CanSAS Update Sample Environments

Nick Terrill (Diamond) & Katie Weigandt (NIST)

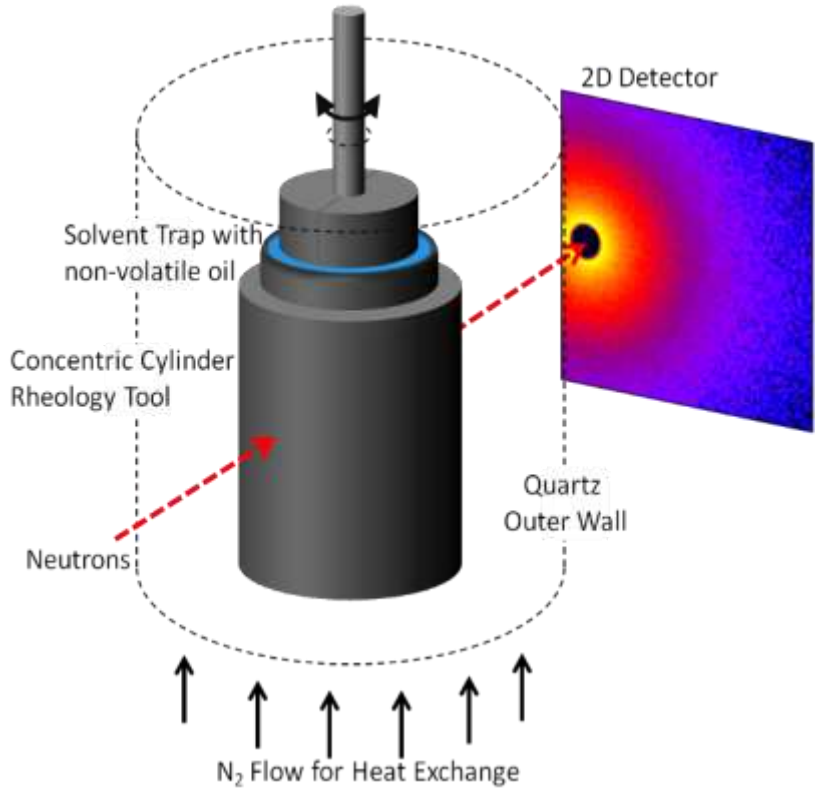
diamond

RheoSAS Subgroup

We met with Anton Paar after the canSAS meeting in Germany in 2019 to discuss SAS-Rheometer communication and other issues:

- Rheometer command output was printed in rheoCompass and could be used to create a command dictionary for direct control from scattering instruments.

- Issues obtaining quartz cup and bobs due to retirement at Hellma



RheoSAS Subgroup

Is there interest in a virtual RheoSAS half day meeting to discuss progress and issues since the Germany meeting?

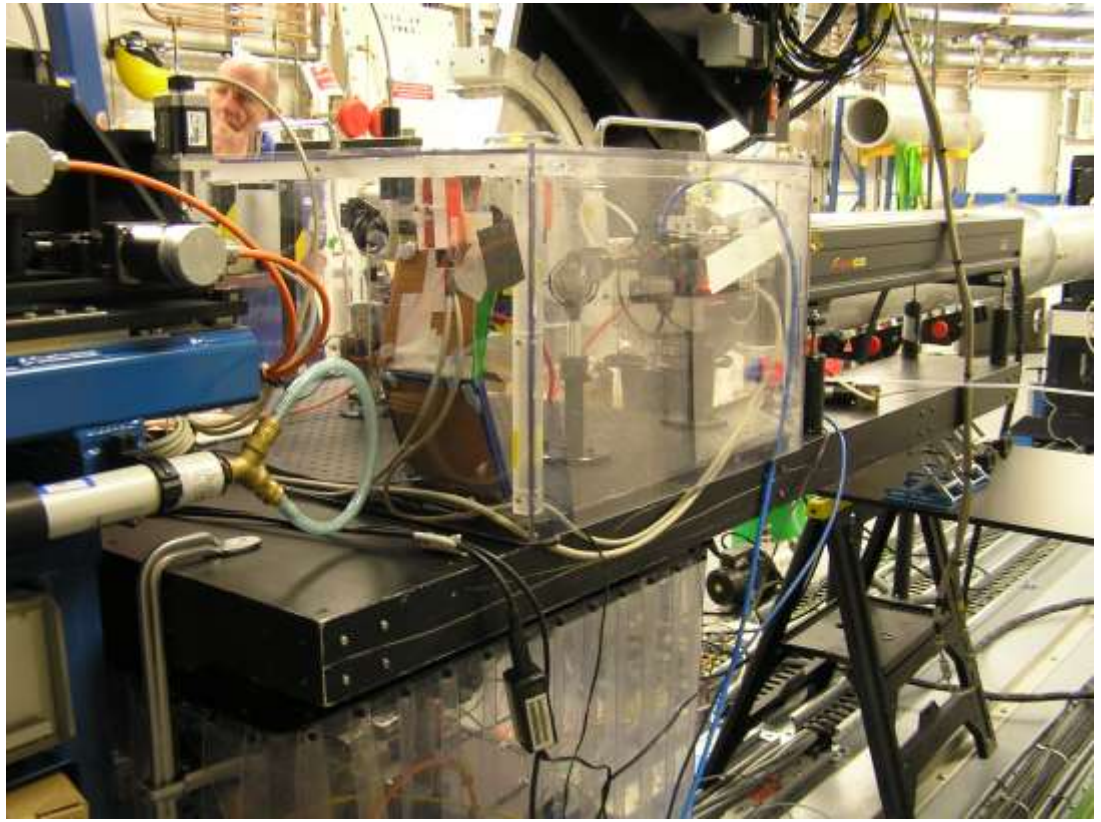
- EPICS drivers at facilities (ORNL, ESS?)
- SECOPS for rheoSANS
- Direct control of rheometer from SANS Instrument?
- Quartz cup and bob?
- Are there other topics of interest or anyone with specific updates they would like to share?

Please Email Katie Weigandt at Katie.Weigandt@nist.gov if you would like to join a virtual rheoSAS update discussion in late 2024 or early 2025. Thank you!

diamond

User Brought Sample Environments

- The problem
  - They are not designed to be integrated
  - They use a different operating system
  - They don’t use the right cables!
  - They are triggered weirdly
  - The user wants all the variables saved with the facility data



EPICS Integrations

- Python based caproto program
- Allows for easy PV access
- Trialing with users now

```python
# Some bits to set up the PV server side of things
class ArduinoPVs(PVGroup):
    HUMID = pvproperty(value=0.00, doc='Sensor Humidity Readback')
    TEMP = pvproperty(value=0.00, doc='Sensor Temperature Readback')


def arduino_pv_server(beamline_PV_base):
    parser, split_args = template_arg_parser(default_prefix = beamline_PV_base,
                                             desc='Arduino Temperature and Humidity Sensor')

    ioc_options, run_options = split_args(parser.parse_args(args = ['-q']))
    ioc = ArduinoPVs(**ioc_options)
    print('PVs:', list(ioc.pvdb))
    run(ioc.pvdb, **run_options)


# The main method for the script
if (__name__ == "__main__"):
    try:
        options, arguments = getopt.getopt(argv[1:], 'b:d:', ['beamline', 'device='])
    except getopt.GetoptError as error:
        print('\n Error: ' + str(error))
        how_to_use_script()
        exit()

    serial_device = None
    beamline = None

    for option, argument in options:
        if ((option == '-b') or (option == '--beamline')):
            beamline = argument
        elif ((option == '-d') or (option == '--device')):
            serial_device = argument

    if (serial_device != None):
        temp_and_humidity_cell = serial.Serial(serial_device, 9600, timeout = 1)
        atexit.register(temp_and_humidity_cell.close)
```

Please Email Tim Snow at tim.snow@diamond.ac.uk if you are interested or already do something similar.

diamond