AT-PMSC-1 NMR AUTOTUNING SYSTEM

Installation and Operation Manual

Firmvare version 1.0a, August 20th 2010

Contents

Preface

Notation used in this guide:

- small caps are used for names of hardware components. Example: POWER ON button.
- text in italics is used for modes of operation. Example: *tune* mode.
- string literals are enclosed in double quotes. Example: "TRIM INFO" remote command.

1 Introduction

TEHO AT-PMSC-1 is an autotuning system for NMR spectroscopy, comprising of a broadband research-grade goniometer probe ATP-63G-CF, detachable stepper motor drive unit ATM-C4G, high/low power RF switch ATS-500, and CompactPCI-based autotuning controller ATC-400. It provides for automatic tuning of RF resonant circuits in NMR experiments with X-nuclei resonance frequencies from 10 MHz to 400 MHz. Its application areas span from wideline NMR of disordered solids where spectra are acquired using point-by-point frequency scan methods, to investigations of smeared phase transitions via NMR of low-sensitivity nuclei or low-concentration probes where detuning takes place due to temperature-dependent capacitance and inductance of the trimming elements of the NMR detection circuit. In addition, ATC-400 controls the goniometer of the ATP-63G-CF probehead.

AT-PMSC-1 autotuning system is typically integrated into the RF detection circuit of the spectrometer by an in-line connection of the remotely controlled RF switch, immediately in front of the probe (see section 2.1). During the signal acquisition, high-power RF pulses and low-power NMR signal are redirected in the RF switch directly to or from the probe. If the autotuning functionality and reorientation of the sample are needed in the course of experiment, they can be initiated via the RS-232 connection from the workstation which controls the operation of the spectrometer. Specifically, the synchronization of autotuning operation of AT-PMSC-1 with the spectrometer is established via remote calls from within the spectrometer automation language (Python or C with most modern spectrometers). In this way, operation of AT-PMSC-1 independently of spectrometer's hardware is achieved; no RF or I/O connectivity between the spectrometer and AT-PMSC-1 is required.

The system can also be used as a standalone device to tune the RF resonant circuit of the integrated broadband probe. This is particularly useful in the preparation phase of an NMR experiment whenever user-designed RF coils are used in order to improve the S/N ratio or to expand the frequency range of the probe tunability.

The ATP-63G-CF broadband multipurpose goniometer probe is designed to fit into the Oxford Instruments SpectrostatNMR wide-bore cryostat of standard inner bore dimensions (ϕ 2.5"). When the ATM-C4G stepper motor unit is detached or not powered, the ATP-63G-CF probe can be used in manual mode for conventional NMR spectroscopy (see section 3.3 for instructions on how to realign the trimmer shafts after using the probe in the manual mode).

2 Installation

2.1 Interconnecting the autotuning system and the spectrometer

The TEHO NMR autotuning system consists of four units: the ATC-400 digital autotuning controller, the ATS-500 high-power RF switch, the ATM-C4G stepper motor unit and the ATP-63G-CF wideline NMR probe. You should interconnect these devices with your NMR spectrometer as shown in Fig. 2.1.

The ATS-500 unit uses electromechanical means to switch between *tune* and *standby* modes, therefore it malfunctions in strong magnetic fields. Please make sure that the stray magnetic field where the unit is located never exceeds 10 G (1 mT). With the modern ultra-shielded magnets, an appropriate location is in the neighborhood of the preamplifier mini-tower, typically positioned at one of the legs of the magnet stand. Similarly, the ATC-400 controller should be located in a stray field lower than 5 G (0.5 mT), e.g. in the spectrometer rack cabinet.

2.2 Turning the ATC-400 on and off

The main power switch is located on the back panel of the ATC-400. By turning on this switch, the CompactPCI-based CPU blade is booted and the power supplies of the built-in RF power meter, stepper motor controllers, and interface boards are energized. Main power should normally be turned on/off concurrently with the NMR spectrometer.

The front panel power button is used to power-up or power-down the CPU blade and the digital acquisition and control cards connected to the CompactPCI backplane. The button is red-colored when the CPU is in the power-down state, whereas green color designates the power-up state. By pressing the front-panel power button in the power-down state, the ATC-400 operating system and control application are loaded and the color of the button changes via yellow to

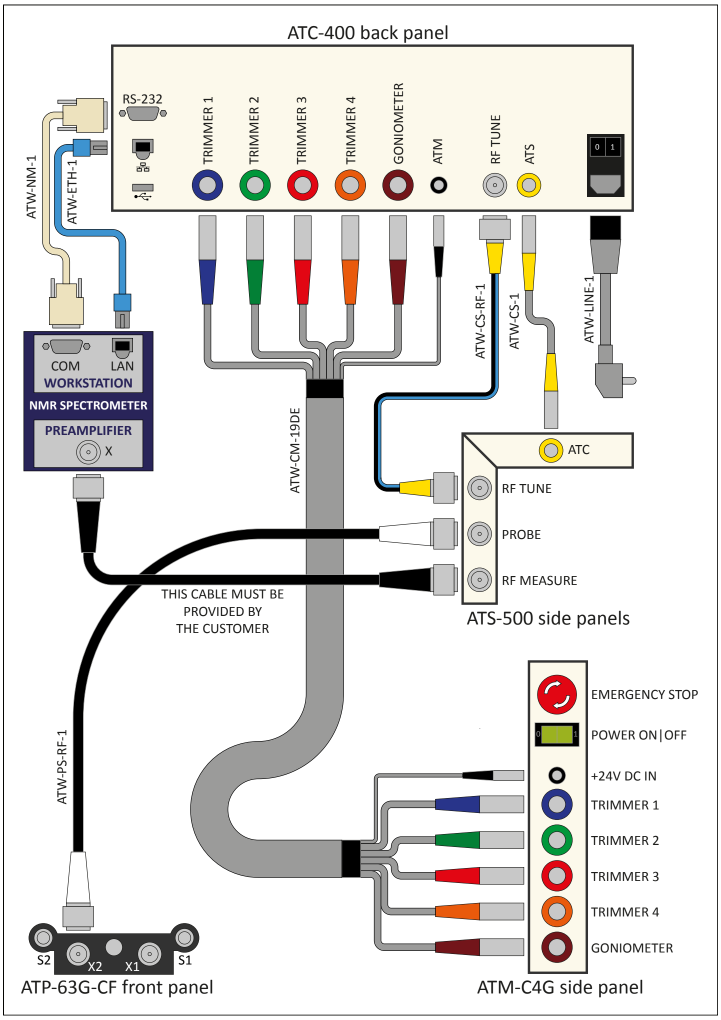


Figure 2.1: Autotuning system – spectrometer interconnection diagram.

green.

In case of software or hardware errors, you should always first try to troubleshoot them by cycling the power with the front panel power button.

2.3 ATM-C4G and ATS-500 powering considerations

Power to ATM-C4G motor unit and ATS-500 high-power RF switch is supplied by the ATC-400 autotuning controller. After a successful power-up of the ATC-400 controller, turn on the POWER ON switch on the ATM-C4G front panel. Also check that the EMERGNECY STOP button is released. There are no power control switches on the ATS-500.

In the normal, i.e. *tune* operational mode of the ATC-400, the green light of the ATM-C4G power switch is on, signaling that the on-board diagnostics (limit switch circuits and front panel LED indicators) is active. On the ATS-500, the tuning mode is indicated by the active yellow LED.

In the *standby* (NMR signal acquisition) mode, the ATM-C4G diagnostic circuit, the ATC-400 RF power meter, and the stepper motors are powered-down in order to minimize RF interferences. The green light of the ATM-C4G power switch is off, in spite of the fact that the power switch is in the *on* position. This mode is indicated by a white LED on the ATS-500.

3 Operation

3.1 Operating the ATC-400

Use the touchscreen controls to operate the autotuning system. You can optionally use a USB mouse or keyboard by plugging them into any of the three front panel USB ports.

All the operating parameters are saved immediately, so that whenever you restart the ATC-400, the user interface and parameters of the last session are reloaded.

3.2 Updating/restoring the ATC-400 firmware

In case of a fatal hardware or software event, you should restore the firmware using the Update/Restore USB Flash Key supplied with the system or newly provided by TEHO. You may also need to update the firmware to a newer version. Please follow these instructions:

1. Turn off the main power switch at the back of the ATC-400.

2. Insert the Update/Restore USB Flash Key into one of the front panel USB ports.

3. Turn on the main power switch at the back of the ATC-400 and wait for the automatic update/restoration procedure to complete. You will be notified on-screen about the progress.

4. When prompted to do so, remove the Update/Restore USB Flash Key and power-cycle the main power switch.

3.3 Realignment of trimmer shafts

Under normal circumstances, you should never manually rotate the trimmer and goniometer shafts while the ATM-C4G motor unit is detached from the ATP-63G-CF probe, e.g. for probe insertion. However, you may want to manually operate the probe in some experiments, without using the ATM-C4G motor unit. If you do so, the shafts need to be realigned before you can reattach the ATM-C4G unit as follows:

1. With ATM-C4G detached from the ATP-63G-CF, set the slider indicators of the four trimmer shafts of the ATM-C4G to position *0 turns* by rotating the respective axes.

2. Set the four trimmer shafts of the ATP-63G-CF probe to the middle position. Specifically, for a trimmer capacitor with a full rotational range of N turns, first rotate the respective axis in the negative direction (indicated by – on the label next to the shaft) till you reach the end of the range. Then rotate the shaft by N/2 turns in the positive direction (indicated by +).

3. Reattach the ATM-C4G unit to the ATP-63G-CF probe. Initially, the three support bars of the ATM-C4G will not seat fully into the holes on the ATP-63G-CF, because the orientation of the male shaft mechanical connectors of the ATM-C4G will not match the orientation of the female connectors of the ATP-63G-CF. In order to provide for a perfect fit, you must fine-adjust the shaft orientations manually (by not more than a quarter of a turn).

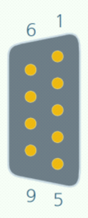
4 Remote control

The bi-directional serial data communication between the computer and the ATC-400 Digital Autotuning Controller takes place via a 9-pin D-sub male socket on the rear panel of the ATC-400. The onboard ESD protected RS-232E transceiver allows for a bit rate of up to 230 kbps. ATC-400 is configured as a DCE with the standard pinout (Fig. 4.1). In order to make a connection with the spectrometer, use the null-modem serial cable supplied with the ATC-400. Default factory settings for the communication protocol are:

- baud rate: 9600
- data bits: 8
- stop bits: 1
- parity: none

4.1 List of remote commands

Remote command strings must be terminated by the carriage return (CR) character. Remote commands will evoke a response from ATC-400 in the form



		1	DCD
DSR	6		
		2	RXD
RTS	7		
		3	TXD
CTS	8		
		4	DTR
RI	9		
		5	GND

Figure 4.1: Rear panel RS-232 D-sub male connector pinout.

par_name	par_val	Description
LAST_LEVEL	<level>	Tuning level in dB of the last successful autotuning operation. Read-only.
STATUS	<status>	ATC-400 status. Read-only.

Table 4.1: Controller parameters.

of a string of printing ASCII characters, terminated by the CR and line feed (LF) characters. The response will normally be sent immediately following the command. An unrecognized command will return the "?" character followed by the command string. Optional values/directives are denoted by [...], whereas → denotes the return value. When entering these values, please make sure that they are of an appropriate type (see tables 4.1, 4.2 and 4.3). In case of an unresolved error, all commands will return "ERROR".

Currently, most commands do not return "BUSY" when an operation is pending on the ATC-400. In order to avoid issuing a command before the preceding command is completed, you should check the status of the ATC-400 by calling "GET STATUS" (see the example code 4.1). Syntax of remote commands is not case sensitive.

Possible commands are:

GET par_name → par_val

This command will return the value par_val of an ATC-400 controller parameter par_name. See table 4.1 for possible parameter names.
Example: "GET LAST_LEVEL" → "-35.3"

RUN op_name [op_val1 [op_val2 ... [op_valN]]] → op_return_val

This command will start the operation op_name using op_val1 to op_valN. See table 4.2 for a list of possible operation names.
Example: "RUN AUTOTUNE 58.375021" → "OK"

op_name	op_val1 ... op_valN	op_return_val	Description
AUTOTUNE	<frequency>	OK	Initiate autotuning to frequency <frequency>.
LEVEL	no value	OK	Measure tuning level at cursor frequency.
	<frequency>	OK or OUT OF RANGE	Measure tuning level at frequency <frequency>.
ORIENT	<angle>	OK	Initiate sample reorientation to <angle>.

Table 4.2: Controller operations.

Designator	Description
<frequency>	Positive decimal number in the range from 10 to 400. Accuracy 6 decimal digits. Dot as a separator. Example: 128.357389
<level>	Negative decimal number in the range from -50 to 0. Accuracy 2 decimal digits. Dot as a separator. Example: -35.23
<angle>	Positive decimal number in the range from 0 to 360. Accuracy 1 decimal digit. Dot as a separator. Example: 35.2
<status>	READY, BUSY, STANDBY

Table 4.3: Parameter and operation values.

Algorithm 4.1 Sample Python code for a frequency-scan measurement.

```
# Using PyVisa to access COM port
# http://pyvisa.sourceforge.net/
from visa import *
import time

atc = instrument("COM1")
rez = []

frqlow = 70
frqhigh = 80
frqstep = 0.5

indices = range((frqhigh-frqlow)/frqstep+1)
for frq in [frqlow+x*frqstep for x in indices]:
    print frq
    atc.ask("run autotune" + str(frq))
    while atc.ask("get status") != "STANDBY":
        time.sleep(1)
    rez.append(atc.ask("get last_level"))
    time.sleep(3)

print rez
```