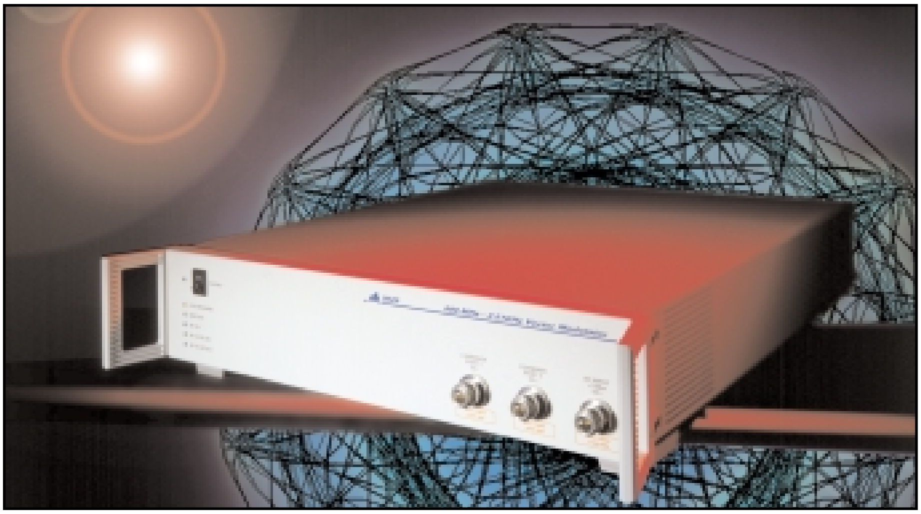
application
note

Creating IQ data files for the 2029 vector modulator

By Thijs Waardenburg



IFR's 2029 vector modulator can be used in conjunction with any signal source to provide a high quality digitally modulated carrier. This application note explains how a user can create and load the files needed to generate test waveforms.

Overview

IFRs 2029 vector modulator can be used in conjunction with any analogue signal source to generate a high quality digitally modulated carrier. The 2029 may be used to generate many different digital modulation schemes, for example WCDMA, EDGE, IS-95, TETRA and GSM. The 2029 uses a large arbitrary waveform generator to provide a base band IQ signal that modulates an external source. The large memory size and the use of interpolation techniques allows the 2029 to generate long test sequences. For example, it is possible to save 15 IS-95 waveforms each of 80 ms duration. Alternatively one IS-95 waveform of 1.2 seconds may be stored. The use of non-volatile memory ensures that it is possible to rapidly change between waveforms.

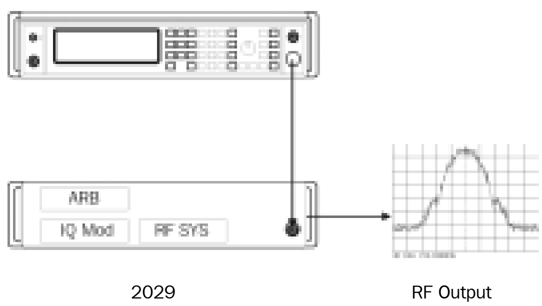


Figure 1

The internal ARB can be loaded with customer waveforms or those provided by IFR. This application note describes how waveforms can be generated and loaded into the ARB. The note will describe:

- The required format of the base band waveforms.
- Constraints that are placed on waveform.
- Creating and downloading a waveform.
- Issues with wrap around when a time limited waveform is continuously played.

Waveform generation

Typically a simulation package, such a Matlab, is used to generate base band signals that can be formatted and downloaded to the 2029. If the created waveform is to be successful it is essential that certain design criteria be adhered to. Some of these are general signal processing concepts, and others are constraints that are placed upon the waveform designer by the 2029 hardware. The following sections address some of the more important issues to consider when generating a waveform.

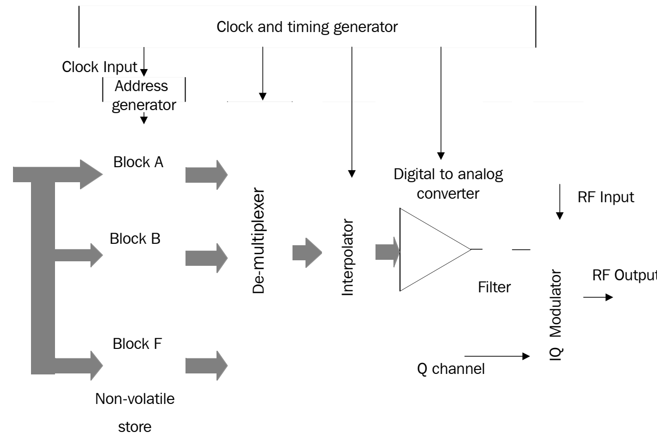


Figure 2

Waveform sampling rate

The following factors govern the choice of sample rate.

1. The bandwidth of the signal to be generated

The bandwidth of the signal will limit the minimum sample frequency that is allowed according to the Nyquist rate. In simple terms the Nyquist theorem states that a band-limited signal must be sampled by at least twice the band-limited frequency of the signal. In other words, if we have a signal that is limited to 1 MHz, then this signal must be sampled at more than 2 MHz if it is to be uniquely determined by its samples. In practice the use of 4 times over-sampling is recommended.

The maximum IQ (complex) bandwidth supported by the 2029 is approximately 20 MHz - generated from a 10 MHz base-band bandwidth.

2. 2029 hardware constraints

The 2029 ARB design limits the input sample rate to between 3.67 MHz and 66 MHz. Such a broad range of sample rate is possible with the 2029 because the design uses interpolating filters that ensure that the DACs receive an optimal sample rate. Due to this design, it is essential that the following conditions are adhered to:

a) The input waveform must not be compensated for sinx/x distortion. This is taken care of in the hardware design.
b) The ARB design (hardware interpolation) assumes that the input waveform has been over-sampled by at least 4 times the symbol rate of the original signal and that the bandwidth of the signal is less than one sixth of the waveform sample frequency. These conditions are not required if the waveform sample rate is above 44 MHz. In this case the interpolation factor implemented in hardware will be one and the constraints given above do not apply.

3. Waveform length

The maximum length of the waveform is limited by the amount of memory in the instrument. The maximum size of an IQ waveform that can be stored in the 2029 is 5 898 240 samples. The minimum length of a waveform is 50 samples. Obviously, the length of the waveform in time, is determined by the sample rate. For example, 80 ms of an IS-95 waveform that has been sampled at 4*1.2288 Msamples per second will require 393 216 samples. In this instance, there is enough memory to store 1.2 seconds of such a waveform.

There is also a limit to the number of different waveforms that may be stored. The maximum number of different waveforms is 15. In this instance the maximum length of any single waveform is 393 216 samples.

Downloading and formatting a waveform

Before the 2029 can play a signal it has to be formatted and downloaded to the instrument. The front panel software provided with the instrument handles the download process. This software supports four file formats:

1. IFR's propriety format. These are files that have been generated by IFR and are generally not available to the user. Typically these files will have the extension *.iq.
2. Agilent's ESG binary format (.bin)
3. 16 bit and 32 bit signed integer binary files that are in Little Endian format with samples stored as I then Q then I then Q etc. (referred to as IQIQIQ from this point on).
4. 32 bit IEEE floating point notation, again with samples stored as IQIQIQ.

For user generated waveforms only file types 3 and 4 are supported. The details of these formats follow.

16 bit integer format

The 16 bit signed integer format refers to a 16 bit twos complement representation, with a range of -32768 to 32767. For the 2029, the waveform values must be scaled so that the peak values do not exceed $\pm 2^{13}$. This is because the 2029 uses 14 bit DACs. The soft front panel software assumes that the waveform has been scaled to lie within the 14 bit range. It is important that the sign information contained in bits 14 and 15 is not removed. In other words the 16 bit twos complement representation must be retained as the front panel software reads the stored data as 16 bit signed integer values. The waveform must be stored as IQIQIQ samples, with I as the first sample. In addition, the binary file must be in Little Endian format. This is the default for Intel based machines.

Assuming that a waveform has been generated using Matlab, and that the IQ data are stored as complex vectors (I + jQ), then the Matlab function given below may be used to save the samples in a format that can be readily downloaded into the 2029 using the front panel software. Details on how to do this are given later.

```
function [error, msg] = saveiq(file_name, iq_data,
req_peak);
% This functions saves the i and q data as 16 bit
% integers in the order I Q I Q. The data is scaled so
% the peak vector is equal to req_peak. If the required
% peak value is greater than 2^13-1 or <= 0 an
% error will occur (function will return -1). Note that
% iq_data is assumed to be complex

error = 0;
msg = [];
if ((req_peak > 2^13-1) | (req_peak <= 0))
  msg = sprintf('Peak value is not valid');
  error = -1;
else
  peak = max(max(abs(real(iq_data))),
  max(abs(imag(iq_data))));
  scale = req_peak/peak;
  int_iq_data = zeros(1,2*length(iq_data));
  int_iq_data(1:2:end) = round(real(iq_data)*scale);
  int_iq_data(2:2:end) = round(imag(iq_data)*scale);
  [fid msg] = fopen(file_name,'w');
  if (fid ~= -1)
    fwrite(fid, int_iq_data, 'int16');
    fclose(fid);
  else
    error = -1;
  end
end
```

The function above shows that the IQ data is stored as signed 16 bit integers that have been scaled to a peak value selected by the user. This particular function also ensures that the maximum peak value is less than or equal to $\pm 2^{13}$. Scaling must be done with caution, as discussed in more detail later.

32 bit signed integer format

The 32 bit integer format takes the same form as the 16 bit signed integer format, except that the data is stored in 32 bits twos complement format. The data must still be scaled to have peak values that are less than or equal to $\pm 2^{13}$.

IEEE 32-bit float

The float format uses the IEEE (Institute of Electrical and Electronics Engineers) single precision format. This format is represented by 4 bytes, consisting of a sign bit, an 8-bit excess-127 binary exponent, and a 23-bit mantissa. The mantissa represents a number between 1.0 and 2.0. Since the high-order bit of the mantissa is always 1, it is not stored in the number. This representation gives a range of approximately 3.4E-38 to 3.4E+38.

The front panel software requires that the data to be scaled such that the peak values are less than or equal to 1.0. Again the samples must be stored as IQIQIQ with I being the first sample.

Preparing a waveform for download

Once you have generated your waveform, you will need to download it to the instrument. This is a two step process. Firstly your waveform needs to be 'packaged' into a form that is understood by the instrument. Once this process is complete, the packaged waveform is downloaded and stored on the 2029. This section describes the first stage, but further information is available in the 2029 operating manual on the CD ROM supplied with the instrument.

The front panel software provided with the instrument allows you to control the 2029 via GPIB and is the only way to access the ARB file packager and file loader software. To package your waveform, start the front panel application and use the GPIB menu on the task bar to capture the 2029 address. The interface presented will be similar to that shown in Figure 3.

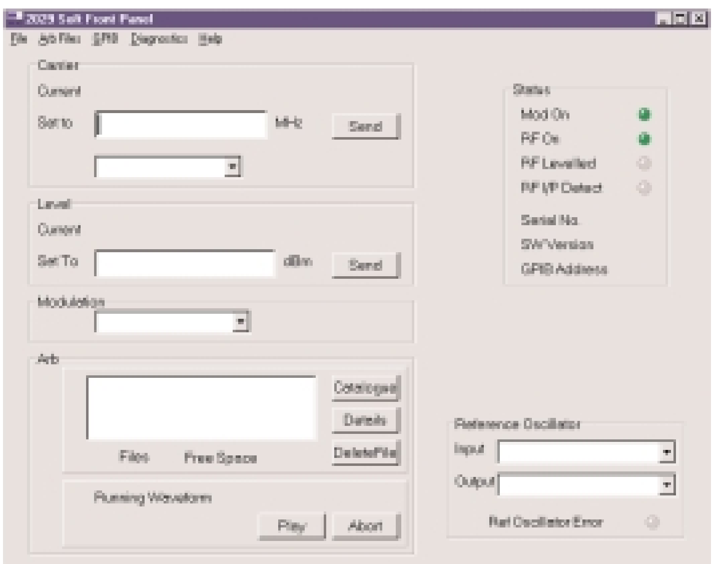


Figure 3 2029 Front panel user interface.

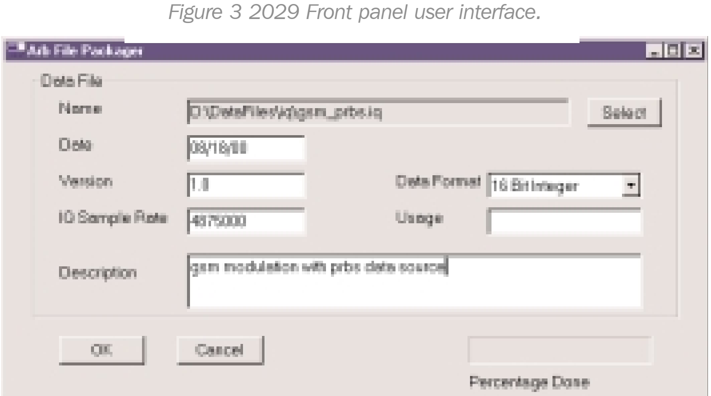


Figure 4 Packager dialog box.

Select the menu ArbFiles followed by Packager. This will then provide you with the dialog box shown in Figure 4. You select your waveform using the Select button. Once you have selected your waveform you will need to set the data format and sample frequency fields. The sample frequency must be entered in Hz. The other fields are optional. Once you are happy with the selections click the OK button. You will then be presented with the dialog given in Figure 5 informing you that for best performance the RMS value of the waveform needs to be calculated.

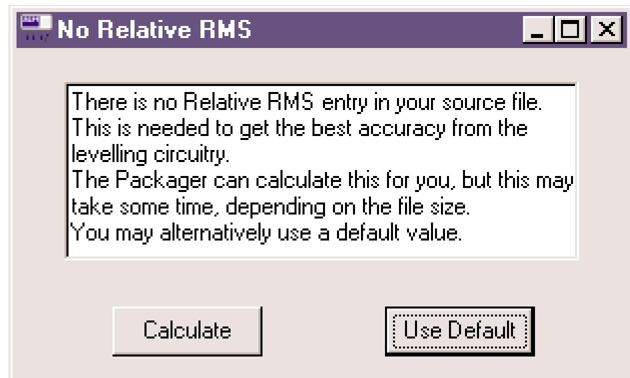


Figure 5 RMS calculation message.

It is recommended that you let the packager calculate the value. Once the software has done the calculation you will be prompted for a file name for the packaged waveform. Choose a file name and continue. Once you have packaged the waveform you are ready to continue to the next step of downloading and playing the file.

Downloading a waveform

The download of a file is a simple operation. To download a file, select the menu option ArbFiles, Loader. You will then be asked to select a file for downloading. Once you have selected a file and done the download, you will find that your waveform will appear in the waveform list. Select this waveform and press the play button. The output should now be modulated with your waveform.

Note that, in the above description it has been assumed that you have selected the carrier frequency, set the amplitude and set modulation to on and turned the carrier on. If not, please refer to the operating manual on how to do this.

Waveform scaling

It was mentioned earlier that the 2029 uses 14 bit DACs. This means that the waveform must be scaled such that the maximum peak value does not exceed $\pm 2^{13}$. However, due to the design of the RF modulators within the instrument, it is recommended that the waveform is scaled so that its RMS value is 2300 in integer format or 0.282 in float format. This will ensure that you get the best adjacent channel performance. If floor noise at the output is more important than adjacent channel power a higher RMS value can be used, provided the peaks of the signal do not exceed the DAC full scale. With waveforms that have a high crest factor, it may be that the RMS value has to be lower so that the peak value is not exceeded.

The RMS value of the waveform may be calculated using where N is the number of IQ samples.

$$y_{rms} = \sqrt{\frac{\sum_{n=0}^{n=N-1}(I_n + jQ_n)(I_n - jQ_n)}{N}}$$

The required scale factor may be determined by dividing the required RMS value by the actual RMS value.

Waveform continuity

The previous sections have given a brief description on how one may generate and download a waveform. It has not discussed any constraints that are placed upon a waveform that is being continuously repeated by an arbitrary signal generator. To prevent periodic spectral re-growth it is important to ensure that there is no discontinuity between the start and end of a waveform. This is not a trivial matter and is very dependent upon the type of waveform that is been generated. This section gives some guidelines that can help to ensure that the waveform you generate is cyclic.

The simplest example to consider is the generation of a single side-band carrier. In this case we would generate a cosine waveform for the I vector and a sine waveform at the same frequency for the Q vector.

Let's assume that we want to generate a side-band at 10 kHz and that we intend to sample this at 45 MHz. To ensure that the waveform is repeatable we must have an integer number of complete cycles of the sinusoids. If we consider one period, this is equivalent to 0.0001 seconds or 4500 samples. In this case sample 4500 will occur at t = 0.0001-1/Fs seconds. The following sample (i.e. sample 0) will then be equal to the sample at t = 0.0001 which is what we need to ensure no discontinuity.

This is all very well for signals that have not been filtered. If a channel filter has been used, we have to ensure that the filter delay has been taken into account and that the basic waveform is cyclic. In general this can be achieved by using a circular filter. The best way to appreciate this is to look at some examples. Firstly consider the waveform in Figure 6 which is a filtered I channel of a QPSK signal.

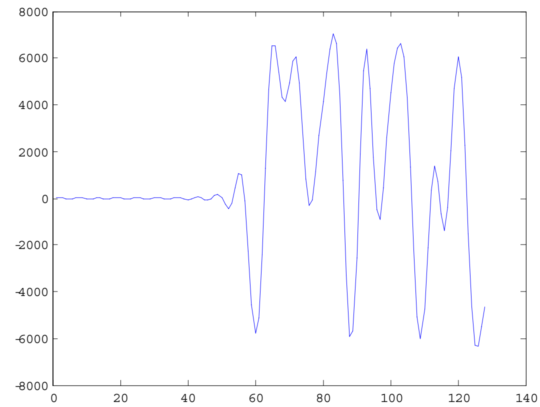


Figure 6 Filtered I channel of a QPSK signal.

From this figure we can see that if the above signal was continuously repeated there would be a discontinuity at the transition between the end and start of the waveform, as shown in Figure 7.

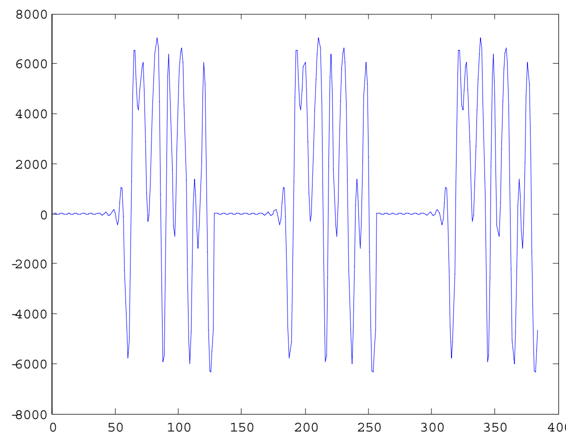


Figure 7 Wrapped version of original QPSK signal showing discontinuities.

The main reason for this is that the delay due to filtering has not been taken into account. One way to remedy this situation is to circularly filter the I and Q data. This may be done by repeating the input data to the filter for the duration of the filter delay and storing the full delayed version of the signal.

In other words, consider the plots in Figure 8. Figure 8a is the original input to the filter. Extend this by filter delay samples by appending the start samples to the end. This is shown in Figure 8b. Then filter this extended signal, the result is shown in Figure 8c. Once this has been done take the delayed version and save this as the cyclic waveform (Figure 8d). The repeated version of this signal is shown in Figure 9.

A Matlab function that does this is given below.

```
function y = cyclic_filter(filter_delay, h, x)
% This function will take an input signal, extend it by
% filter_delay samples. Then filter the extended input
% using the coefficients in h. The output y is the
% filtered input with the delay removed. In general the
% resulting waveform will be cyclic.

[m,n] = size(x);
if m > n
    x = x';
end
x_ext = [x,x(1:filter_delay)];
x_filt = filter(h,1,x_ext);
y = x_filt((filter_delay:end));
```

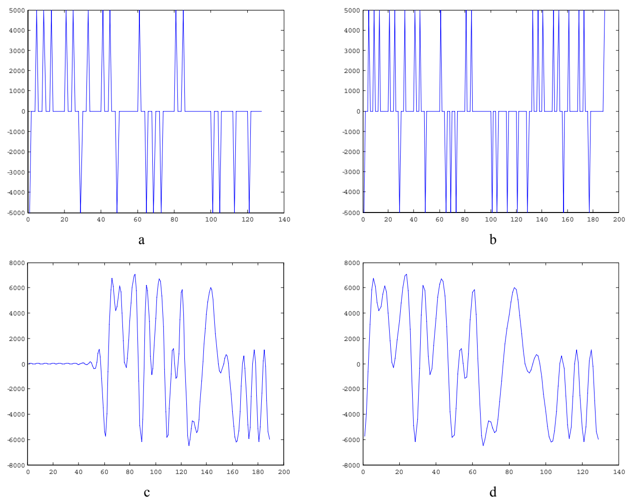


Figure 8 Filtering the signal to make the waveform cyclic.

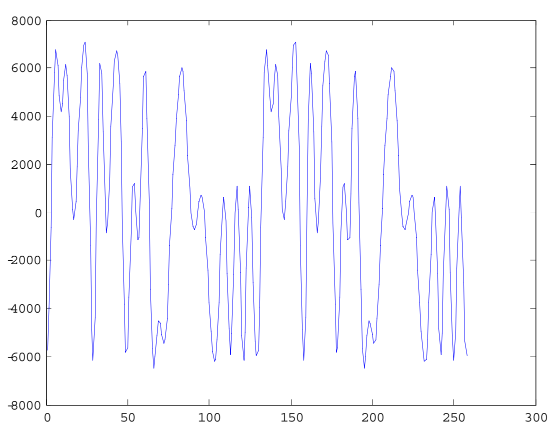


Figure 9 Cyclic waveform after filtering.

Obviously, not all waveforms can be made to be cyclic as shown in the above example. The topic is complex and each type of waveform will have to be analysed individually to ensure that it is cyclic. For example, pi/4 DQPSK waveforms can be cyclic in the sense that there is no spectral discontinuity, but there may be a symbol discontinuity. That is, a symbol transition from the end of the waveform to the start of the waveform may not be a valid one. In such cases, the number of symbols in the waveform will have to be considered, or the bit sequence will have to be chosen so that the transitions from the end to the start are valid. It is beyond the scope of this application note to go into further details on the subject, suffice to say that in general cyclic filtering will be adequate, but there are instances where a lot more thought will need to go into the waveform generation.

IFR, 10200 West York Street, Wichita, Kansas
67215-8999, USA. E-mail: info@ifrsys.com
Tel: +1 316 522 4981 Toll Free USA: 1 800 835 2352 Fax: +1 316 522 1360

IFR, Longacres House, Norton Green Road, Stevenage, Herts
SG1 2BA, United Kingdom. E-mail: info@ifrsys.com
Tel: +44 (0) 1438 742200 Freephone UK: 0800 282 388 Fax: +44 (0) 1438 727601

Part no. 46891/889
Issue 1