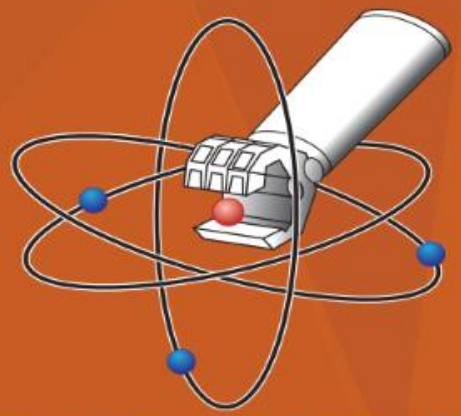
The University of Texas at Austin
WHAT STARTS HERE CHANGES THE WORLD

Design of a Multiuse Workcell and Incorporation of the Descartes Package

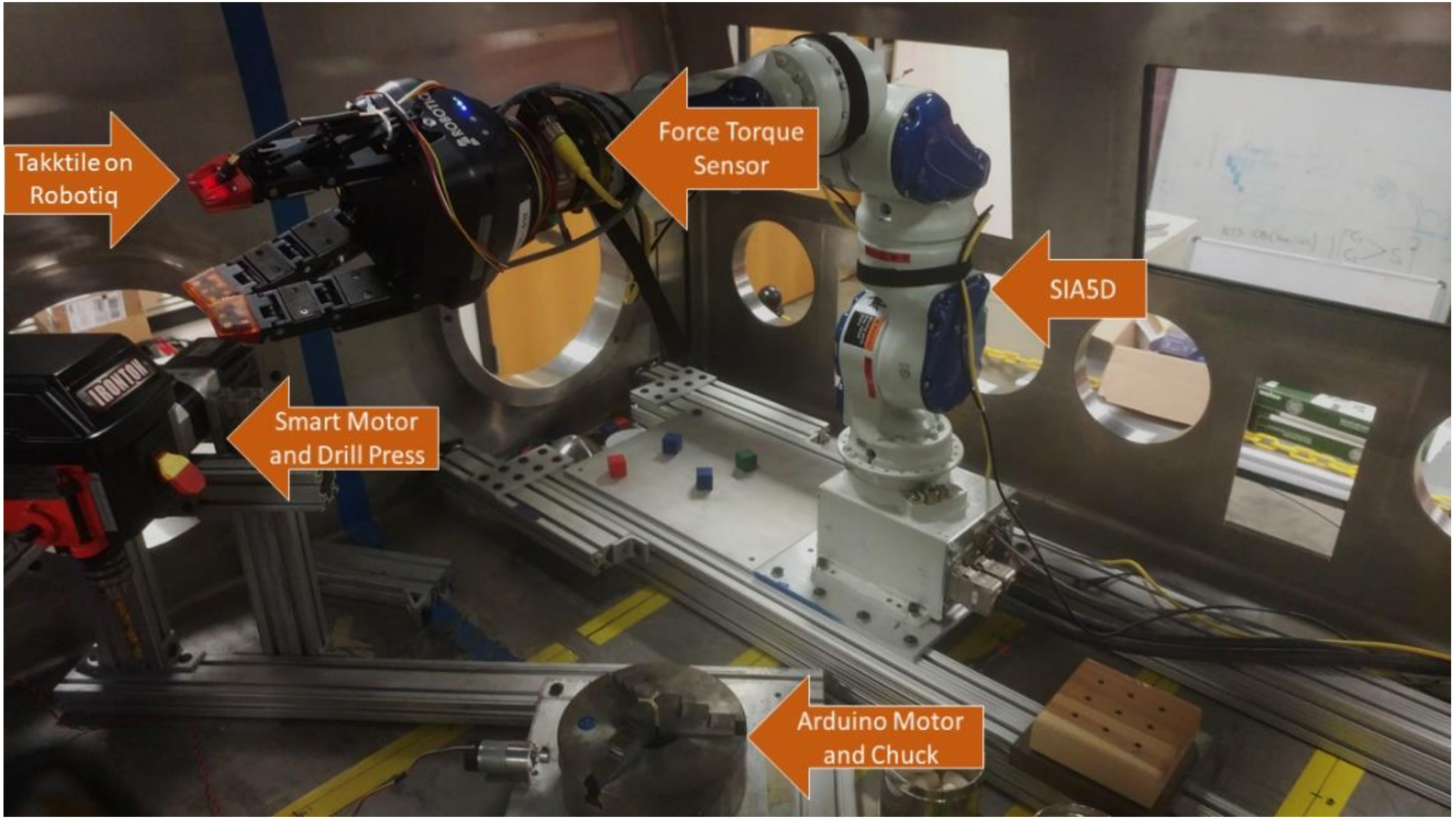
Christina Petlowany



NUCLEAR AND APPLIED ROBOTICS GROUP

Outline

- Multiuse workcell
  - Background
  - Video
  - System requirements
  - Tasks
  - Results
- Descartes


Takktile on Robotiq
Force Torque Sensor
SIA5D
Smart Motor and Drill Press
Arduino Motor and Chuck

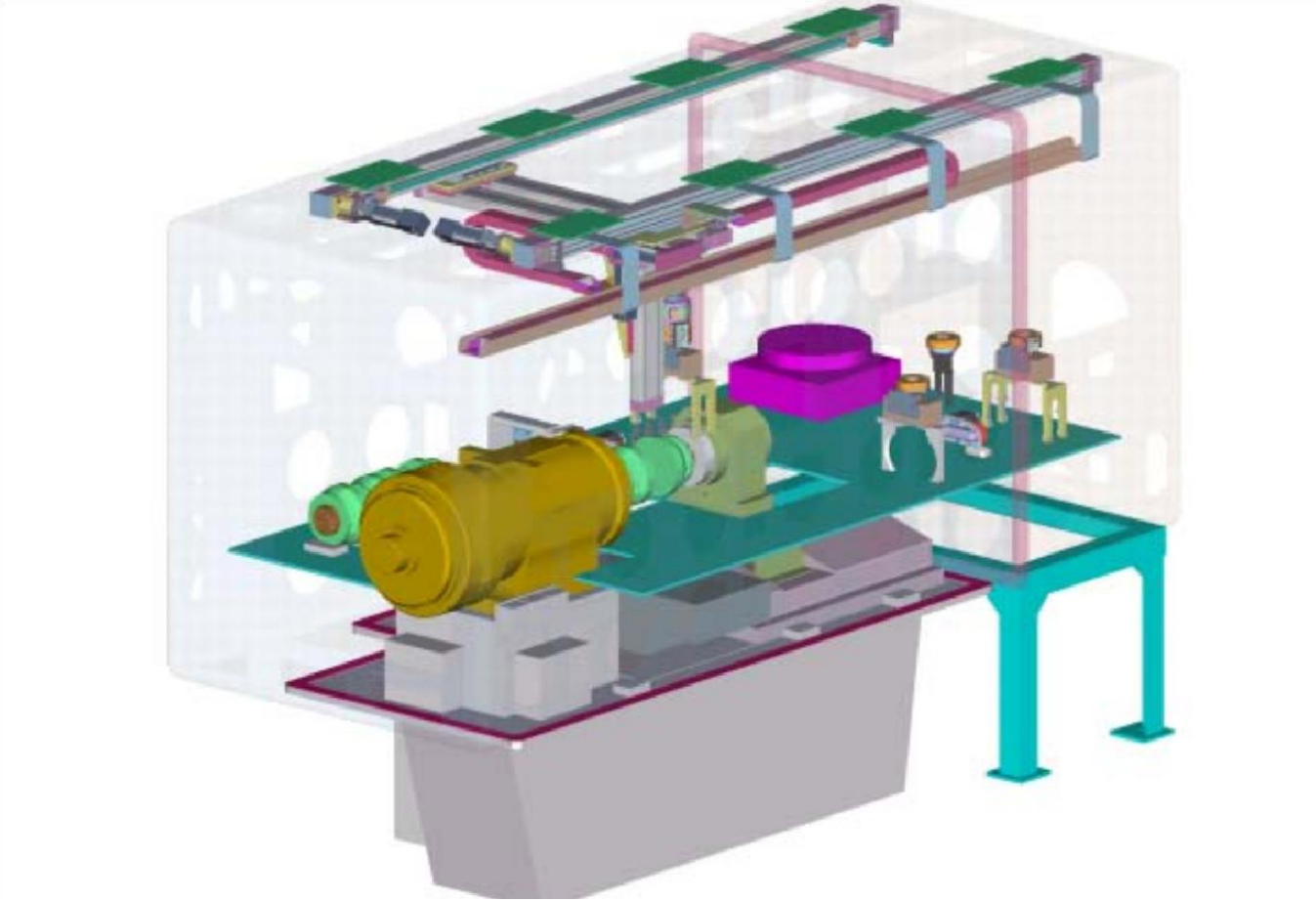
Background

- Nuclear and Applied Robotics group partnership with Los Alamos National Labs
- Use of gloveboxes to protect workers from radiation
- Gloveboxes lead to ergonomic, safety concerns

Rentetzi, M. (2017). Determining Nuclear Fingerprints: Glove Boxes, Radiation Protection, and the International Atomic Energy Agency. *Endeavour*, *41*(2), 39-50.

Background Continued

- ARIES glovebox at LANL
- Move towards fixture-free, adjustable workcell design



D. Nelson B. Santistevan W. Brown P. C. Pittman, T. Staab. Automation of the lanl aries lathe glovebox. In *ANS Ninth Topical Meeting on Robotics and Remote Systems*, number LA-UR-01-0313. Los Alamos National Laboratory (LANL), 2001.


Bowl Demo
2x

System Requirements

- Environment
  - Radiation resistant
  - Confined area
- Serial Manipulator
  - High manipulability
  - Force sensing capabilities
- Gripper
  - Ability to use tools
  - Can handle uncertainty in object location
  - Can place objects on a 3-finger chuck
  - Compatibility with delicate materials
- Vision
  - High degree of accuracy
  - Robust support

Hardware Platform

- Hardware components
  - 7-DOF robotic arm (Motoman SIA5D with Agile Planet AX Controller)
  - 3-finger adaptive gripper (Robotiq)
  - ROS-integrated drill press (Ironton 8 in. Bench Mount Drill Press)

- Sensor components
  - Force/torque sensor (ATI Gamma)
  - Gripper surface pressure sensors (TakkTile Robotiq Kit)
  - Depth camera (ASUS Xtion PRO Live)

Automated Workcell

Input: Detected Objects
Output: Task Completion
Task Identification and Execution
Sorting Task
Drill and Press-fit Task
Descartes Inspection Task

Vision Classifiers - ORP

Input: Camera Data
Output: Detected Objects
Responsibilities
RGB Classifier – detect different colored objects
Peg Classifier – Use edge detection to detect peg location and orientation
General Object Classifier – detect object type and location

Contact-Controlled Movements

Input: Current Task
Output: Force-controlled motion
Responsibilities
Control of the manipulator to ensure objects are placed with minimal impulses.
Control of the press-fit task.

General Manipulation and Motion Planning

Input: Detected Object
Output: Manipulation of object to next task area
Responsibilities
Keep manipulated objects in the Collision Scene
Pick up and place objects where they are needed
Verify objects were picked up safely

Gripper Interface

Input: Current Task
Output: Robotiq Manipulation
Responsibilities
Update gripper parameters (force and grasp type) for each type of object

Path-Controlled Movements

Input: Current Task
Output: Path-constrained motion
Responsibilities
Given object location, plan a continuous path around the object to ensure object is inspected completely.

Peripheral Integration

- Animatics SmartMotor to move drill
- Timer, current-controlled chuck
- Development of peripheral architecture for ROS integration is in progress
  - Break down complex parts into components (actuators, etc.) with congruent functions that utilize similar code

Supervisory System

```
1 notify system is ready
2 while(object list is not empty):
3  detect objects
4  classify objects by type of task
5  make sure extra items are not included
6  try
7    switch object set
8       disk and peg: Drill and Press-fit task
9       bowl: Laser Ablation/Descartes task
10        cubes: Sorting Task
11  catch
12    if(task succeeded)
13       wait for more object sets
14       continue loop
15    else
16       replace objects onto tray if possible
17       ask for human intervention
18  wait for user confirmation to restart
19 end
```

Sorting Task

1 **receive execution request**
2 while(cube object list is not empty):
3 try
4 move over object
5 grasp object
6 **2-stage verify grasp**
7 lift object
8 move over appropriate receptacle
9 release object
10 remove object from list //success, next
11 catch
12 if(object grasped):
13 put object back
14 remove object from list //failure, skip
15 move to home
16 end
17 **return control to main node**

Bowl and Disk Demo

1 **receive execution request**
2 while(hemisphere/disk object list is not empty):
3 try
4 move over object
5 grasp object
6 **2-stage verify grasp**
7 lift object
8 move over chuck
9 force move down into chuck
10 **Arduino**: secure object in chuck
11 if (hemisphere)
12 **Arduino**: release chuck
13 move hemisphere to tray
14 force move down to tray
15 perform surface finish task
16 end
17 if (disk)
18 alert user to move chuck to drill
19 **Arduino**: drill hole in disk
20 alert user to move chuck back to home
21 grab peg
22 perform press-fit
23 **Arduino**: release chuck
24 move disk back to tray
25 force move down to tray
26 catch
27 if(task succeeded)
28 wait for more object sets
29 continue loop
30 else
31 replace objects onto tray if possible
32 ask for human intervention
33 wait for user confirmation to restart
34 end

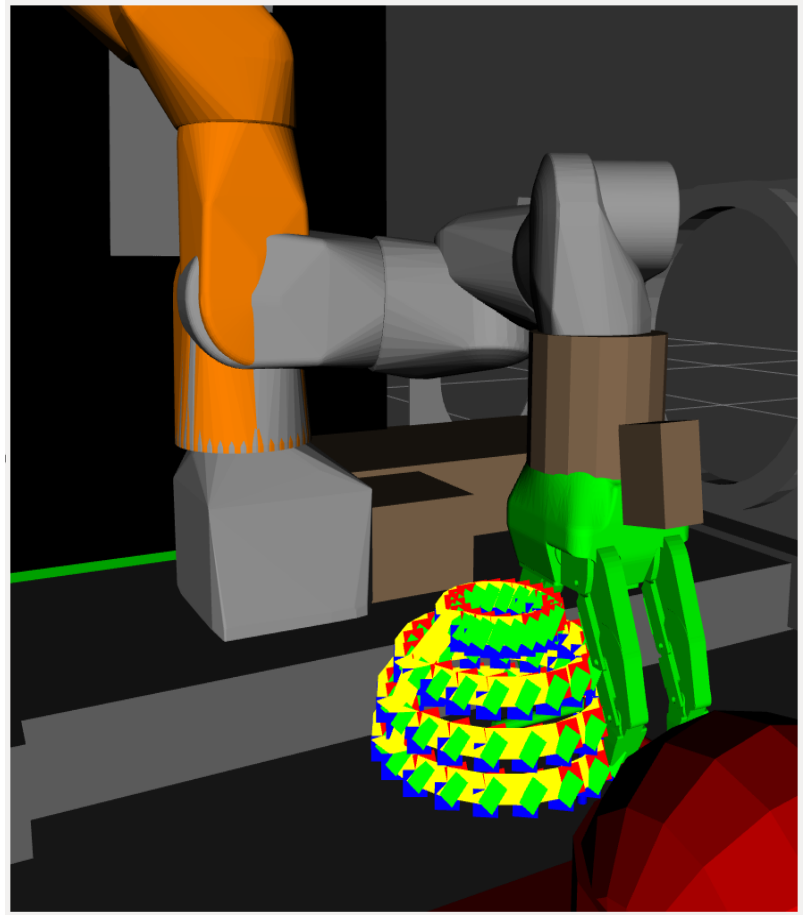
Results

Metrics	Attempts	Successes	Failures	Average Task Completion Time (s)
Sorting	23	22	1	280.2 ± 38.7
Descartes/Laser Ablation	5	5	0	353.4 ± 10.5
Cylinder Manipulation	5	4	1	276.4 ± 18.6
Automated Drilling	5	5	0	162.4 ± 5.3

-> Flexible, robust automation

Descartes

- Successes
  - Creates smooth paths quickly
  - Gives the user lots of control
  - Relatively minimal learning curve
- Limitations
  - No identification of failure points
  - Can collide with objects in between planned points during joint reconfigurations
  - Still experimental



Descartes Continued

- Used to build off Andrew Sharp's work with Virtual Fixtures
- Master's thesis – design tool for reachability visualization for complex paths