TWO-ARMED MOBILE ROBOT ASSISTED RETRIEVAL OF TARGETS FROM HIGH RADIATION AREAS

Dominik Bartsch
University of Notre Dame
dbartsch@nd.edu

Jerry Nolen
Argonne National Laboratory Lemont, IL
nolen@anl.gov

Ravi Gampa
Argonne National Laboratory Lemont, IL
gampa@anl.gov

Doga Y. Ozgulbas
Argonne National Laboratory Lemont, IL
dozgulbas@anl.gov

Ryan D. Lewis
Argonne National Laboratory Lemont, IL
ryan.lewis@anl.gov

ABSTRACT

We present proof-of-concept demonstrations for a robotic retrieval of targets from a beamline soon after the end of an irradiation run. The implementation of such robots will significantly reduce the dose received by human technicians who must currently wait until radiation reaches safe levels and enter such areas to retrieve targets. The study shows that tele-robotic retrieval of isotopes is now within the realm of possibility for state-of-the-art robots provided that the robot is trained precisely to perform the various subtasks involved and the hardware is adapted to be robot friendly.

I. BACKGROUND

Medical radioisotope production often involves irradiation of target material with a particle or photon beam to induce nuclear reactions. Examples include ^{68}Zn(γ, p)^{67}Cu, ^{48}Ti(γ, p)^{47}Sc and ^{156}Gd(p, 2n)^{155}Tb.

Retrieval of irradiated targets is delayed until radiation levels fall below DOE-designated threshold. Robots tolerate higher doses than humans, so they can enter such areas earlier than technicians. A robot would have to:

- navigate into the production vault.
- close the gate valves of the vacuum chamber.
- vent the target module to atmosphere.
- unplug coolant connections and undo bolts.
- deposit and secure a target in a heavily shielded cask that it then tows out of the target area.

Using a robot would enable earlier target retrieval, allowing for targets with much higher activity and lowering exposure for technicians handling the target.

Previous iterations of the robot work cell involved demonstrating the viability of using robot arms on mobile bases to perform hose transfers and docking pin transfers. These are important tasks as:

- target modules use integrated water-cooled heat exchangers with quick-connect hoses that supply the coolant.
- docking pins are used to secure the movable base to the workstation while the arms perform other tasks.

These earlier iterations provided standalone demonstrations of subtasks performed by multiple single-armed robots. The current work integrates the activities onto a single unit with workflows that seamlessly proceed through subtasks. Additionally, the robot communicates using full wireless connectivity and control, eliminating wired ethernet connections in line with the expectation for the eventual application.

II. EXISTING ROBOT SOFTWARE AND PYTHON-BASED MODULES

The current mobile robot work cell is composed of a self-driving base and two arms. The base, an MiR 250, was fed scripts through a ROS-based (Robot Operating System) Graphical User Interface (GUI) which was accessed by connecting to the robot via a browser on a device connected to the same local network.

The tele-operation of the two Universal Robots UR5e and UR16e robotic arms attached to the base is less straight-forward. In practical application, these arms are programmed to communicate with a specific tablet connected via data cable. These robot arms can remotely execute programs saved to a .urp file (a custom script developed by UR) using the now defunct Python URX module.

The Self Driving Laboratories initiative at Argonne National Lab has since developed its own open-source software that enabled tele-operation of UR arms using only Python scripts. The software used for these tasks is under the **isotop** branch of the **ur_module** Python package at the link https://github.com/AD-SDL/ur_module.git

III. ROBOTIC SYSTEM INTEGRATION AND DEMONSTRATED TASKS

The setup consists of the movable base with an extruded aluminum frame on top and two robotic arms mounted at 45°angles to the vertical on an extruded A-frame attached to the extruded aluminum frame. The arms are adjacent to one another at the center of the base's attached frame (see **Figure 1.a**). The base is the MiR250 while the arms themselves are Universal Robots' UR5e and UR16e robots. Note that the numbers in the name of each of the three devices corresponds to their maximum payload in kg.

An electrical panel board has been installed to safely provide power to the required onboard wireless router with fuses providing overcurrent protection. To power the arms, a two-way splitter from the single 48V output of the MiR places the two arms electrically in parallel and a key switch was implemented to delay the power-up of the UR16e until the 5e was fully powered on, as the power required for a simultaneous start up of two arms is too great for the MiR to output.

To control and communicate with all three subunits of the robot (The MiR base and both arms) at the same time fror
DC converter and configured to act as a repeater of the closed network at Argonne.

The actual tasks accomplished by this work cell include the transfers of a custom-engineered docking pin (see **F**
python script was developed using the **transfer** function from the **UR** library in the **ur_module** Python package. This

- home position (an arbitrary joint location for the robot to move it out of the way while the base moves).
- source position (the TCP position where the pin starts out).
- target position (the TCP position where the pin is intended to be placed).
- approach axis and distance for the source and target locations (so the pin moves linearly parallel to one of the Cart

Due to the inherent bimanual nature of the hose transfer task, the operation breaks up the order of the original **tran**
be temporarily interrupted while it waits for the other arm to approach and pull down on a spring-loaded fitting to rele hold the hose in place. These functions require the arguments:

- home, source, and target positions for both arms.
- source and target approach distances and axes for both arms.
- depth (the distance which the one arm had to move the quick-connect piece downwards in the hose securing mec

The user "trains" the robot's positions (i.e. home, source, target) by placing the UR arms in a Freedrive mode (a m
resistance, and the user can position the arm into a desired position) and then copying the position of the arm.

The home location is fundamentally different from the source and target locations as it is a joint location (a list of six
of six floats that represent the linear point in space of the tool head and its rotation about each of the cartesian ax
"homes" itself (moves to the home position) but still has a linear approach to the source or target locations. The pytho

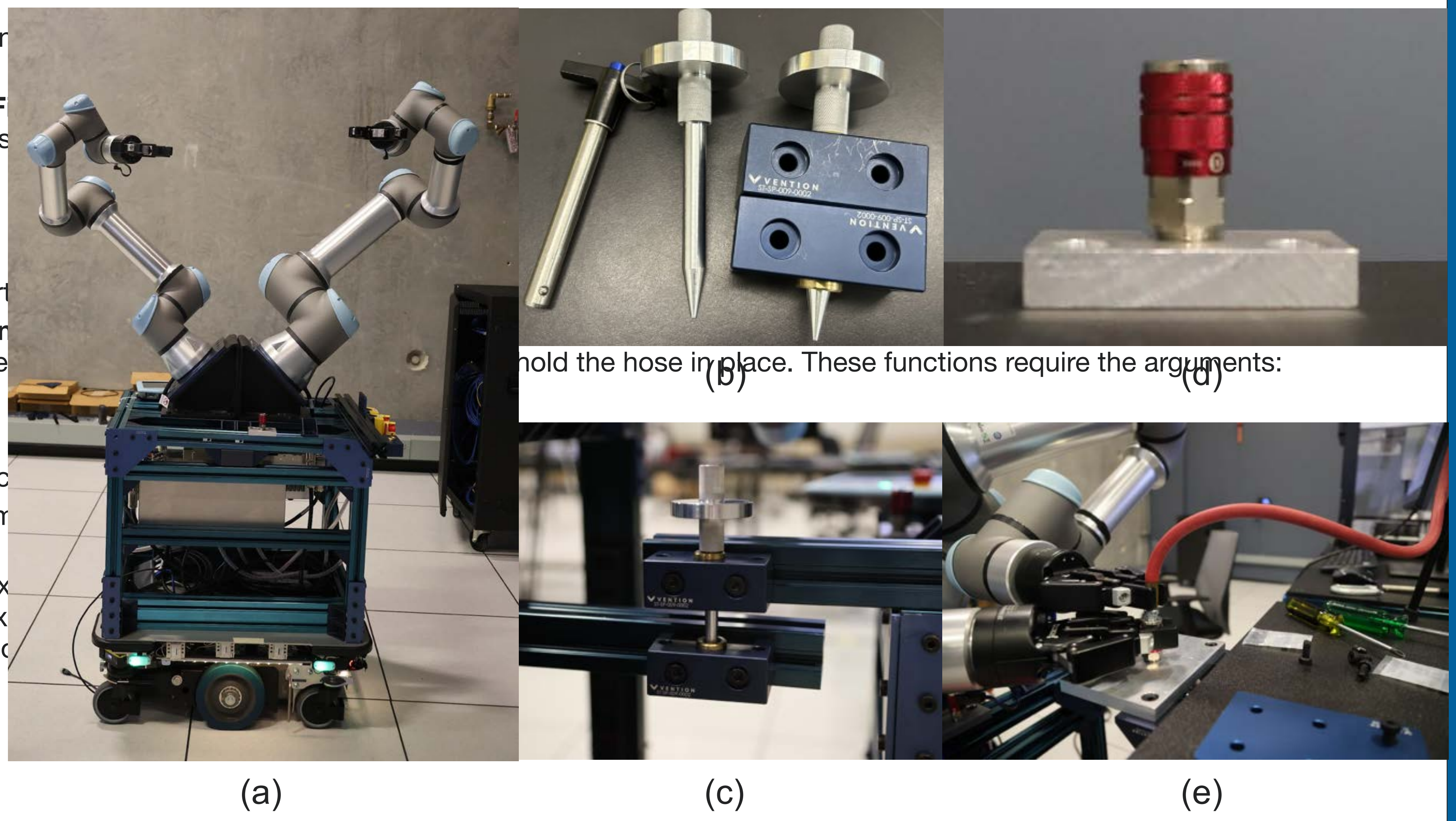

Figure 1. Photographs from testing of the mobile robot work cell. **(a)** Assembled mobile robot work cell. **(b)** Docker pins on display. **(c)** Docker pin attaching the frame of the work cell to the frame of the workstation. **(d)** Quick-Connect Hose dock. **(e)** Robots working in tandem to release the quick-connect hose fitting.

IV. RESULTS AND FUTURE ITERATIONS

Once the modifications to the python package were made, the programming could be completed with relative ease. The script for the operation of the arms was relatively short, containing only declarations for the positions (as lists of six floating point numbers), declaration of objects of the UR class for each of the arms, the transfer function, and closing the connection between the remote PC and the robot.

With the robot now being fully integrated on one MiR base, the next step in this project would likely be looking at other functions the robot would likely have to perform, like loosening vacuum connections. The existing method of sealing the conflat (CF) vacuum connections secured by many large bolts seems incompatible with a robot setup. Therefore, the design of the target assembly will be adapted to make it robot-friendly. Another thing to do would be to replicate the setup in LEAF's target room to map the navigation of the MiR base about the target room.

V. ACKNOWLEDGMENTS

We appreciate the entire RPL team for their development of the **ur_module** and **madsci** python packages, on which the scripts for the UR module control was built. These modules are open-source packages available on GitHub at github.com/AD-SDL.

This research was supported by the U.S. Department of Energy Isotope Program, managed by the Office of Science for Isotope R&D and Production and Argonne National Laboratory under U.S. Department of Energy contract DE-AC02-06CH11357. This work is supported in part by the Horizon-broadening Isotope Production Pipeline Opportunities (HIPPO) program, under Grant DE-SC0022550 from the Department of Energy's Isotope R&D and Production Program.

Summer 2025

Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.