# A User Interface for Robotic Fixturing

Jon DeCastro

*Abstract*—**The goal of this project is to create a robot that is capable of fixturing a workpiece such that a person can easily work with the piece and make any adjustments, as necessary. My portion of the project involves developing a user interface that translates natural spoken language into unambiguous robot commands. The robot should be able to respond to commands from the human in the form of gesture and voice commands. The object is twofold: 1) understanding voice commands by translating those into robot actions; and 2) fusing together voice commands with gestures to give the robot a definitive position on the workpiece to grasp.**

## I. Introduction

Effective practical robots need to work alongside humans, helping them perform tasks that are difficult or impossible to perform without the help of another individual or robot. Several works have focused on providing high-level commands to the robot with the expectation that the robot carry out tasks in a correct manner. For example, Tellex, et. al. [4] demonstrate using natural language in an autonomous forklift scenario in which the forklift is required to "Go to the first crate on the left and pick it up" and "put the box on the pallet beside the truck." The authors introduce a probabilistic model known as a Generalized Grounding Graph to instantiate a command based on its semantic structure. The authors in [3] use parse trees and environment models to infer the meaning of probabilistic graphical models, exploiting learning to make the graph search more efficient. The authors of [2] combine speech and gesture to understand a user's task specification, where the system asks clarifying questions to resolve uncertainty.

Workpiece fixturing is another concrete example of robotic assistance. In this scenario, the requirement is for the robot to hold a workpiece in such a way that the human can work on it, allowing the object to be re-positioned as necessary. This is a complicated task, because the robot needs to interpret and understand intent such that it reacts according to some given human commands. For instance, "Hold this workpiece" is one command that could be issued by the user. The robot will be expected to be issued a specific command that, in turn, activates controllers that first move robot close to the workpiece, localize the object, then grasp the piece. An effective voice-activated user interface (UI) will be an essential component to performing the fixturing task. To be effective, several commands will need to be parsed by the UI as commands and location descriptions.

The ultimate goal of the UI is to be able to interpret a multitude of sensory information and be able to fulfill more specific requirements of the task. For example, the user might want to work on some part of the piece and issue the command: "grasp this end of the board" while the human points to some part of the object in the camara's field of view. I will be exploring this challenge later in the project.

## II. Proposed Solution

Developing an effective UI will require some work in natural language understanding and sentence parsing.

### *A. Natural Language Understanding*

The UI will take as an input a command expressed in natural language and output a binary variable representing a known command for the robot. At its most sophisticated, this involves automatically establishing groundings between words and objects in the world (in our case limited to the view of the Kinect sensor). We will need to annotate the board itself, along with possibly the user's hand holding the board. Since the goal is to have grounding on a continuum of board locations, we may also ground specific parts of the board. At first, there will be a finite number of possible grip locations that will need grounding. For example, "Choose grip locations to the right of the board" should have a positive correlation to those grip locations that are on the right-side of the board in the board coordinate frame. The intent, however, is to extend this to a continuum of board locations, so I will explore extending this to a set of poses.

### *B. Commands*

The commands will be encoded into the UI. Table I lists a few commands that the interface will accommodate. We can add to this list as we move further along in the project.

### *C. Gesture-based Recognition*

In the event that the user has a specific intent in mind for grasping, it is useful to fuse the human's gesture instructions with his/her voice instructions. An instruction such as "grip the board at the left side," for example, should confine the possible grip locations to the left-hand side of the board.

Gesture-based instructions can further confine the grip locations. To accomplish the task of fusing voice instructions with gestures, I will explore a means for probabilistic fusion based on gesture information. To

TABLE I: A list of commands for encoding.

| Command | Description | Example natural language input |
|---|---|---|
| graspObj | General command to localize the object and grasp it | "Hold this IC board." |
| graspHere | Command to locate the grasp points with possible aid of visual cues (finger pointing) | "Grasp the board along the left-hand edge." |
| graspStiffness | Command that switches modes from compliant to stiff or vice-versa | "Hold the board in place." |
| stop | Stop moving | "Stop." |

perform data transfer between ROS nodes, a client will be used to get finger pose information from perception in order to apply weights to the set of possible grasp poses in board coordinates. Either the weights themselves will be published (to be subscribed to by Alex's grasping module) or a list of grasp poses will be generated that meet some threshold (a subset of the full set of grasp points).

Note that, although Ben's module will likely have a fixed set of grip locations as a fixed set of objects in the world, it is interesting from a research perspective to be able to establish grounding for a continuum of board locations (anywhere between a small portion of the board to the entire board). I intend on developing the UI module with this in mind. One possibility is to use, for example, a Gaussian sum representation of the groundings. For debugging, the UI could provide graphical feedback (a heatmap of most likely board locations). This can optionally be used as feedback (back-channel) to the user.

### D. Workplan

The proposed solution involves parsing a voice-based command into a valid command for the robot to act upon.

1) Create and test a stub for a ROS node that publishes fake robot commands and a client node that requests board locations and finger point pose (for gesture-based grounding).
2) Implement the four basic commands from Table I using a natural language interface engine, such as the Natural Language Toolkit (NLTK) [1]. The input device will be the keyboard initially.
3) Extend the functionality to include voice commands by way of, for example, PocketSphinx[1].
4) Optimize the language-based interface using a probabilistic model. Evaluate the number of sentences correctly correlated to the intended command.
5) **Stretch goal:** Implement gesture-based information gathered from Ben's perception module (via a client-service call) and either the weights or thresholded grip locations will be published.
6) Test the system thoroughly through unit testing and a full system test using the actual YouBot and Kinect sensor.

[1] http://cmusphinx.sourceforge.net/

## References


[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009. ISBN 0596516495, 9780596516499.

[2] Jonathan Connell, Etienne Marcheret, Sharath Pankanti, Michiharu Kudoh, and Risa Nishiyama. An extensible language interfacefor robot manipulation. In Joscha Bach, Ben Goertzel, and Matthew Ikl, editors, *Artificial General Intelligence*, volume 7716 of *Lecture Notes in Computer Science*, pages 21–30. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35505-9. doi: 10.1007/978-3-642-35506-6_3. URL http://dx.doi.org/10.1007/978-3-642-35506-6_3.

[3] T.M. Howard, I. Chung, O. Propp, M.R. Walter, and N. Roy. Efficient natural language interfaces for assistive robots. In *Workshop on Rehabilitation and Assistive Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[4] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth J. Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011. URL http://dblp.uni-trier.de/db/conf/aaai/aaai2011.html#TellexKDWBTR11.