origin state. The rule-based method has difficulty distinguishing between forward and up gestures and also has some issues in classifying the origin and down gestures. The mislabeling of multiple states explains the low macro-averaging scores we saw above as the individual accuracy of multiple classes were lower. In general, SVM performed slightly better than the other three techniques.

From both the confusion matrix and performance measures (accuracy and F-scores), we see machine learning based systems perform similar if not better than an expert created rule-based system. While all three machine learning techniques could be used in the system, we picked SVM to be used in our system for two reasons: (1) SVM was the best in all our measurements and (2) it is less susceptible to noisy data. For K-NN, a large number of mislabeled training data will significantly skew the results. In one of our initial tests, an incorrect labeled data caused K-NN’s performance to drop to the mid 50 percent range. While Logistic regression and SVM are both suitable, we chose SVM due to the slightly better performance.

This choice is also supported by individual results as shown in table II where LR and SVM performed similarly across participants. However, we see a slightly better result with SVM for users with lower performance (P3 and P4). P4’s accuracy (micro) increased from 0.549 to 0.626 and P3 saw a slight increase in accuracy (micro) of 0.023.

**B. Additional Sensory Feedback**

While the classification methods work well at classifying gestures, we suspected we could further improve the micro result by providing additional feedback to users when the system registered a gesture and thus allowing them to readjust their gestures. Depending on the interaction needs, this feedback could be haptic, audio, or visual. In our test, we provided audio feedback letting users know when they entered a new state and repeated the feedback at a 0.75 second interval if they remained in the same state. The feedback was done by attaching a ROS text-to-speech module to the function callbacks when the user transitioned to a new state and when they remained in the state.

To test the system, the same participants were asked to execute a random sequence of movements with the audio feedback enabled. The training data for the classifier consisted of additional noise caused by the code recording extra data after the system stopped. However, we do not think this slightly flawed classifier could have changed the result as there were larger portions of correct data and the SVM method would have minimized the possible difference. The audio feedback was triggered by the SVM classifier trained on their own training data. The predicted label of the classifier was then compared with the ground truth.

We saw improvement for all the users on both the macro and micro measurements with the addition of audio feedback. There were large improvements in the accuracy (micro)

**Fig. 3: Confusion matrix for each technique.(from left to right) Rule-Based system, Support Vector Machine, K-Nearest Neighbor, and Logistic Regression.**

| TABLE I: The average performance measurement of each technique. The standard deviation is shown in parentheses. The subscript \( m \) represents micro-averaging, where as \( M \) means macro-averaging. |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | Rule-Based | SVM          | LR             | KNN             |
| Avg. Acc. \( m \) | .757 (.166) | .751 (.170)  | .728 (.199)   | .732 (.202)   |
| Avg. Acc. \( M \) | .919 (.055) | .917 (.059)  | .909 (.066)   | .910 (.067)   |
| Precision \( m \) | .751 (.145) | .902 (.071)  | .890 (.091)   | .897 (.078)   |
| Recall \( M \)    | .757 (.152) | .860 (.045)  | .844 (.045)   | .851 (.048)   |
| F1-score \( M \)  | .738 (.146) | .879 (.047)  | .868 (.057)   | .872 (.057)   |

| TABLE II: Comparison between individual for logistic regression and SVM. The subscript \( m \) represents micro-averaging, where as \( M \) means macro-averaging. |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | LR             | SVM             |                |                |
| Acc. \( m \)    | Acc. \( m \) | F1 \( m \)     | Acc. \( M \) | Acc. \( M \) | F1 \( M \) |
| P1              | .847           | .847           | .839           | .846           | .872     |
| P2              | .947           | .947           | .946           | .948           | .949     |
| P3              | .570           | .857           | .824           | .593           | .864     | .850     |
| P4              | .549           | .850           | .823           | .626           | .875     | .847     |
TABLE III: Individual performance comparison with adding audio feedback cues. The subscript \( m \) represents micro-averaging, whereas \( M \) means macro-averaging.

<table>
<thead>
<tr>
<th></th>
<th>Without Audio Cue</th>
<th>With Audio Cue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( Acc_m )</td>
<td>( Acc_M )</td>
</tr>
<tr>
<td>P1</td>
<td>.839</td>
<td>.946</td>
</tr>
<tr>
<td>P3</td>
<td>.593</td>
<td>.864</td>
</tr>
<tr>
<td>Avg.</td>
<td>.751</td>
<td>.917</td>
</tr>
</tbody>
</table>

VI. SCALABILITY AND CROSS-PLATFORMS

A benefit of using machine learning instead of hard-coding the rules is providing the designer the ability to scale the number of states up or down and to use the system on multiple platforms easily. For instance, a designer could design a simple two state interaction, where the robot only understands yes and no; or scale it up until they are happy with the number of options. To demonstrate these benefits, we use our system with SVM as the learning method and ran additional tests with different restrictions on the Baxter robot. All the tests were run with a member of the research team who trained the system and executed a random test sequence of actions with audio cues after restarting the system. The classification result was then compared with the labeled ground truth. We explored four different possible configurations of the system and Baxter robot, which are: (1) 7 degree-of-freedom (DoF) arm without Feedback with 7 states, (2) Single DoF arm with 2 states, (3) Alternative configuration of 7 DoF arm with 6 states, (4) 2 DoF arm with 5 states.

1. **7 DoF Arm with 7 states without controller** — This configuration is similar to tests described in the previous section. However, instead of an active feedback PD controller that tries to return to the origin position. The controller maintains the maximum torque feedback, causing maximum stiffness at all joints. The arm was still able to move around by backdriving the actuators. The system still performs at a similar level as those without actuator feedback as shown in figure 4. However, the arm was harder to push around, which might be a bad experience for the user.

2. **Single DoF arm with 2 states** — In this configuration, the arm is restricted to a single joint as shown in figure 5. This was done by setting all the joints except the first joint to maximum stiffness and adding physical support right below the arm. As the arm is only limited in one dimension and we are only classifying two states (origin and forward), the system was able to classify states with an accuracy of 0.988 and F1-score of 0.985. This configuration might be useful in interactions where the users can interrupt the system or where they select items from two categories.

3. **Alt. configuration of 7 DoF arm with 6 states** — This configuration has the Baxter’s arm in a completely different configuration compared to the one used in the initial exploration. We also added the backward gesture and removed up from the list of gestures. The gestures picked by the member of the research team can be seen in figure 6(right). Even with the different configuration, the system was able to learn and classify different gesture states with an accuracy of 0.991 and F1-score of 0.992.
Fig. 7: C4: 2 DoF arm with 5 states. (right) Shows the robot’s starting position with the movement space of the two joints (shown in gray) and also the gestures (shown in yellow) that were trained in the system. (left) Confusion matrix of this configuration

<table>
<thead>
<tr>
<th></th>
<th>Acc_Micro</th>
<th>Acc_Macro</th>
<th>Pre_Macro</th>
<th>Rec_Macro</th>
<th>F1_Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4) 2 DoF arm with 5 states.</td>
<td>.986</td>
<td>.994</td>
<td>.992</td>
<td>.981</td>
<td>.987</td>
</tr>
</tbody>
</table>

VII. DISCUSSION

Laid out in the introduction, the motivation of our system is to explore alternative methods of input for HRI using a different modality. Here, we will talk about how social HRI could leverage this system.

A. Potential Use Cases

Example use cases for this approach include a simple yes/no interface, complex list navigation, and start/stop robot commands. The modality also creates new opportunities within existing applications. For example, gestures on the robot’s arm can be subtle and have potential as an authentication method where users move the arm around in a learned sequence to access sensitive data or call up their preferences.

B. Universal Design

The goal of the project is to support travelers who are blind or low vision and we need an alternative method for users with disabilities to communicate with the robot. As this method does not require speech interaction, it could also facilitate communication with people who have trouble speaking, those who use another language, and people who are deaf-blind. The method also has promise for users without disabilities who are experiencing a situation-induced disability due to their environment or task. For example, letting go of a robot underwater could be problematic and SCUBA equipment can prevent speech. Physical gestures also allow users to input commands without interrupting their task – a nurse using a robotic transfer system can talk to the patient without confusing the speech based robot. Therefore, there is value in this type of interaction for a wide range of users.

C. Implementation Benefits

While attached hardware like joysticks or touchscreens can have similar functionality, our method reuses existing hardware without loss of the original functionality. Interaction can also be designed where the output (actions) and inputs (our system) work in collaboration. This lack of additional hardware reduces the need for added circuitry and interface surfaces, thus benefiting small form factor and low cost robots.

VIII. LIMITATIONS AND FUTURE WORK

While we have analytic results showing the feasibility of the method, we have yet to conduct an extensive user test that evaluates usability in realistic scenarios. Future studies will explore usability in more depth and compare our method with existing input devices. The main goal of this paper is to provide motivation, describe an initial system design, demonstrate system performance, and show potential for generalizability, and focus the approach in advance of detailed evaluation. This research pathway is recommended when maturing new HRI systems.

Furthermore, we only explored a few ways to use the robot’s arm as an input device – there remain many features and functions to explore in the design space. High on our list is to classify time-series gestures, such as drawing a circle, and explore vibrating motions in place of audio feedback.

IX. CONCLUSION

In this paper, we demonstrated a novel interaction technique for a robotic system through movement of the end-effector by the user. The approach is highly customizable and can learn user-defined gestures through support vector machine. Our preliminary exploration showed that our method was able to classify gestures from four different users with average accuracy of 0.965 and precision/recall of 0.894/0.903 after being trained on individual data. Our results demonstrated the feasibility of a direct haptic communication between human and robots, opening the possibility for new types of interaction design and leading the way towards creating an HRI method that has potential for a wide range of users.

ACKNOWLEDGMENTS

This work was funded by a grant (National Robotics Initiative, IIS-1317989) from the National Science Foundation. The authors also thank the participants in the user studies that informed this work and Ermine Teves for her assistance.

REFERENCES


