# Experimental ESR Pulse Optimisation: Feedback Control using an Arbitrary Waveform Generator

*D.L. Goodwin[1], W.K. Myers[2], C.R. Timmel[2], Ilya Kuprov[1]*
*[1]University of Southampton, UK; [2]University of Oxford, UK*

### Introduction

Pulsed magnetic resonance spectroscopy has well documented advantages over its continuous wave counterpart, including an increased number of variables to control specific interactions. When the excitation bandwidth exceeds the spectral width, hard pulses (Simple rectangular pulses) work well.

Pulses with variable amplitude and phase can be designed to achieve higher excitation bandwidth, and hence higher sensitivity, in the form of optimal control theory [1,2] or shaped pulses for broadband excitation [3-7]. The main problems in directly applying techniques from liquid state nuclear magnetic resonance (NMR) to electron spin resonance (ESR) are the time-scales and spectral widths involved; the gyromagnetic ratio of the electron spin is much more than that of the nuclear spin, and spectral widths range from tens of MHz to GHz. A hurdle in implementing optimal control pulses with an ESR spectrometer is that the pulses seen by the sample can be markedly different than those sent from the console.

Previous work to remedy this has been to create a transfer matrix (or response function), one that contains the information to transform a set of input pulses to those expected at the resonator. Progressive work on this first set out to measure the spectrometer's impulse response using a pick-up coil in the resonator [1], sending many sets of pulses through the microwave components causing linear and non-linear distortions, forming a transfer matrix from their measured responses from the pick-up coil. A further method for obtaining data for this transfer function is to use a sample with very narrow line-width [2].

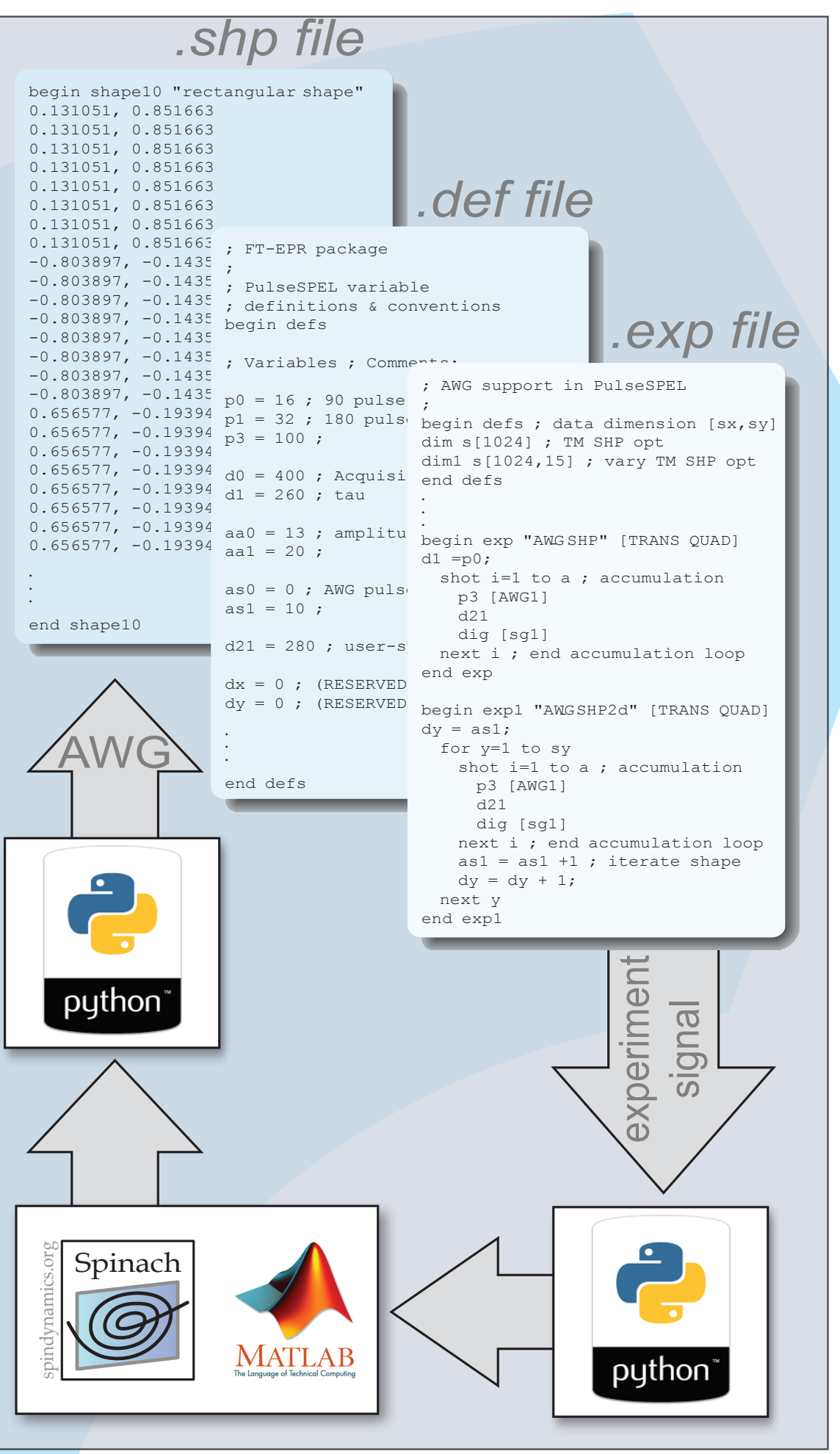
Using Bruker SpinJet AWG, with its commercial software, this manuscript intends to show how simple closed-loop feedback control can be used to optimize an ESR experiment such as echo experiments and Fourier-transform ESR experiments.

### Arbitrary Waveform Generator

The AWG allows the use of shaped microwave pulses with variable amplitude and phase. These shapes are compiled from a plain text .shp file into the current experiment, which can contain many shapes but is limited by the file size (262144 bytes). When a new shape file is introduced it must be newly compiled. There is a limit of compiling 115 shape files. After this the AWG reaches maximum working memory and the current experiment must be removed, re-compiled, then sent to the AWG.

Variable definitions are contained in a plain text .def file, such as pulse delays, amplitudes, interpolated shapes etc. These variable definitions are used by the plain text .exp file, setting out the experiment design. Most of the functional code of these files can be modified directly with Xepr API, based on Python scripting. Command line button presses are used to compile files.

The integration of Matlab with Python scripted Xepr API commands is set out on the right. The experiment signal is stored in a temporary plain text file to be imported into Matlab.



.shp file

```
begin shape10 "rectangular shape"
0.131051, 0.851663
0.131051, 0.851663
0.131051, 0.851663
0.131051, 0.851663
0.131051, 0.851663
0.131051, 0.851663
0.131051, 0.851663
0.131051, 0.851663
-0.803897, -0.1435
-0.803897, -0.1435
-0.803897, -0.1435
-0.803897, -0.1435
-0.803897, -0.1435
-0.803897, -0.1435
-0.803897, -0.1435
-0.803897, -0.1435
0.656577, -0.19394
0.656577, -0.19394
0.656577, -0.19394
0.656577, -0.19394
0.656577, -0.19394
0.656577, -0.19394
0.656577, -0.19394
0.656577, -0.19394
.
.
.
end shape10
```

.def file

```
; FT-EPR package
;
; PulseSPEL variable
; definitions & conventions
begin defs

; Variables ; Comments:

p0 = 16 ; 90 pulse
p1 = 32 ; 180 pulse
p3 = 100 ;

d0 = 400 ; Acquisi
d1 = 260 ; tau

aa0 = 13 ; amplitu
aa1 = 20 ;

as0 = 0 ; AWG puls
as1 = 10 ;

d21 = 280 ; user-s

dx = 0 ; (RESERVED
dy = 0 ; (RESERVED
.
.
.
end defs
```

.exp file

```
; AWG support in PulseSPEL
;
begin defs ; data dimension [sx,sy]
dim s[1024] ; TM SHP opt
dim1 s[1024,15] ; vary TM SHP opt
end defs
.
.
begin exp "AWGSHP" [TRANS QUAD]
d1 =p0;
  shot i=1 to a ; accumulation
    p3 [AWG1]
    d21
    dig [sg1]
  next i ; end accumulation loop
end exp

begin exp1 "AWGSHP2d" [TRANS QUAD]
dy = as1;
  for y=1 to sy
    shot i=1 to a ; accumulation
      p3 [AWG1]
      d21
      dig [sg1]
    next i ; end accumulation loop
    as1 = as1 +1 ; iterate shape
    dy = dy + 1;
  next y
end exp1
```

### A simple strategy to overcome pulse shape distortions

Existing methods require a transfer matrix [2,4,5], pre-calculated to allow for distortions a waveform will suffer being sent through electronic equipment. Without prior in-depth knowledge of the spin system, this work uses feedback control with a gradient free optimiser to attempt to find a “good” signal.

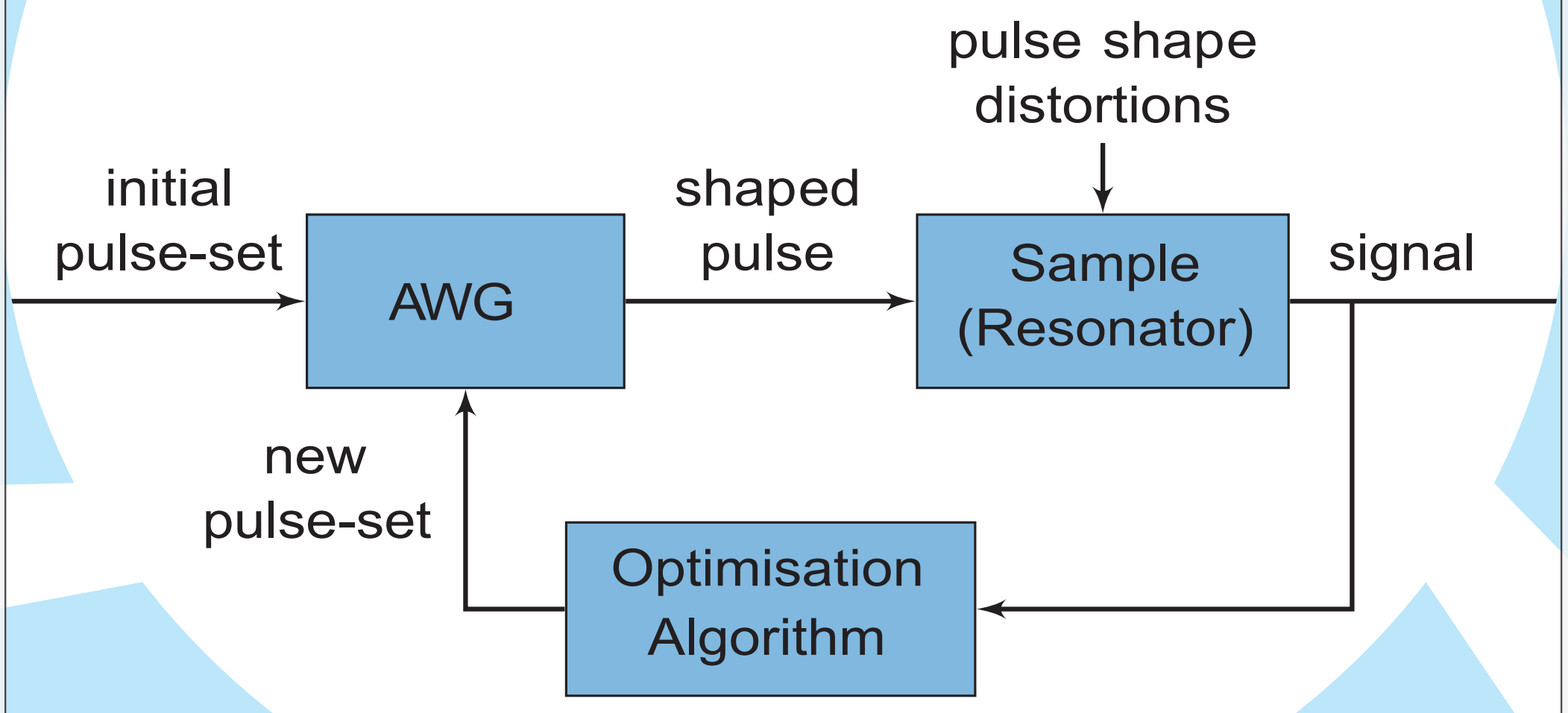
Using the software cycle indicated (left), a real-time optimisation does not need to deal with the complications of pulse shape distortions, these become part of the cost function itself. Although a background noise may seem problematic for an optimiser, the simplex method outlined in this text is quite resilient to these perturbations.

There are some issues with using a simplex method e.g. its inability to deal with bounded or discretevariables. The first of these can be overcome by realising that the functional space can be repeated. As an example, the shape file is bounded by +1 and -1, but the matlab code, `abs(mod(waveform-1,4)-2)-1`, ensures a reapeating of the functional space (where the waveform is the vector array of the shaped pulse points).

The PulseSPEL definitions file does have some discretised variables, e.g. pulse delays must be in units of 1ns. Although there is no real solution to this problem, if care is taken to ensure variables are normalised with a sensible normalisation constant, the optimiser can cope until it gets near a minimum (this is probably masked by the background noise).

### Finding Efficiencies

There are two sets of variables available to the optimisation algorithm: PulseSPEL variables (definitions file) and the shape of each pulse (shape file). PulseSPEL variables can be modified directly and this can be quite quick. The shape file must go through a number of steps, including compiling, to be sent to the AWG - this can be quite a slow process. A shape file can contain more than one shape (limited by the size of the shape file) and it is recommended to process many shapes in a 2d experiment (shapes need to be independent of each other).



### A Metric to Optimise

Any optimisation requires a figure of merit (also called a cost functional), being a measure of that which we would minimise/maximise. Below are a number of these measures already in use [1-7]:

*Optimal ESR Echo* ⟶ *integral of the echo*
*Optimal Fourier Transform ESR* ⟶ *curve matching to reference*
*Optimal ESEEM* ⟶ *modulation depth*

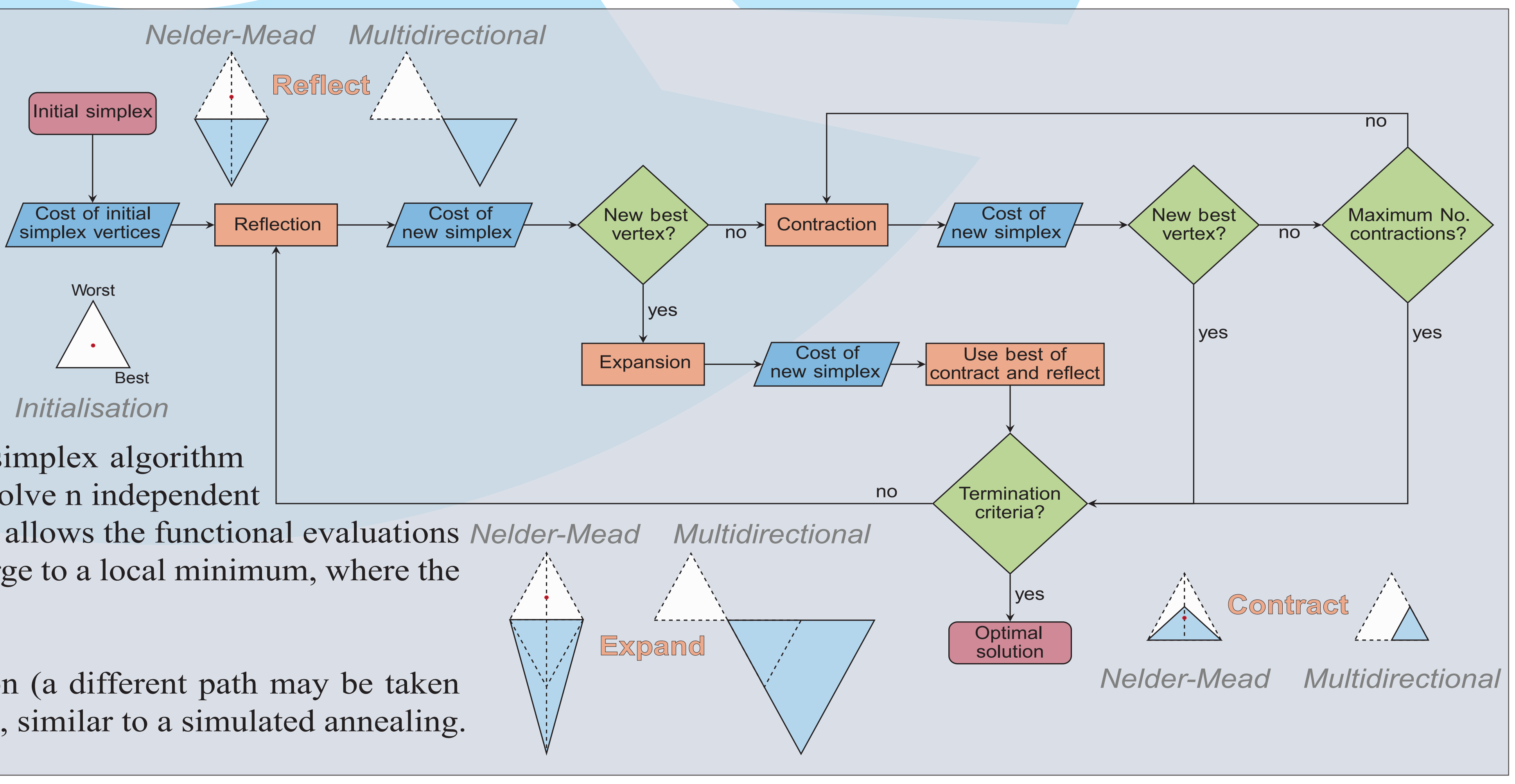
### Gradient-free Optimisation

“*There is no such thing as a free lunch*” where gradient-free optimisation is concerned, especially when a functional evaluation is expensive (performing an experiment). A popular algorithm in this context is the ***Nelder-Mead*** simplex algorithm [8], where a simplex is a polytope of size dimension n+1, n being the number of variables the optimisation can control. Essentially, the algorithm wanders around the functional space according to its algorithmic rules (right), accepting new vertices of a new simplex if one is better than the best vertex of the previous simplex.

### Parallel Functional Evaluation

An extension to the ***Nelder-Mead*** algorithm, named the ***multidirectional-search*** simplex algorithm [9], has a different set of processes to construct a new simplex. These processes involve n independent functional evaluations (where one is required for Nelder-Mead). This independence allows the functional evaluations to take place in parallel. This algorithm has better convergence, in that it will converge to a local minimum, where the standard Nelder-Mead algorithm may never converge.

Evaluating all reflections, expansions, and contractions gives further parallelisation (a different path may be taken within the optimisation), and a number of iterates can be evaluated at the same time, similar to a simulated annealing.

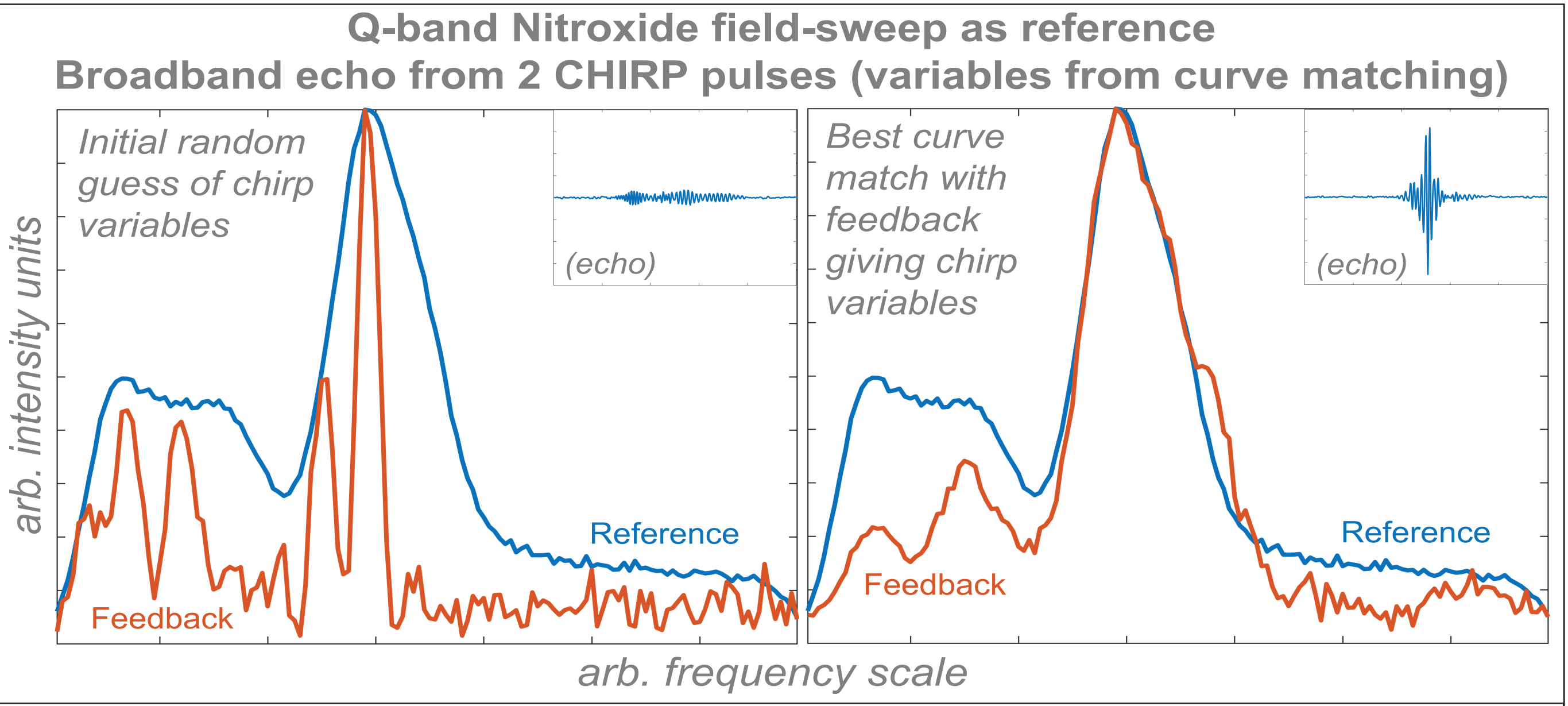


### Experimental Results

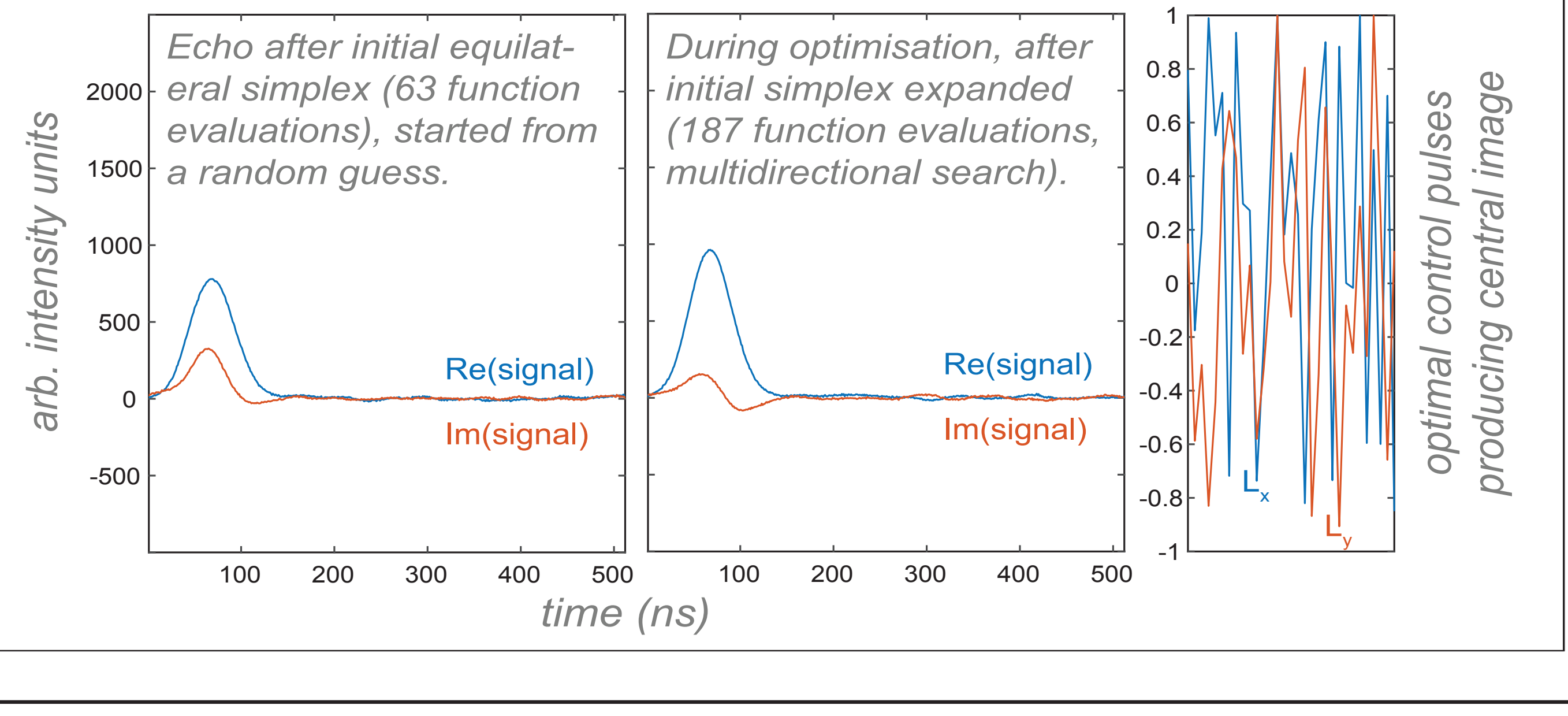
Feedback control is used with curve matching as a figure of merit for broadband excitation with a field-sweep as a reference curve. Both curves, feedback and reference, are normalised to their highest intensity and a fit of one to the other can be thought of as the outer-product of the two vector arrays plotting the shape of the curve i.e. unity if they are the same curve, and zero if they are orthogonal. Results are encouraging, although the simple ***Nelder-Mead*** algorithm failed to converge - seemingly to fight with different parts of the curve equally. Part of the problem is that a course method was used to find the left and right extremities of the feedback curve. possible solutions are to calculate the matching sections at the start, or to remove background noise from both curves.

The second set of results is for feedback control using the echo integral as a figure of merit. The optimisation algorithm alternates between the ***Nelder-Mead*** and ***multidirectional search***, each allowed 300 functional evaluations. Although the ***Nelder-Mead*** method made the best progress, the ***multidirectional search*** allowed convergence after a total of a few thousand functional evaluation - the optimal solution gave exactly the same integral as that with hard pulses but with less power.

Q-band Nitroxide field-sweep as reference
Broadband echo from 2 CHIRP pulses (variables from curve matching)



Q-band Coal echo - hard-pulse followed by a 32 point, shaped pulse



### References

[1] Spindler et. al., J. Mag. Reson., 218, 49 (2012)
[2] Kaufmann et. al., J. Mag. Reson., 235, 95 (2013)
[3] Spindler et. al., Angew. Chem. Int. Ed., 52, 3425 (2012)
[4] Doll et. al., J. Mag. Reson., 230, 27 (2013)
[5] Doll, Jeschke, J. Mag. Reson., 246, 18 (2014)
[6] Schöps et. al., J. Mag. Reson., 250, 55 (2015)
[7] Jeschke, Pribitzer, Doll, J. Phys. Chem. B, 119, 12570 (2015)
[8] Nelder, Mead, Computer Journal, 7, 308 (1965)
[9] Torczon, Journal on Optimization, 1, 123 (1991)

### Acknowledgements

The authors are grateful to Arzhang Ardavan and Gunnar Jeschke for helpful and insightful discussions. This work is supported by EPSRC (iMR-CDT doctorial training centre), EU FP7 (QUAINT), and EPSRC grant to Centre for Advanced Spin Resonance EP/L011972/1.