

# Running ODNP

Ryan Barnes - Songi Han's Lab

August 13, 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 whalabh.

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 `"dnpepr"` as well as `"mwpower"` which just sets the microwave power and amplifier state. Both commands `"dnpepr"` 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 gigatronix 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 'instrumentServer-Topspin.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.

## 2 Setting up and Running The ODNP Experiment

### 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 xepr instance. *'To fix this you must go into xepr and enable the XeprAPI. Go Processing -> XeprAPI -> click Enable. The server should now work.'*

## 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.