Remote Operation of 12-ID SAXS/GISAXS with UR3 robot

BYEONGDU LEE

CMS GROUP
User flow chart

1. Submit Proposal
2. Request Sample Plates
3. Place Samples & Ship to APS
4. Submit sample info & ESAF
5. Remote Experiment
6. Discuss Plans
7. Send Sample Plates
8. Send Schedule & Load samples
9. Add user to DELOS list
10. Dispose samples

User
Staff
12-ID Remote Experiment System

1. Remote operation
2. Mail-in measurement

12ID Database (MySQL)

- Access Interface
  - Python (pyQt)
  - PHP
  - Registration
  - Request

- 12ID Remote Experiment System
- Python (pyQt)
  - Database (MySQL)
  - Interface

- Storage
  - PyEPICS
  - SPECclient
  - Open-cv, Qrcode, ...

- UR3 Robot
- Link
- Shipping

- RF Link
- Virtual Beamline

- Beamline Control
  - Byungdu Lee
  - Alexis Quental

- Standard Sample Plates
  - Chuck Kurtz
  - Soenke Seifert
  - Byungdu Lee
  - Xiaobing Zuo
  - Alexis Quental

- Others
  - Capillary
  - Solid
  - Biosamples
  - GISAXS
  - Static
  - Capillary
  - Temp
  - DSC
  - Bio
  - Liquid Autosampler

- Xiaobing Zuo
- Sung Sik Lee
- Byeongdu Lee
- Alexis Quental
- Chuck Kurtz
- Soenke Seifert
- Xiaobing Zuo
- Ivan Kuzumenko
Database

Mysql (Xiaobing, Byeongdu)

- Tables
  - frames, frame_items, order, order_items, users, order_statuses
  - These are constructed from an example for a store that needs to manage product inventories, orders, and shipping.

- Can be easily communicated with python and PHP.

- Running on a PC at the beamline
Web Interface

PHP programming (Xiaobing)

- Portal to communicate with 12ID Remote Operation Database
- For Users
  - User registration
    - Not linked to APS DB
    - User ID: email address
    - Other info to provide: mailing address and GUP number
  - Requesting sample plates
  - Registering sample information for the sample plates
- For staff
  - DB searching for statistics and management
  - Update status of the user’s sample plates
Sample Plate Shipping

PHP/Python codes for Dymo label printer

- Print shipping labels (user’s registered address)
- Scan QRcodes, generate web-links, and email to users
- Can check the status of a sample frame.
Sample Frame

3D printed (Chuck, Alexis, Soenke, Ben Reinhart)
Sample Plates

3D printed
Assembly of Sample Plate on Sample Frame

- Sample holder/Frame ID
- Sample holder frame
- Sample holder
- Magnetic mount
Robot
How to get started.
- https://youtu.be/nFP_z5l68_g
- So many YouTube videos.

Vendor’s online training courses ↦ Short but very informative
- https://www.universal-robots.com/academy/
Teach Pendant and Control box

Linux, Debian
Many control I/O
Communication: USB, Ethernet
Comes with python2.7
Definition
TCP (Tool Center Point)
Forward and Inverse Kinematics

UR3: \( q: q_1, q_2, ..., q_6 \)  
\( p: X, Y, Z, ax, ay, az \)  
TCP position and orientation.
Coordinate definition
How to program on the teach pendant

- Free drive to a position A
- Free drive to a position B
- Decide how you like to move from A to B
  - Movej: move minimum number of joints simultaneously
  - Movel
  - Movec
  - Movep
- If you want to wait for a signal, define the signal too.
- Save and run
- The graphical program will be saved as a URscript:
  - /data/program/*.script
Teaching waypoints

A
moveJ
moveL
B
moveC
moveP
moveL
Advanced scripting

Program
Here you can program your robot to do tasks.

To program your robot, select the nodes from the Node List and they will appear on the Program Tree.

- Add Before Start Sequence
- Set Initial Variable Values
- Program Loops Forever
Having the Pendant screen on your PC.
Debian installation file can be obtained from the vendor (Tanaka).
Steps to do:
- Login to the control box using putty with an IP of 164.54.xxx.xxx.
- Run x11VNC
- Then on your windows computer, use a VNC client such as “VNC viewer” and connect to 164.54.xxx.xxx:5901
Operation of UR3

- GUI mode
- Script level (using UR Script)
- C-API level

1.5 Function

A function is declared as follows:

```python
def add(a, b):
    return a + b
end
```

The function can then be called like this:

```python
result = add(1, 4)
```

It is also possible to give function arguments default values:

```python
def add(a=0, b=0):
    return a + b
end
```

URScript also supports named parameters.
Interface for programming


- Primary/Secondary interfaces
  - Primary can send additional messages

- Real-time Interface

- RTDE interface

- Dashboard Server (controlled by sending simple commands to the GUI over a TCP/IP socket)

- Socket communication (UR robot became a client and Urscript provides socket commands)
  - For example, in polyscope, use ‘socket_open’,

- XML-RPC (to transfer structured data between programs over sockets)
  - For example, make the robot move using a pose retrieved from a remote camera. The remote camera program provides the next target pose based on the camera image analysis. The camera program can be python or C++.

Ideal for off-line programming. For example, to synchronize a real UR3 with a virtual one on a computer.
Python-URX

- Control the robot in REMOTE mode.
- Communicate through the real-time and secondary interface.
  - Main comm interface is the secondary.
  - Using real-time interface only for force sensing, which requires fast comm.
- https://github.com/SintefManufacturing/python-urx
  - Sending a UR script and execute
  - This has a script to control a “Robotiq Gripper”
UR3 programming

Python-urx

- Uses URscript
  - A command string
    - Ex) Set_gravity(5)
    - Ex) movej([1,1,1,1,1,1],a=1,v=1,r=1)
  - A full program
    - def myProg():
    - ....
    - end

- Support force mode and gripper.

class URRobot(object):
    
    ""
    Python interface to socket interface of UR robot.
    programs are send to port 30002
    data is read from secondary interface(10Hz?) and real-time interface(125Hz) (called Matlab interface in documentation)
    Since parsing the RT interface uses som CPU, and does not support all robots versions, it is disabled by default
    The RT interfaces is only used for the get_force related methods
    Rmq: A program sent to the robot i executed immendiatly and any running program is stopped
    ""

    def __init__(self, host, use_rt=False, use_simulation=False):
        self.logger = logging.getLogger("urx")
        self.host = host
        self.cs = None

        self.logger.debug("Opening secondary monitor socket")
        self.secmon = ursecmon.SecondaryMonitor(self.host, use_simulation) # data from robot at 10Hz

        self.rtmon = None
        if use_rt:
            self.rtmon = self.get_realtime_monitor()
            # precision of joint movem used to wait for move completion
            # the value must be conservative! otherwise we may wait forever
            self.joinEpsilon = 0.01
            # It seems URScript is limited in the character length of floats it accepts
            self.max_float_length = 6 # FIXME: check max length!!!
        self.secmon.wait() # make sure we get data from robot before letting clients access our methods
12-ID python codes

- Modification URX for 12IDB use
  - `/home/beams15/S12IDB/python_codes`
  - Enable force mode (This is the only part using RT).
  - Timeout time changed
  - The thread for “wait” is changed to prevent an infinite loop when a timeout occurs.

- 12ID code
  - Classes for basic operation: Robot12idb.py
  - Qt GUI : multiheaterWin2.py
  - Some other code to tweaking TCP (tweakRobot class in multiheaterWin2.py).
Remote Ops software (Python/pyQt5)

User can load their sample information that they have provided through the 12ID web. The loaded information goes onto this table and users can change as needed.

Users can perform various data acquisition scans for positioning or collecting scattering data either automatically or one at a time manually.

Menu to control the UR robot using Robot12idb.py

Users can choose a sample frame to load, number 0 to 19.
Virtual Environment (offline programming)

- CoppeliaSim
  - Young Soo Park (Robotics, AMD), Summer students (Rian Simpson)
- RoboDK
  - The free version allow only one robot or a translator.
- Visual Component (SolidWorks)
  - [https://www.youtube.com/watch?v=JVxYZbDpu_8&ab_channel=VisualComponents](https://www.youtube.com/watch?v=JVxYZbDpu_8&ab_channel=VisualComponents)
  - OPC UA server/client
- Conversion of CAD file to step files
  - Brian Rusthoven (Design & Draft), Summer students
  - Made a simplified drawing version (Chuck and Byeongdu)
Future direction

- Autonomous experiment (Weekend users)
  - Self-positioning the robot by camera image analysis coupled with AI
  - Data acquisition and screening

- Easier virtual environment generation
  - A software program to convert complex 3D CAD into a simplified 3D SAT file.