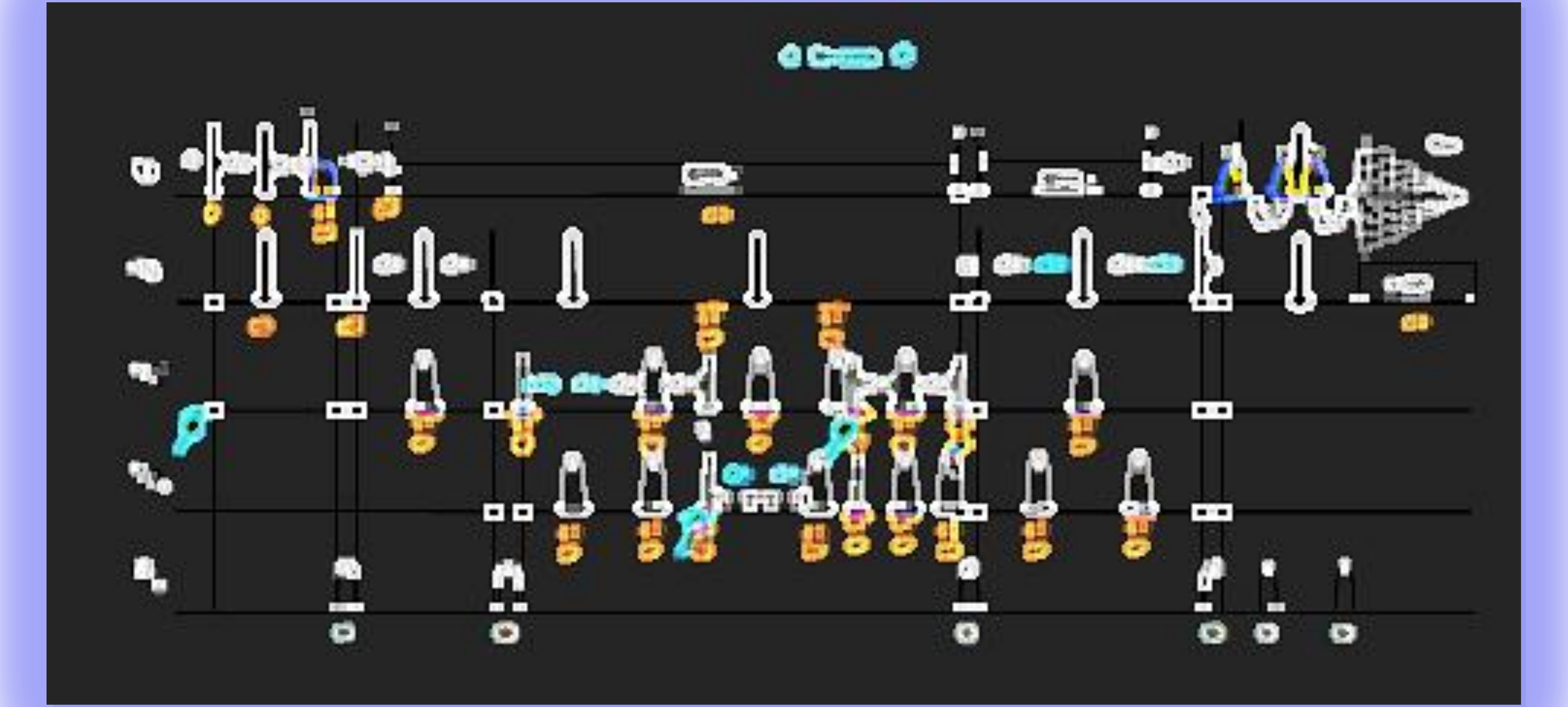
Pulse programming

Some notes and news



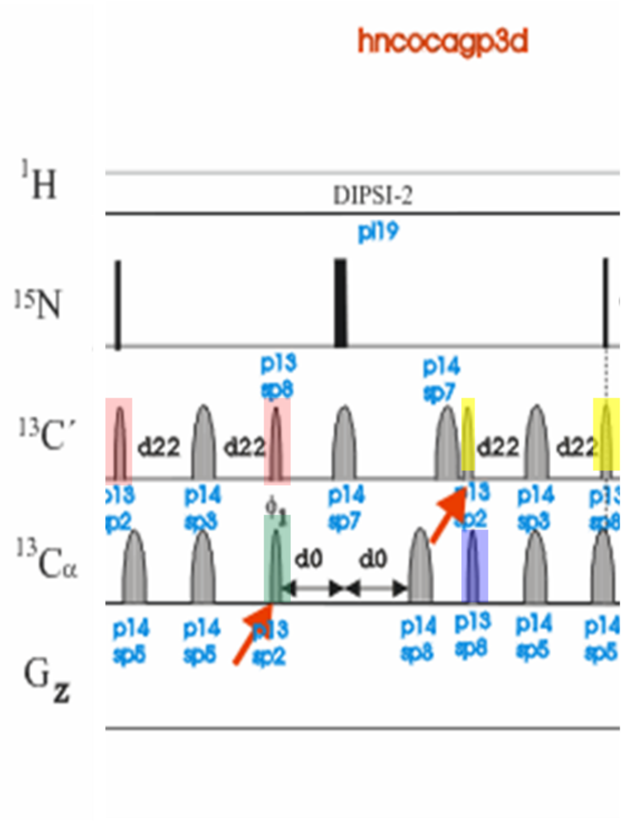
Content

1. Timing for change of rf parameters
2. New mc-statement
3. Shapes and power level settings

Timing for changing rf parameters

phase, frequency, amplitude (shapes only)

HNCOCA



```
(p21 ph1):f3
(p13:sp2 ph3):f2
4u
(p14:sp5 ph1):f2
DELTA3
(p14:sp3 ph1):f2
4u
(p14:sp5 ph1):f2
DELTA3
(p13:sp8 ph2):f2          time reversed
4u
30u fq=cnst23(bf ppm):f2
(p13:sp2 ph4):f2
d28
(p14:sp3 ph1):f2
d28
(p13:sp8 ph2):f2          time reversed
d0
(center (p14:sp7 ph1):f2 (p22 ph8):f3 )
d0
```

```
(p14:sp3 ph1):f2
DELTA4
(p14:sp7 ph1):f2
4u
(p13:sp2 ph9):f2
d28
(p14:sp3 ph1):f2
d28
(p13:sp8 ph10):f2         time reversed
4u
30u fq=cnst21(bf ppm):f2
(p13:sp2 ph2):f2
DELTA3
(p14:sp5 ph1):f2
4u
(p14:sp3 ph1):f2
DELTA3
(p14:sp5 ph1):f2
4u
(p13:sp8 ph1):f2          time reversed
(p21 ph1):f3
```

mc-statement
Introduction

FnMODE parameter =

FQ	phase sensitive 1D
F1QF	magnitude mode QF
F1PH, F2PH	QSEQ, TPPI, States or States-TPPI
F1EA, F2EA	phase sensitive echo-antiecho

- Simplifies pulse programming
  - includes
    - disk write (wr)
    - file pointer incrementation (if)
    - memory initialization (zd)
    - expanded loop structure possible
  - no need to control actions required for phase sensitive 2D experiments (phase or delay incrementation) with delays
- One pulse program can be used for different 2D phase modes

mc-statement

Original implementation, 1D experiments

without mc:

```
1 ze
2 d1
  p1 ph1
  go=2 ph31
  wr #0
exit
```

with mc:

```
1 ze
2 30m
  d1
  p1 ph1
  go=2 ph31
  30m mc #0 to 2 F0(zd)
exit
```

Expanded with mc:

```
define delay MCWRK
define delay MCREST
“MCWRK=0.333333*30”
“MCREST=30m-30m”

1 ze
2 MCWRK*3
  LBLF0, MCREST
  d1
  p1 ph1
  go=2 ph31
  MCWRK wr #0
  MCWRK ze
  MCWRK zd
  lo to LBLF= times td0
exit
```

mc-statement

Original implementation, 2D experiments

up to XWINNMR 2.6

```
;noesytp

#include <Avance.incl>

"d0=3u"

1 ze
2 d1
3 p1 ph1
  d0
  p1 ph2
  d8
  p1 ph3
  go=2 ph31
  d1 wr #0 if #0 ip1 id0 zd
  lo to 3 times td1
exit
```

original mc:

```
;noesyph

#include <Avance.incl>

"d0=3u"

1 ze
2 d1
3 p1 ph1
  d0
  p1 ph2
  d8
  p1 ph3
  go=2 ph31
  d1 mc #0 to 2 F1PH(ip1, id0)
exit
```

mc-statement

Original implementation, 2D experiments

F1QF	phase insensitive	QF
F1PH	phase sensitive	QSEQ, States, TPPI, States-TPPI
F1EA	Echo-Antiecho	Echo-Antiecho

The acquisition and processing parameter FnMODE define the mode for F1 (and F2, for 3D-experiments) incrementation

For details, see XWINNMR help -> Other topics -> Writing pulse programs

MC clause	t1 quadrature mode	action after	
		odd increment	even increment
F1PH(ip1, id0)	TPPI	ip1+ id0	ip1 + id0 again
F1PH(ip1, id0)	States-TPPI	ip1	id0
F1PH(rd10 & rd30 & ip4, id0)	TPPI	rd10+rd30+ip4+	id0
F1PH(rd10 & rd30 & ip4, id0)	States-TPPI	rd10+rd30+ip4	id0

mc-statement

Challenge for non-uniform sampling

Explicit programming of a 2D experiment with non-uniform sampling

```
define list<loopcounter> t1list=<$VCLIST>        variable counter list for sampling scheme

1 ze
……
4 d1
"d0=3u+in0*t1list"             calculate value of t1 delay d0
 3m t1list.inc
…..
go=2 ph31 cpd2:f2
d11 do:f2 wr #0 if #0 zd

"cnst30=(t1list%2)*180"        calculate phase increments
"cnst31=cnst30"
  3m ip6+cnst30
  3m ip9+cnst30
…..
```

mc-statement

New implementation for non-uniform sampling

	Old	*New*
Phase	**ip1**	**calph(ph1, +90)**
Delay	**id10**	**caldel(d10, +id10)**
Loop counter	**iu0**	**calclc(l0, 1)**
Phase reset	**n.a.**	**exec(rppall)**

Execution of conventional commands:

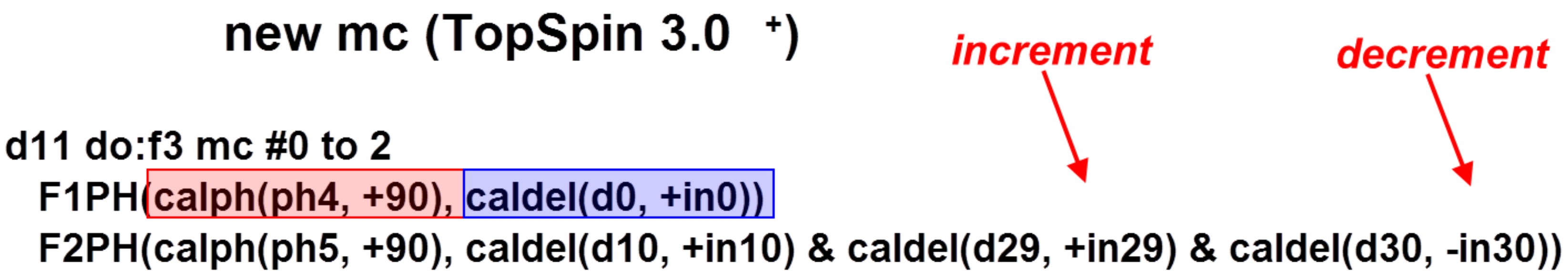
Incrementation of a frequency list defined by: define list<frequency stdlist=<$FQ2LIST>

exec(stdlist.inc)

Lists		**calclist(name, incr)**

mc-statement

changes for non-uniform sampling



new mc (TopSpin 3.0 $^+$)

increment

decrement

```
d11 do:f3 mc #0 to 2
  F1PH(calph(ph4, +90), caldel(d0, +in0))
  F2PH(calph(ph5, +90), caldel(d10, +in10) & caldel(d29, +in29) & caldel(d30, -in30))
```

mc-statement

Important note

Do not mix old and new mc-statements

Shaped pulses and power level settings

Power level switching before and after shaped pulses

```
d16 pl0:f1
 (p12:sp1 ph2:r):f1          shaped pulse
 4u
 d12 pl1:f1
 p2 ph3                      hard pulse
```