

Running ODNP

Ryan Barnes - Songi Han's Lab

September 9, 2015

1 Outline of Experimental Setup

This document attempts to explain how to run the ODNP experiment on the setup at Ruhr Universitat Bochum. This section is more of a theoretical explanation of how everything works and how each item is orchestrated into the bigger picture. See section 2 for the practical outline on setting up and running the experiment.

The ODNP experiment is primarily an NMR experiment in that we measure the NMR signal, but enhancement and T_1 of the proton of water as a function of microwave power so all that is done on the EPR side is to change the microwave power. Therefore this experiment is primarily run by TOPSPIN however the experiment is also dependent on Xepr to set the microwave power which is managed by TOPSPIN through an instrument server.

1.1 Enhancement Measurement

In the ODNP experiment it is necessary to measure the NMR signal enhancement as a function of microwave power. We measure the NMR signal enhancement via the integral of the Fourier Transform of the FID. We typically use a 4 step CYCLOPS phase cycle for each enhancement measurement at a given microwave power. Note that the CYCLOPS routine is just to eliminate an offsets due to the IQ mixer and such.

We measure the NMR signal enhancement for a given power setting, see section 1.3. We also record the microwave power throughout the enhancement measurement for a given power setting, see section 1.4. To calculate the overall signal enhancement we plot the integral of the NMR signal against the corresponding microwave power, we extrapolate this to infinite power and whalhh.

In the experiment file the enhancement measurements start at experiment 5 and continue to 5 + the number of enhancement steps (defined in the exp setup)

1.2 T_1 Measurement

In the ODNP measurement we measure the nuclear T_1 as a function of the microwave power as well. This is an attempt to correct for the temperature change during the experiment [1].

1.3 Setting Microwave Power

This is taken care of automatically by the ODNP code `''dnpxp''` as well as `''mwpower''` which just sets the microwave power and amplifier state. Both commands `''dnpxp''` and `''mwpower''` are issued in the TOPSPIN command line as this is the main user interface.

TOPSPIN issues the microwave power setting commands to the Xepr software via the instrument servers, see section 1.5. Xepr ultimately is the program that controls the microwave power adjustment. This is done via adjusting the high power attenuation setting in pulse mode.

1.4 Measuring Microwave Power

We measure the microwave power with the gigatronics microwave power meter. It's necessary to sample the microwave power after the high power amplifier so right now the sampling occurs between the microwave bridge output and the cavity. This isn't ideal as it reduces the cw-EPR signal to noise by about 85% which isn't good.

1.5 Instrument Servers

For the setup in Bochum two instrument servers are required each running on it's own computer. That is one runs on the TOPSPIN computer and one runs on the Xepr computer and they're named accordingly 'instrumentServerTopspin.py' and 'instrumentServerElexsys.py' and they're both stored in the rubODNP directory on each computer. If you note both computers hold an identical copy of the directory, as the directory is managed by git. As you may be able to tell both servers are python programs, as python is awesome!

The server running on the TOPSPIN computer is the parent or the 'main hub' this server listens for commands from TOPSPIN and parses commands either to the Xepr server (for setting the microwave power) or to the power meter (for measuring microwave power).

TOPSPIN communicates to the server on the TOPSPIN computer via a TCP/IP communication, in reality the TOPSPIN program can talk just as easily to the server on the Xepr computer I just thought it better to have all communication go through one server. In this format TOPSPIN gives commands to the server but does not receive commands from the server

Xepr only receives commands from the server on the Xepr computer and does not issue commands to either server. Really because Xepr sucks and there is no reason for it. Communication between Xepr and the server occurs via the XeprAPI which is Bruker's method of allowing python to manage the Xepr platform.

1.6 File Locations

The files pertinent to the ODNP experiment are contained in different locations on both computers because well my organization blows... Any way the instrument servers are stored in the rubODNP folder. On the topspin box this is in '[C:/Projects/rubODNP](#)' on the xepr box this is in the '[/home/xuser/Projects/rubODNP](#)'.

I store the ODNP pulse program and variable defs for xepr in '[/home/xuser/xeprFiles/PulseSPEL/sharedPulseSPEL/standard/dnpElexsys/](#)'.

I store the code to run the ODNP experiment, make a new experiment file, run T_1 experiments, and set the microwave power in

['C:/Bruker/TopSpin3.2/exp/stan/nmr/au/src'](#). Note all of the files are written in c. If you change or edit any of the files all you need to do to compile the files is to write the filename in the topspin command window and it will compile the file and run it. I go through what each of these files are used for in section 1.7

1.7 ODNP Specific Topspin Commands

There are a couple of commands that can be called from the Topspin command line that are pertinent to setting up and running the ODNP experiment. I go through these commands here. All of the scripts are housed in '[C:/Bruker/TopSpin3.2/exp/stan/nmr/au/src](#)', for example the code for the command mwpower is stored in a script called mwpower.

1.7.1 mwpower

The command '[mwpower](#)' sets the attenuation of the high power attenuator in the bridge as well as the state of the microwave bridge. You can use this to manually adjust the power of the microwave output as well as turn the microwave output on or off.

The command operates by communicating with both the server on the Topspin computer and the xopr computer. The commands are initially parsed in the script executed by topspin and sent to the server running on the topspin computer. The server on the topspin computer interprets the command as something for the xopr software and passes the command along to xopr. In all cases the server on the topspin computer is the parent and the server on the xopr computer is the child.

1.7.2 newexp

The command '*newexp*' copies experiment numbers 1-5 in the currently selected experiment directory and deletes any data stored in the experiment. Experiments 1-5 are templates for the ODNP experiment. This is a handy tool to just make sure everything that you need for a successful ODNP experiment is copied successfully from a previous ODNP experiment.

1.7.3 dnpexp

The command '*dnpexp*' is what actually runs the ODNP experiment. This will run the NMR enhancement as a function of microwave power automatically. This will also run the NMR T_1 as a function of microwave power. This command will output data in the form that can be worked up by Ryan's workup software.

The command will grab all of the optimization parameters given in experiment 1. This includes the RF (SFO1), pulse power (PLW1), and pulse length (P1) and copy these parameters to the template files for the enhancement series (experiment 5) and the T_1 series (experiment 4).

The experiment proceeds as follows:

1. Check the receiver noise
2. Measure the thermal NMR signal, exp 5
3. Let the amplifier warm up
4. Measure the enhancement series, exp 6 - number of attenuation steps + 6
5. Turn the microwave pulsing off and let the sample cool down
6. Measure the zero power T_1 , exp 304
7. Let the amplifier warm up
8. Measure the T_1 of power series, exp (number of attenuation steps + 6) - exp (number of attenuation steps + 6 + t1type)

There are several parameters that one could edit to change the odnp experiment.

2 Setting up and Running The ODNP Experiment

2.1 Sample Prep

1. Cut a capillary using the cermaic cutting square to a length where the capillary sticks out of the bottom of the NMR probe by approximately 2 mm.
2. Place $4\mu L$ of sample into the capillary.
3. Seal the sample end of the capillary with critoseal. The sample end is the side of the capillary that you placed the sample in.
4. Seal the other side of the capillary, the side with air, with hot wax.
5. Put a touch of wax on the critoseal side of the capillary. This will help hold the sample in the NMR probe.

2.2 Put Sample in Cavity

1. Bring up the tuning window in Xepr, in tune mode so you see the dip.
2. Place the sample into the NMR probe. The critoseal side goes to the top of the probe.
3. Slide the NMR probe into the cavity. Make sure the B_1 of the NMR probe is oriented appropriately relatively to the static field of the magnet.
4. Check and make sure the tuning dip is still there. The position of the probe in the cavity should be such that the probe sits in the center of the cavity. *'With some resonators, not this 'high Q' unfortunately, you can tell by the dip where the sample is. For this resoantor just mark the position of the probe with a sharpie or something.'*

2.3 Run a CW EPR experiment

1. Critically couple the epr cavity, manual tune seems to work best but what ever you prefer.
2. Open a cw experiment. Go to Aquisition → New Experiment → select the cw tag and name the experiment *"cw"* or something that you remember because you will select this later when you set up another experiment.
3. Run the cw experiment.
4. Note the field of the central transition. *'You will set the static field to this value later.'*

2.4 Configure Xepr to run ODNP - XeprAPI

1. Create a new advanced pulse experiment. Name this *'odnp'* so you know which one it is.
2. Set the bridge to pulse mode.
3. Open the ODNP pulseSPEL experiment and compile it. To open it go to *'Aquisition'* in the experiment options, click the *'PulseSPEL'* button, the click *'Load Program'*. Navigate to *'/home/xuser/xepFiles/PulseSPEL/sharedPulseSPEL/Standard/dnpElexsys/'* and open *'DNP_contShot.exp'*. Then click *'Compile'*.
4. Open the variable definitions by clicking *'Load Var Defs'* in the PulseSPEL window. Open the file *'DNP_contShot.def'* in the same folder as the pulse program and compile it by clicking *'compile'*.
5. Enable the XeprApi. Go to *'Processing'* → *'XeprAPI'* → *'Enable Xepr API'*.
6. Start the server that runs on the Xepr computer. Open

2.5 Setting up the NMR - TOPSPIN

2.6 Enable the topspin server and check for Enhancemnet

3 Observed Bugs with Xepr

Yes this software blows, however it's what we have to deal with so we must get through it.

On a side note being Xepr is buggy it is very possible / simple to utilize the auxillary output of the microwave bridge and circumvent the use of Xepr for the ODNP experiment. In this context you would just configure the aux microwave output to go through a digital attenuator (seperately purchased) then go to the Bruker solid state amplifier. Here you could use a raspberry pi to set the attenuation of the attenuator and to trigger the solid state amplifier. *'This setup would free the ODNP experiment from Xepr, which if Xepr continues to be as buggy as it is now I think this might be a necessary thing to do! Also if you're reading this and thinking WTF? just send Ryan Barnes a mail :).'*

Here I just list a collection of observed bugs that I've noticed and how I've dealt with them in the past. When you observe something that prohibits you from running an experiment there is a chance that I've observed the same bug and have listed a way to deal with it

1. **Tuning window is unresponsive and shows a flat line:** *'First make sure the bridge is in cw mode!! If it is in pulsed mode the tuning window will do this.'* I find that when I close and reopen the Xepr software the tuning window no longer responds to commands and shows a flat line, uncharacteristic of the microwave transmission line. *'To fix this, I close the software; I turn off the spectrometer and wait a couple seconds then turn the spectrometer back on; I then reopen the software. Now I can see the tuning dip.'*
2. **The green button on the console does not stay in:** It seems if you leave the power supply on and the console off for any extended period of time the breaker throws and power is not delivered to the console. *'To fix this go to the back of the power supply and flip the switch that is pointing down. This will turn the power to the console back on, once this occurs you can press the green button'*
3. **XeprAPI does not initialize in the server:** The server fails to open because the XeprAPI does not find an xrepr instance. *'To fix this you must go into xrepr and enable the XeprAPI. Go Processing -> XeprAPI -> click Enable. The server should now work. If this still doesn't work it's probably because something has the xrepr instance. Close any terminal application that could have had xrepr. I notice if I run xrepr from python and jctrl-c exit that python program it doesn't let go of the xrepr and I must close that terminal window completely to re-access xrepr'*

References

- [1] John M Franck, Anna Pavlova, John A Scott, and Songi Han. Quantitative cw Overhauser Dynamic Nuclear Polarization for the Analysis of Local Water Dynamics. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 2013.