CAD-AR: An Intuitive Robotic Teaching Pendant for Skill-based Industrial Robot Programming

Sarah Li[1], Cole Shing[2], Yvonne Coady[3], H.F. Machiel Van der Loos[4], Elizabeth Croft[5]

***Abstract*— Traditional industrial robot programming methodologies are often time consuming, non-intuitive, and visually non-collocated with user's intended task. These factors hinder the adoption of collaborative robots that supports the new transformable production paradigm for smart factories. We outline a pilot user study for a multimodal robot programming system that allows intuitive motion planning, offers safety previews, and achieves better efficiency.**

Our robot programming environment builds on skill-based task programming, and is augmented with gesture recognition, speech recognition, immersive 3D object manipulation, and Augmented Reality (AR) simulations. In our pilot study, we found that gesture recognition and virtual fidelity to the real robot are crucial to developing a usable and accurate programming system.

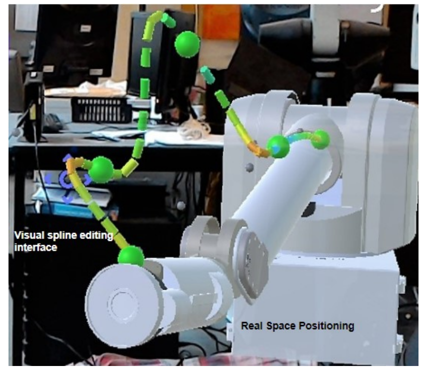

Fig. 1. An important part of robotic programming is trajectory editing. The user may edit the position (green) via points or the path direction (grey). Path colour indicates the velocity of the trajectory along the path.

I. INTRODUCTION

Industrial robotics has revolutionized the manufacturing industry, and the introduction of collaborative robotics is the necessary next step in this process. Along with the expansion of robotic applications and functional complexity, there is a rising demand for reprogrammable and redeployable robots [1]. To achieve a quickly transformable production environment, an intuitive robot command interface is the key to introducing flexibility.

Our research is focused on developing an interaction environment that can offer the expertise of Computer Aided Design (CAD) to simulate predicted motions and precisely define complex spatial relationships, without requiring experience. To simplify the input interface for non-designers, we shift the method of programming from robot kinematics to pre-programmable skill primitives [2].

II. MULTIMODAL CAD-AR ROBOT PROGRAMMING INTERFACE

Our proposed multimodal AR, CAD inspired, robot programming system is for operators with no CAD background. Thus, we must maximize intuition and accuracy. Our objective is to combine the interactive aspect of traditional robot teaching with CAD's precision and preview abilities, using AR as a platform.

As a first step, we have built a proof of concept prototype to explore this interaction modality. This paper will present some of our initial findings and challenges.

A. System Description

The CAD-AR robotic teaching interface is implemented using a Microsoft Hololens[1] and a 7 degrees of freedom Barrett WAM$^{\text{TM}}$ robotic arm[2] (herein referred to as "the WAM"). Software for the Hololens is developed using Unity and Microsoft Visual Studio in C#.

The WAM controller is a C++-based Robot Operating System[3] (ROS) package that communicates to the robotic arm via CANbus. It provides positional commands and queries to the WAM in both Cartesian space and joint space. Communication between the controller and Hololens is achieved using ROSbridge [3] in Unity and Linux.

B. Proposed Workflow

The pilot study uses the following speech-based workflow:

1) **"Come here"**: the virtual WAM moves towards relative position of user and generates via points (cubic spline)
2) **"Try again"**: resimulate the WAM motion
3) **"Edit"**: modify via points (see Fig.2 left)
4) **"Do it"**: the real WAM receives position updates at 60 Hz to achieve virtual trajectory (see Fig.2 right)
5) **"Reset"**: move the real WAM back to initial position

[1]Sarah Li is with Department of Aeronautics & Astronautics, University of Washington, Seattle, WA 98195, USA sarahli@uw.edu

[2]Cole Shing is with the Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada coleshin@cs.ubc.ca

[3]Yvonne Coady is with the Faculty of Computer Science, University of Victoria, Victoria, BC V8P 5C2, Canada ycoady@uvic.ca

[4]H.F. Machiel Van der Loos is with the Faculty of Mechanical Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada vdl1@mail.ubc.ca

[5]Elizabeth Croft is with the Faculty of Mechanical Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada Elizabeth.croft@ubc.ca

[1]https://www.microsoft.com/en-ca/hololens

[2]http://www.barrett.com/products-arm.htm

[3]http://www.ros.org/

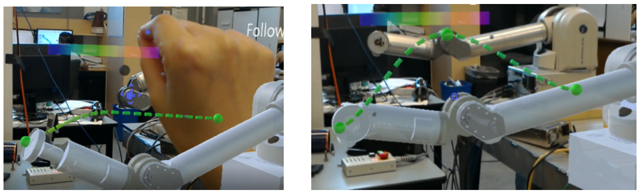

Fig. 2. Proposed workflow: Hand controlled path editing(left), and physical arm following virtual trajectory(right)

A preliminary study was done on the prototype with a small group of five participants. Two participants have used the Hololens daily. The others became acquainted with the AR environment by first learning gestures through Hololens tutorials.

C. Results

All participants completed the proposed workflow and found the system intuitive. Two participants had problems completing the path-editing process due to difficulties with gesture recognition, while all participants agreed that manipulation was the most physically error-prone task. User confidence and efficiency increased with each repeated trial.

1) Gesture Recognition: As [4] has also noted, we observed that users naturally reach forward at holographic targets, extending the index finger perpendicularly outwards. This indicates that the 2D vision based recognition algorithm used by Hololens cannot always capture the editing gesture. The more experienced users did not have this problem, indicating that users can adapt after extensive training.

2) Manipulation: All participants had problems manipulating via points. During manipulation, Hololens uses programmer's centre of gaze as the origin and tracks the hand position. Participants found it difficult to keep head and hand completely still. Participants also reported slight fatigue in their arms after about 10 minutes of study.

During manipulation, participants tended to drag points beyond gesture boundaries. The gesture recognition space is constrained to the middle of the user's view, and is narrow compared to the hand's reachable space. All participants expressed frustration at the limitation in hand motion.

Lastly, two participants mentioned a lack of spatial awareness for via point manipulation. In particular, they asked for tangible feedback such as a physical object to grasp or a resisting force, instead of solely relying on holographic feedback.

3) Speech Recognition: All participants found speech input to be natural and intuitive. One participant reported that the robotic arm felt "approachable" and "friendly". Participants also reported similar ease of usage with a 2D, holographic, and selectable command menu. This suggests that a holographic menu is a potential replacement for speech input in noisier environment.

4) Fidelity: Three participants reported doubts about the consistency between the real and virtual trajectories. One suggested moving the spline editing interface from the virtual to the real WAM, such that the user can see the real end-effector moving through the virtual path.

In only specifying position, the velocity is left unconstrained. As a result, the trajectory planning will occasionally generate jerky, behaviour in attempting to hit all of the via points. This behaviour is not reflected in simulation. Participants reported this phenomenon as a factor causing doubts in the fidelity of the robot.

III. Next Steps

The following solutions are proposed to address the issues found in pilot study.

Physical motion sensor: A clickable hand-held motion sensor has been proposed to replace the existing gesture manipulation system. We hope this artifact will address limitations of 2D vision based recognition, lack of tangible feedback and physical stress from extended use. Due to common usage of a computer mouse, a clicker is likely to be intuitive in 3D as well.

Velocity-based path control: To alleviate the jerky movements, velocity updates has been proposed to replace position updates. By controlling the time derivative, a smoother motion can be achieved, but discrepancies between predicted position and actual position may occur. A feedback control loop on position values could remediate fidelity issues.

Further CAD integration: Finally, we plan to integrate more CAD concepts and investigate their benefits. A possible concept is the plane/point mating constraint method that may be useful for certain industrial applications [5].

IV. Conclusion and Future Work

Our exploratory study highlights the ways in which our proposed trajectory planning interface may be intuitive and efficient for non-expert operators. The preliminary study indicates that despite its limitations, CAD-AR is potentially well-suited for industrial robot programming. Further studies and development are required for a thorough implementation.

References

[1] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87–94, 2012.

[2] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.

[3] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Robotics Research*. Springer, 2017, pp. 493–504.

[4] A. Millette and M. J. McGuffin, "Dualcad: integrating augmented reality with a desktop gui and smartphone interaction," in *Mixed and Augmented Reality (ISMAR-Adjunct), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 21–26.

[5] N. Somani, A. Gaschler, M. Rickert, A. Perzylo, and A. Knoll, "Constraint-based task programming with cad semantics: From intuitive specification to real-time control," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2854–2859.