Applications of Continuous wave free precession sequences in low-field, time-domain NMR

Tiago Bueno Moraes [1,*], Tatiana Monaretto [2] and Luiz Alberto Colnago [3]

ExpSpel codes to implement the CWFP-T_1 and CP-CWFPx-x pulse sequences at the spectrometer Minisepc mq20, Bruker (NF electronics).

CWFP-T_1 pulse sequence

```
-----------------------------------------------------------------------------
CWFP-T1       (Continuous Wave Free Precession - T1)
T.B. Moraes, T. Monaretto, L.A. Colnago
Rapid and simple determination of T1 relaxation times in time-domain NMR by Continuous
Wave Free Precession sequence, J. Mag. Res., 270 (2016) 1-6.
Embrapa Instrumentação, São Carlos-SP, Brazil    (luiz.colnago@embrapa.br)
-----------------------------------------------------------------------------


Program setup ();
#-----------------Get main parameter from parameter table-------------------#
par;
  scans(16);
  dummy_shots(0);
  rd(5.000000000);
  gain(68);
  det_mode("complex");
  magn_mode   ( "PSD" );
endpar;
return(TRUE);

#----------------Set application configuration table----------------------#
program config();
int temp_int;
real temp_real;
char temp_string[128];

strcpy( temp_string, get_text(CALIBRATION_FILE,"fname" ));
if(ERROR) set_conf (CI_INPUT,TRUE,"File Name","F:\usuarios\Eduardo\Sequencias");
else set_conf (CI_INPUT,TRUE,"File Name",temp_string); endif;

temp_real = get_real(CALIBRATION_FILE,"tau");
if(ERROR) set_conf (CI_INPUT,TRUE,"Delay tau [ms]: ","1");
else set_conf (CI_INPUT,TRUE,"Delay tau [ms]",temp_real); endif;
```

```
temp_int = get_int(CALIBRATION_FILE,"npi");
if(ERROR) set_conf (CI_INPUT,TRUE,"Number of 90 deg. pulses in cwfp","2");
else set_conf (CI_INPUT,TRUE,"Number of 90 deg. pulses in cwfp",temp_int); endif;

temp_int = get_int(CALIBRATION_FILE,"savefid");
if(ERROR) set_conf (CI_SELECT,TRUE,"Save Signal",FALSE);
else if(temp_int==1) set_conf(CI_SELECT,TRUE,"Save Signal",TRUE);
else set_conf(CI_SELECT,TRUE,"Save Signal",FALSE); endif;
endif;

temp_real = get_real(CALIBRATION_FILE,"acq");
if(ERROR) set_conf (CI_INPUT,TRUE,"Acquisition time [ms]","3");
else set_conf (CI_INPUT,TRUE,"Acquisition time [ms]",temp_real);
endif;

set_conf (CI_TEXT,TRUE,"File Name and Delays");
get_conf ("Options","FID/ECHO aquisition",0);
if (ESC) goto escape; endif;
print_line (CALIBRATION_FILE,"fname", tst_conf (CI_INPUT,0));
print_line (CALIBRATION_FILE,"tau", ator ( tst_conf (CI_INPUT,1)));
print_line (CALIBRATION_FILE,"npi", atoi (tst_conf (CI_INPUT,2)));
print_line (CALIBRATION_FILE,"acq", ator ( tst_conf (CI_INPUT,3)));
print_line (CALIBRATION_FILE,"savefid", tst_conf (CI_SELECT,0));
label escape;
return(TRUE);

program measure();
#----variable declaration----#
int tdFid,grpDly,td,dispGrpDly,decim, asdshift;   #---int parameters for digital
filter adjustement---#
real dw,aqFid,swh,toffset,tau,d1, gainr;          #---real parameters for digital
filter adjustement---#
int ph1[20],phrc180[20] phrc[20],ns,ph4[20], ph5[20], phrc1[20], npi, cnt, nsc;
real
p90,p180,acq,rdt,attenrose,no,durtot,durhr,durmin,dursec,recdelay,fidinttrunc,res
t,roundrest,total_time,save_flag, flag_cpcwfp;
real x_array[3204800], yre_array[3204800], yim_array[3204800];
char name[256],name1[256];

#----get parameter from configuration table----#
p90=get("90P");
p180 = get("18P");
d1 =  get_rd;
print_line(RESULTBOX, "aqFid =  ", aqFid, " ms. " );
```

```
acq = get_real(CALIBRATION_FILE,"acq");
tau = get_real(CALIBRATION_FILE,"tau");
rdt=get("RDT");
npi = get_int(CALIBRATION_FILE,"npi");
npi = trunc(npi);
strcpy ( name, get_text(CALIBRATION_FILE,"fname" ));
save_flag = get_int(CALIBRATION_FILE,"savefid");
print_line(RESULTBOX);
print_line(RESULTBOX,"----FID/Echo Acquisition----");
print_line(RESULTBOX,"General data filename: ",name);

#--- Experiment Messages------------#
print_line( RESULTBOX, "------------------------" );
print_line( RESULTBOX, "  " );
if(flag_cpcwfp==1)
    print_line( RESULTBOX, "(CP-CWFP is activated (CP-CWFP)");
else
    print_line( RESULTBOX, "Doing Standard CWFP experiment");
endif
print_line( RESULTBOX, "  " );
print_line( RESULTBOX, "------------------------" );

#-----------------initialize counters------------------------------#
cnt = 0;
#------------------Compatibility check-V8.0-Rev0-FW320-------------#
int compat,appv;
char comp_message1[500];
appv=8;
compat=isCompat(appv);

strcpy(comp_message1,"WARNING - Incompatibility detected! Versions expected:
Firmware 3.2, FPGA V1BA, the minispec Software V3.00, Application Pool V8.");
strcat(comp_message1,"Please check the configurations of your system in <Help>/<About
the minispec>. Results might be invalid - Ignore?");
if (compat == 0)
    print_line(CONFIRMBOX,comp_message1);
    if (ESC) goto end;
        else
         print_line(RESULTBOX,"-----------------------------------");
         print_line(RESULTBOX,"   WARNING - Incompatibility detected!" );
        print_line(RESULTBOX,"Versions expected: Firmware 3.2, FPGA V1BA, the
minispec Software V3.00, Application Pool V8" );
         print_line(RESULTBOX,"Please check the configurations of your system in
<Help>/<About the minispec>. ");
```

```
         print_line(RESULTBOX,"                                  Results might be
invalid.");
        print_line(RESULTBOX,"---------------------------");
        endif;
endif;
#---------------END-Compatibility check---------------------------#

#-----------Customize your filter settings-----------# #V7.0#
decim = 16; #digital bandwidth of the filter is 12500kHz/decim, decim=2^N #
#----------------------------------------------#
#------------- Group Delay table --------------#
    if (decim==1)     grpDly=35; endif;
    if (decim==2)     grpDly=18; endif;
    if (decim==4)     grpDly=22; endif;
    if (decim==8)     grpDly=17; endif;
    if (decim==16)    grpDly=17; endif;
    if (decim==32)    grpDly=14; endif;
    if (decim==64)    grpDly=14; endif;
    if (decim==128)   grpDly=13; endif;
    if (decim==256)   grpDly=13; endif;
    if (decim==512)   grpDly=12; endif;
    if (decim==1024)  grpDly=12; endif;
    if (decim==2048)  grpDly=12; endif;
    if (decim==4096)  grpDly=11; endif;
#------END-V7.0-FW-3-1-0------#

#---Defining the dwell time and the digital bandwidth and recalculating the acquisition
time to match the digitalization rate---#
swh=12500000/decim;
dw=1/swh;
aqFid=acq/1000;
tdFid=trunc(aqFid/dw);
tdFid=tdFid+grpDly; # litle trick to keep the acquisition with always the same duration
regarless the decim selected #
aqFid=tdFid*dw;
asdshift=grpDly;
par;
dbw(swh/1000);
endpar;

#-------------------  phase cycling--------------------------------#
ph1[0]  = 0;    ph4 [0] =0;      ph5 [0]=180;   phrc[0] = 0;     phrc1[0] = 180;
ph1[1]  = 0;    ph4 [1] =180;    ph5 [1]=0;     phrc[1] = 180;    phrc1[1] = 0;
ph1[2]  = 180;  ph4 [2] =0;      ph5 [2]=180;   phrc[2] = 0;     phrc1[2] = 180;
```

```
ph1[3]  = 180;   ph4 [3] =180;     ph5 [3]=0;     phrc[3] = 180;   phrc1[3] = 0;

ph1[4]  = 90;    ph4 [4] = 90;      ph5 [4]=270;  phrc[4] = 90;   phrc1[4] = 270;
ph1[5]  = 90;    ph4 [5] = 270;    ph5 [5]=90;   phrc[5] = 270;   phrc1[5] = 90;
ph1[6]  = 270;   ph4 [6] = 90;     ph5 [6]=270;  phrc[6] = 90;    phrc1[6] = 270;
ph1[7]  = 270;   ph4 [7] = 270;    ph5 [7]=90;   phrc[7] = 270;   phrc1[7] = 90;
ph1[8] = REDO;   ph4 [8] = REDO;    ph5 [8]=REDO; phrc[8] = REDO; phrc1[8] = REDO;

#------------------estimated experiment time------------------------------#
ns = get_scans;
recdelay = get_rd;
durtot= ns*(recdelay+aqFid/1000+p90/1000000);
durhr=trunc(durtot/3600.0);
durmin=trunc((durtot/3600.0-durhr)*60);
dursec=round(((durtot/3600.0-durhr)*60-durmin)*60);
print_line(RESULTBOX,"Estimated experiment time is ",durhr," hours ", durmin,"
minutes ", dursec," seconds");
if (save_flag == 1)
print_line(CONFIRMBOX,"Estimated experiment time is ",durhr," hours ", durmin,"
minutes ", dursec," seconds");
if (ESC)
    go to end;
endif;
print_line(FLASHBOX,"FIDs WILL BE SAVED");
endif

pulses;
cta;
 ssp (p180,ph1);
      sd ((tau/2)- p90/1000);
ploop ((npi+1)/2);
        ssp ((p90/9),ph4);
        sd ((tau/2) - p90/1000. - aqFid*1000+1000*asdshift*dw);
        adi(aqFid*1000, 1, phrc1);
        sd ((tau/2) - p90/1000. -1000*asdshift*dw);
         ssp ((p90/9),ph5);
        sd ((tau/2)- p90/1000 - aqFid*1000+1000*asdshift*dw);
        adi(aqFid*1000, 1, phrc);
        sd ((tau/2) - p90/1000. -1000*asdshift*dw);
cnt=cnt+1; # loop counter #
endploop;
cnt=cnt+1; # loop counter #
    print_line( RESULTBOX, "-----------------------" );
    print_line( RESULTBOX, "tau = ",  tau," ms");
    print_line( RESULTBOX, "d1 = ",  d1," s");
    print_line( RESULTBOX, "number of echoes = ",  npi);
endpulses;

#--------------------- Data pre-processing -----------------------------#
toffset=1000*grpDly*dw; #----temporal offset due to group delay----#
sig_offset(-toffset); #----setting the temporal axis at the center of the excitation
pulse----#
range(1000*grpDly*dw,rdt+p90/2000+1000*(aqFid-dw)-toffset, TRUE);
ph0corr(1000*grpDly*dw,1000*(tdFid*1000-grpDly-1)*dw,0,ON);
measure;
#-------------------Saving FIDs ---------------------------------------#
    no = data_points(rdt, -1 )-1;
    sig_abscissa(0, -1, x_array );
    sig_ordinate(0, -1, yre_array );
    sig_swap;
    sig_abscissa(0, -1, x_array );
    sig_ordinate(0, -1, yim_array );
    sig_swap;
    #replace_signal(x_array,yre_array,no);  #
    #replace_signal(x_array,yim_array,no);  #
    strcpy(name1, name);
    strcat(name1, ".dat");
    file_name(ASCII_FILE, name1 );
    print_table( ASCII_FILE, x_array, yre_array,yim_array,no);
    print_line( RESULTBOX, "-----------------------" );
    print_line( RESULTBOX, "Output file:" );
    print_line( RESULTBOX, name1);
if ( ESC ) print_line( CONFIRMBOX, "USER INTERRUPT !" ); return( FALSE ); endif;
label end;
beep; beep;
return( TRUE );
```

CP-CWFPx-x pulse sequence

```
-----------------------------------------------------------------------------
CP-CWFPx-x        (Carr-Purcell - Continuous Wave Free Precession – x-x)
T. Monaretto, F.D.Andrade, T.B. Moraes, A.A.Souza, E.R.Azevedo, L.A. Colnago
On resonance phase alternated CWFP sequences for rapid and simultaneous measurement
of relaxation times, J. Mag. Res., 259 (2015) 174–178
Embrapa Instrumentação, São Carlos-SP, Brazil    (luiz.colnago@embrapa.br)
-----------------------------------------------------------------------------

Program setup ();
```

```
#----------------------Get main parameter from parameter table--------------#
par;
  scans(8);
  dummy_shots(0);
  rd(5.000000000);
  gain(68);
  det_mode("complex");
  magn_mode  ( "PSD" );
endpar;
return(TRUE);

#----------------------Set application configuration table------------------#
program config();
int temp_int;
real temp_real;
char temp_string[128];

strcpy( temp_string, get_text(CALIBRATION_FILE,"fname" ));
if(ERROR) set_conf (CI_INPUT,TRUE,"File Name","F:\usuarios\Eduardo\Sequencias");
else set_conf (CI_INPUT,TRUE,"File Name",temp_string); endif;

temp_real = get_real(CALIBRATION_FILE,"tau");
if(ERROR) set_conf (CI_INPUT,TRUE,"Delay tau [ms]: ","1");
else set_conf (CI_INPUT,TRUE,"Delay tau [ms]",temp_real); endif;

temp_int = get_int(CALIBRATION_FILE,"npi");
if(ERROR) set_conf (CI_INPUT,TRUE,"Number of 90 deg. pulses in cwfp","2");
else set_conf (CI_INPUT,TRUE,"Number of 90 deg. pulses in cwfp",temp_int); endif;

temp_int = get_int(CALIBRATION_FILE,"savefid");
if(ERROR) set_conf (CI_SELECT,TRUE,"Save Signal",FALSE);
else if(temp_int==1) set_conf(CI_SELECT,TRUE,"Save Signal",TRUE);
     else set_conf(CI_SELECT,TRUE,"Save Signal",FALSE); endif;
endif;

temp_real = get_real(CALIBRATION_FILE,"acq");
if(ERROR) set_conf (CI_INPUT,TRUE,"Acquisition time [ms]","3");
else set_conf (CI_INPUT,TRUE,"Acquisition time [ms]",temp_real);
endif;

set_conf (CI_TEXT,TRUE,"File Name and Delays");
get_conf ("Options","FID/ECHO aquisition",0);
if (ESC) goto escape; endif;
```

```
print_line (CALIBRATION_FILE,"fname", tst_conf (CI_INPUT,0));
print_line (CALIBRATION_FILE,"tau", ator ( tst_conf (CI_INPUT,1)));
print_line (CALIBRATION_FILE,"npi", atoi (tst_conf (CI_INPUT,2)));
print_line (CALIBRATION_FILE,"acq", ator ( tst_conf (CI_INPUT,3)));
print_line (CALIBRATION_FILE,"savefid", tst_conf (CI_SELECT,0));

label escape;
return(TRUE);
program measure();
#----variable declaration----#
int tdFid,grpDly,td,dispGrpDly,decim, asdshift;
#---int parameters for digital filter adjustement---#
real dw,aqFid,swh,toffset,tau,d1, gainr;

#---real parameters for digital filter adjustement---#
int ph1[20],phrc180[20] phrc[20],ns,ph4[20], ph5[20], phrc1[20], npi, cnt, nsc;
realp90,p180,acq,rdt,durtot,durhr,durmin,dursec,recdelay,fidinttrunc,rest,roundre
st,total_time,save_flag, flag_cpcwfp;
real x_array[3204800], yre_array[3204800], yim_array[3204800];
char name[256],name1[256];

#----get parameter from configuration table----#
p90=get("90P");
p180 = get("18P");
d1 =  get_rd;
print_line(RESULTBOX, "aqFid =  ", aqFid, " ms. " );
acq = get_real(CALIBRATION_FILE,"acq");
tau = get_real(CALIBRATION_FILE,"tau");
rdt=get("RDT");
npi = get_int(CALIBRATION_FILE,"npi");
npi = trunc(npi);
strcpy ( name, get_text(CALIBRATION_FILE,"fname" ));
save_flag = get_int(CALIBRATION_FILE,"savefid");
print_line(RESULTBOX);
print_line(RESULTBOX,"----FID/Echo Acquisition----");
print_line(RESULTBOX,"General data filename: ",name);

#--- Experiment Messages------------#
print_line( RESULTBOX, "-------------------------" );
print_line( RESULTBOX, "  " );

if(flag_cpcwfp==1)
    print_line( RESULTBOX, "(CP-CWFP is activated (CP-CWFP)");
else
```

```
    print_line( RESULTBOX, "Doing Standard CWFP experiment");
endif
print_line( RESULTBOX, "  " );
print_line( RESULTBOX, "-----------------------" );

#-------------------initialize counters---------------------------------#
cnt = 0;
#------------------Compatibility check-V8.0-Rev0-FW320-------------#
int compat,appv;
char comp_message1[500];
appv=8;
compat=isCompat(appv);
strcpy(comp_message1,"WARNING - Incompatibility detected! Versions expected:
Firmware 3.2, FPGA V1BA, the minispec Software V3.00, Application Pool V8.");
strcat(comp_message1,"Please check the configurations of your system in <Help>/<About
the minispec>. Results might be invalid - Ignore?");

if (compat == 0)
    print_line(CONFIRMBOX,comp_message1);
    if (ESC) goto end;
        else
            print_line(RESULTBOX,"-------------");
print_line(RESULTBOX,"                     WARNING - Incompatibility detected!" );
print_line(RESULTBOX,"Versions expected: Firmware 3.2, FPGA V1BA, the minispec
Software V3.00, Application Pool V8" );
print_line(RESULTBOX,"Please check the configurations of your system in <Help>/<About
the minispec>. ");
print_line(RESULTBOX,"
Results might be invalid.");
print_line(RESULTBOX,"-------------------------------------------------");
        endif;
endif;
#---------------END-Compatibility check----------------------------#

#-----------Customize your filter settings------------# #V7.0#
decim =8; #digital bandwidth of the filter is 12500kHz/decim, decim=2^N #
#------------------------------------------------------#
#----------------------Group Delay table---------------------#
    if (decim==1)    grpDly=35; endif;
    if (decim==2)    grpDly=18; endif;
    if (decim==4)    grpDly=22; endif;
    if (decim==8)    grpDly=17; endif;
    if (decim==16)   grpDly=17; endif;
    if (decim==32)   grpDly=14; endif;
```

```
     if (decim==64)    grpDly=14; endif;
     if (decim==128)   grpDly=13; endif;
     if (decim==256)   grpDly=13; endif;
     if (decim==512)   grpDly=12; endif;
     if (decim==1024)  grpDly=12; endif;
     if (decim==2048)  grpDly=12; endif;
     if (decim==4096)  grpDly=11; endif;
#------END-V7.0-FW-3-1-0------#

#---Defining the dwell time and the digital bandwidht and recalculating the acquisition
time to match the digitalization rate---#
swh=12500000/decim;
dw=1/swh;
aqFid=acq/1000;
tdFid=trunc(aqFid/dw);
tdFid=tdFid+grpDly; # litle trick to keep the acquisition with always the same duration
regarless the decim selected #
aqFid=tdFid*dw;
asdshift=grpDly;
par;
dbw(swh/1000);
endpar;

#--------------------------------  phase cycling ---------------------------#
ph1[0]  = 0;     ph4  [0] =0;        ph5 [0]=180;   phrc[0] = 0;      phrc1[0] = 180;
ph1[1]  = 180;   ph4  [1] =180;     ph5  [1]=0;     phrc[1]  = 180;    phrc1[1]  = 0;
ph1[2]  = 0;     ph4  [2] =0;       ph5  [2]=180;   phrc[2]  = 0;     phrc1[2] = 180;
ph1[3]  = 180;   ph4  [3] =180;     ph5  [3]=0;     phrc[3]  = 180;   phrc1[3]  = 0;

ph1[4]  = 90;    ph4  [4]  = 90;     ph5 [4]=270;   phrc[4] = 90;     phrc1[4] = 270;
ph1[5]  = 270;   ph4  [5]  = 270;   ph5  [5]=90;    phrc[5]  = 270;   phrc1[5]  = 90;
ph1[6]  = 90;    ph4  [6]  = 90;    ph5  [6]=270;   phrc[6]  = 90;    phrc1[6] = 270;
ph1[7]  = 270;   ph4  [7]  = 270;   ph5  [7]=90;    phrc[7]  = 270;   phrc1[7]  = 90;

ph1[8] = REDO;   ph4  [8]  = REDO;   ph5  [8]=REDO; phrc[8]  = REDO; phrc1[8]  = REDO;

#-------------------estimated experiment time-------------------------------#
ns = get_scans;
recdelay = get_rd;
durtot= ns*(recdelay+aqFid/1000+p90/1000000);
durhr=trunc(durtot/3600.0);
durmin=trunc((durtot/3600.0-durhr)*60);
dursec=round(((durtot/3600.0-durhr)*60-durmin)*60);
```

```
print_line(RESULTBOX,"Estimated experiment time is ",durhr," hours ", durmin,"
minutes ", dursec," seconds");
if (save_flag == 1)
print_line(CONFIRMBOX,"Estimated experiment time is ",durhr," hours ", durmin,"
minutes ", dursec," seconds");
if (ESC)
    goto end;
endif;
print_line(FLASHBOX,"FIDs WILL BE SAVED");
endif

pulses;
cta;
 ssp (p90,ph1);
         sd (rdt);
        adi(aqFid*1000, 2, phrc);
        sd ((tau/2)- p90/2000);
ploop ((npi+1)/2);
        ssp (p90,ph4);
        sd ((tau/2) - p90/1000 - aqFid*1000+1000*asdshift*dw);
        adi(aqFid*1000, 1, phrc);
        sd ((tau/2) - p90/1000 -1000*asdshift*dw);
         ssp (p90,ph5);
        sd ((tau/2)- p90/1000 - aqFid*1000+1000*asdshift*dw);
        adi(aqFid*1000, 1, phrc1);
        sd ((tau/2) - p90/1000 -1000*asdshift*dw);
cnt=cnt+1; # loop counter #
endploop;
cnt=cnt+1; # loop counter #
    print_line( RESULTBOX, "-----------------------" );
    print_line( RESULTBOX, "tau = ",  tau," ms");
    print_line( RESULTBOX, "d1 = ",  d1," s");
    print_line( RESULTBOX, "number of echoes = ",  npi);
endpulses;

#--------------------- Data pre-processing ------------------------------#
toffset=1000*grpDly*dw; #----temporal offset due to group delay----#
sig_offset(-toffset); #----setting the temporal axis at the center of the excitation
pulse----#
range(1000*grpDly*dw,rdt+p90/2000+1000*(aqFid-dw)-toffset, TRUE);
ph0corr(10000*grpDly*dw,100000*(tdFid*1000-grpDly-1)*dw,0,ON);
measure;

#-------------------Saving FIDs ---------------------------------------#
```

```
          no = data_points(rdt, -1 )-1;
          sig_abscissa(0, -1, x_array );
          sig_ordinate(0, -1, yre_array );
         sig_swap;
          sig_abscissa(0, -1, x_array );
          sig_ordinate(0, -1, yim_array );
         sig_swap;
          #replace_signal(x_array,yre_array,no);  #
          #replace_signal(x_array,yim_array,no);  #
         strcpy(name1, name);
         strcat(name1, ".dat");
         file_name(ASCII_FILE, name1 );
         print_table( ASCII_FILE, x_array, yre_array,yim_array,no);
          print_line( RESULTBOX, "------------------------" );
         print_line( RESULTBOX, "Output file:" );
         print_line( RESULTBOX, name1);

    if ( ESC ) print_line( CONFIRMBOX, "USER INTERRUPT !" ); return( FALSE ); endif;
    label end;
    beep; beep;
    return( TRUE );
}
```