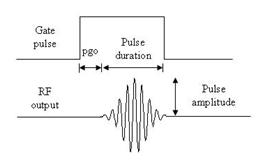
# dualshapedrf2

This command generates RF pulses at the current transmit frequency, which can be modulated in amplitude and phase. It also includes a gate pulse which is used to control an RF amplifier. This version replaces the dual pulse version of shapedrf, the latter not being available on FX3 based Spinsolves.



*Applicability*

*DSP/FX3*

*Syntax*

dualshapedrf2(ampPhaseTables, phase1Offset, phase2Offset, tableSize, stepTime)

ampPhaseTables . the name of table which controls the amplitude and phase during the generation of the shaped pulses. It has the form “tx” where x is an integer greater than 0. The table should contain interleaved amplitude and phase values (ch1 amp, ch1 phase, ch2 amp, ch2 phase). The amplitude values should be positive with a minimum value of 0 and a maximum of 16383. Use the command ucsRun:convertTxGain() to convert the amplitude in dB to one in this linear range. The phase values should contain positive integers with a minimum value of 0 and a maximum of 65535. This corresponds to phase shifts between 0 and 359.995 degrees.

phaseXOffset ... the name of the parameter which controls the RF pulse phase relative to the system clock. X is 1 for channel 1 and 2 for channel 2. It has the form “px” where x is an integer greater than 0. Phase values in the phase cycle table can be 0,1,2,3 corresponding to x, y, -x, –y (0, 90, 180, 270 degrees). Typically this is used to implement phase cycling. Fractional numbers can also be applied to implement other phases between 0 and 360 (0-4). The phase can also be supplied as a positive 16 bit integer in which case the parameter name has the form “nx” where x is an integer greater than 0. Constants can also be used.

tableSize ...... the name of the parameter which controls the number of values in each of the amplitude and phase subtables (all should have the same number of entries). It has the form “nx” where x is an integer greater than 0 or a constant can be supplied. (So if tableSize is 100 this means there are 100 values in a single amplitude or phase table and a total table size of 400).

stepTime ....... the name of the parameter which controls the duration of each amplitude step.  It has the form “dx” where x is an integer greater than 0 or a constant can be used.

*Notes:*

1. The pulse generated by this command has a length which includes a fixed delay called the pulse gate overhead (pgo). This delay is used to set up the RF gate – a signal sent to the high powered amplifier to switch on the DC biasing before the RF pulse is actually sent. You should account for this delay when writing the relationships list described above (the predefined variable “pgo” (pulse gate overhead) can be used. The pgo on the Spinsolve is normally set to 5 ms.

1. There are a maximum number of amplitude steps available determined by the amount of DSP or FX3 memory available. The total for all amplitude and phase tables must be less than 128 k values on a DSP Spinsolve and 512 k on the FX3 (but < 64 k per table).

1. The stepTime value may range from 4 ms to 327,670 ms. **Beware**: values below 4 ms can cause the RF to switch on for very long times.

1. Potentially the shaped RF pulse could have a large amplitude and last for a long time – the combination of these situation could damage a high-powered RF amplifier or probe. Always take care when setting the amplitude and setTime values. Place limits where possible.
2. To update the shaped RF pulse between scans requires resending the tables to the DSP/FX3 however the phaseOffset parameter can be applied using a phase cycle table to adjust the phase by a constant amount based on the scan number.
3. In addition to the pgo delay at the beginning of the command it takes ?? ms (DSP)/ 2 ms (FX3) for the command to finish before another command can be given. On DSP systems you should always include (at least) a 1 us delay after this command before issuing another RF pulse.
4. The total duration (in ms) of this command will be pgo + tableSize\*stepTime + ??/2. (DSP/FX3)

*Example pulse sequence: (PSExamples\DualShapeRF2Test)*

############################################################

#

# Generates a dual channel pulse with a ramped shaped

# amplitude and a phase which jumps in the middle by 180

# degrees using an interleaved table and shapedrf.

#

# pulse - delay - shapedrf - delay - acq

#

############################################################

procedure(pulse\_program,dir,mode,pars)

# Expose parameters for FX3 implementation

if(nrArgs == 3)

assignlist(pars)

endif

# Interface description (name, label, ctrl, vartype)

interface = ["nucleus", "Nucleus", "tb", "readonly\_string";

"b1Freq1H", "1H frequency", "tb", "freq";

"b1Freq13C", "13C frequency", "tb", "freq";

"txMaxdB", "Pulse max amplitude (dB)", "tb", "float";

"nAmpSteps", "Amplitude steps", "tb", "integer";

"stepDur", "Table step length (us)", "tb", "pulselength";

"repTime", "Repetition time (ms)", "tb", "reptime"]

# Define the tab groups and their order

groups = ["Pulse\_sequence","Acquisition",

"Processing\_Std","Display\_Std","File\_Settings"]

# Relationships to determine remaining variable values

relationships = ["nDataPnts = nrPnts",

"a1 = 0",

"a2 = 0",

"aMax = gFX3->convertAmplitude(txMaxdB)",

"ampSweep1 = linspace(0,aMax,nAmpSteps/2)",

"ampSweep2 = linspace(aMax,0,nAmpSteps/2)",

"ampSweepa = join(ampSweep2,ampSweep1)",

"ampSweepb = join(ampSweep1,ampSweep2)",

"phSweep1 = matrix(nAmpSteps/2)",

"phSweep2 = matrix(nAmpSteps/2)+32768",

"phSweepa = join(phSweep1,phSweep2)",

"phSweepb = join(phSweep1,phSweep2)",

"tAmpPhase = gFX3->interleaveTables(ampSweepa,phSweepa,ampSweepb,phSweepb)",

"nTabSz4 = size(tAmpPhase)/4",

"dTabStep = stepDur",

"f1 = b1Freq1H",

"f2 = b1Freq13C",

"totPnts = nrPnts",

"totTime = acqTime"]

# These parameters will be changed between experiments

variables = [""]

# Pulse sequence

initpp(dir)

pulse(1,a1,p1,f1,2,a2,p1,f2,10) # Dual channel pulse for scope triggering

delay(10)

dualshapedrf2(tAmpPhase,p2,p2,nTabSz4,dTabStep) # Dual channel shaped pulse

delay(10)

acquire("overwrite",nDataPnts)

parList = endpp(0) # Combine commands and return parameter list

# Phase cycle list

phaseList = [0,0,0,0; # p1 : Trigger phase

0,1,2,3; # p2 : Shaped phase

0,1,2,3] # pA : Acquire phase

endproc(parList,groups,interface,relationships,variables,null,phaseList)

The shaped pulses are generated in the relationship table and consists of two linear ramps. On channel 1 the first ramping down and the second ramping up with the reverse on channel 2. The phase is zero for the first ramp and 180 degrees for the second.

Note the line defining tAmpPhase – this calls a procedure which interleaves the amplitude and phase tables into one.

*Test parameters*

b1Freq = 2 MHz for both channels, maximum amplitude 0 dB.

10 steps in each shaped pulse.

Table step length = 5 ms => total duration 50 ms.

Transceiver channel 1 and 2 output, showing shaped RF pulses. Note the phase jump in the middle of the channel 2 pulse.

A graph of a sound wave

Description automatically generated