# From Binary to Continuous Gates – and Back Again

Matthias Bechmann*[1], Angelika Sebald[1], and Susan Stepney[2]

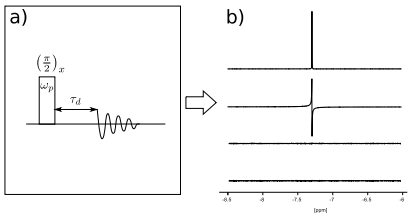
[1] Department of Chemistry, University of York, YO10 5DD, UK
[2] Department of Computer Science, University of York, YO10 5DD, UK

**Abstract.** We describe how nuclear magnetic resonance (NMR) spectroscopy can serve as a substrate for the implementation of classical logic gates. The approach exploits the inherently continuous nature of the NMR parameter space. We show how simple continuous NAND gates with sin/sin and sin/sinc characteristics arise from the NMR parameter space. We use these simple continuous NAND gates as starting points to obtain optimised target NAND circuits with robust, error-tolerant properties. We use Cartesian Genetic Programming (CGP) as our optimisation tool. The various evolved circuits display patterns relating to the symmetry properties of the initial simple continuous gates. Other circuits, such as a robust XOR circuit built from simple NAND gates, are obtained using similar strategies. We briefly mention the possibility to include other target objective functions, for example other continuous functions. Simple continuous NAND gates with sin/sin characteristics are a good starting point for the creation of error-tolerant circuits whereas the more complicated sin/sinc gate characteristics offer potential for the implementation of complicated functions by choosing some straightforward, experimentally controllable parameters appropriately.

## 1 NMR and Binary Gates

Nuclear magnetic resonance (NMR) spectroscopy in conjunction with nonstandard computation usually comes to mind as a platform for the implementation of algorithms using quantum computation. Previously we have taken a different approach by exploring (some of) the options to use NMR spectroscopy for the implementation of classical computation [5]. We have demonstrated how logic gates can be implemented in various different ways by exploiting the spin dynamics of non-coupled nuclear spins in a range of solution-state NMR experiments. When dealing with spin systems composed of isolated nuclear spins, the underlying spin dynamics can be described conveniently by the properties of magnetisation vectors and their response to the action of radio-frequency (r.f.) pulses of different durations, phases, amplitudes and frequencies. Together with the integrated intensities and/or phases of the resulting NMR signals, this scenario provides a

* Corresponding author: mb577@york.ac.uk




| Input 1 (selective pulse offset $\omega_p$) | Input 2 (delay$\tau_d$) | Output |
|---|---|---|
| $\omega_p = \omega_{iso}^{CS}$ (0) | $\tau_d = 0$ (0) | 1 |
| $\omega_p = \omega_{iso}^{CS}$ (0) | $\tau_d \neq 0$ (1) | 0 |
| $\omega_p \gg \omega_{iso}^{CS}$ (1) | $\tau_d = 0$ (0) | 0 |
| $\omega_p \gg \omega_{iso}^{CS}$ (1) | $\tau_d \neq 0$ (1) | 0 |

**Fig. 1.** NOR gate implemented using NMR. a) NMR pulse sequence. b) Spectra corresponding to the four possible gate outputs where the integrated spectral intensity is mapped to logic outputs 0 and 1. c) Logic truth table mapping NMR parameters to gate inputs 0 and 1. (adapted from [5])

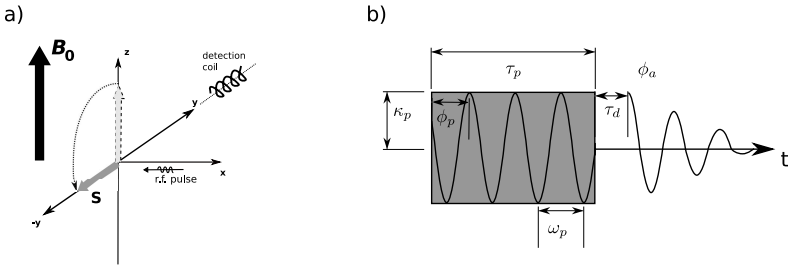



**Fig. 2.** Magnetisation vector manipulation by r.f. pulses, e.g. rotation of magnetisation vector $\boldsymbol{S}$ from the $z$-direction to the $-y$-direction by a suitable r.f. pulse (a)). Structure of a r.f. pulse displaying characterisation parameters for amplitude, frequency, duration and phase as possible gate input controls (b)).

rich parameter space and a correspondingly large degree of flexibility regarding choices of input and output parameters for the construction of logic gates. Fig. 1 shows an NMR implementation of a NOR gate, for illustration.

The effects of r.f. pulses on a given nuclear spin system are fully under experimental control, and the response of the spin system is fully predictable with no approximations involved. An NMR experiment usually starts from the magnetisation vector in its equilibrium position: aligned with the direction of the external magnetic field (the $z$-direction in the laboratory frame). An r.f. pulse tips the magnetisation vector away from the $z$-direction. By choosing the duration, amplitude and frequency of the pulses appropriately, the tip of the magnetisation vector can be used to sample the entire sphere around its origin (Fig. 2).

Our previous NMR implementations of logic gates [5] exploited special positions on this sphere, such as NMR spectra corresponding to the effects of 90°, or 180°, or 45° pulses to create binary input/output values. We have demonstrated that there are many different ways for such implementations of conventional logic gates by slightly less conventional NMR implementations, including many different ways to define input and output parameters. There are many more possibilities for NMR implementations of conventional logic gates and circuits. Note

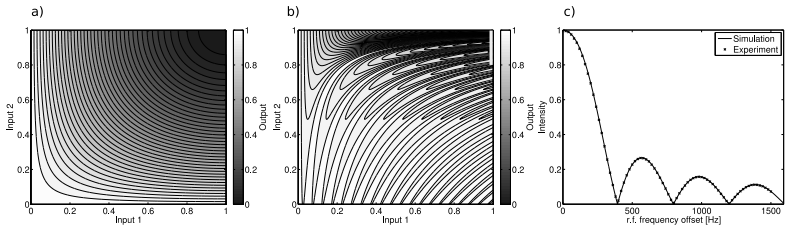



**Fig. 3.** 2D function graphs displaying influence of NMR parameters on the output of continuous NAND gates. a) Using the duration $\tau_{\mathrm{p}}$ of the r.f. pulse and the duration of a preacquisition delay $\tau_{\mathrm{d}}$, resulting in sin dependence of both inputs. b) Using the resonance frequency offset $\omega_{\mathrm{p}}$ and the r.f. pulse duration $\tau_{\mathrm{p}}$, a sinc dependence for $\omega_{\mathrm{p}}$ and a sin dependence for $\tau_{\mathrm{p}}$ is obtained. c) Comparison of experimental and theoretical result for a slice of sinc/sin NAND gate (in b) without mapping to the $[0, 1]$ interval. This corresponds to the region in b) marked by the vertical bar in upper right corner. The deviation between experiment and simulation is always less than 0.5 percent.

that for these discrete logic gates a one-to-one mapping of the NMR parameter(s) to the binary state of the gate is possible in a straightforward manner.

In this paper we concentrate on another aspect of NMR implementations of classic logic gates. Whereas previously our main focus was on the multitude of different options for implementing discrete logic gates and circuits by NMR, here we exploit another property of basic NMR experiments. Only a minute fraction of, for example, the space accessible to the magnetisation vector has so far been exploited for the construction of discrete logic gates. Now we lift this restriction and take advantage of the inherent continuous properties of our system and the natural computational power provided by the system itself [6]. The underlying *continuous* spin dynamics hereby provide the basis to the implementation of *continuous* logic operations. Compared to [5] this means we no longer restrict the inputs and outputs to be the discrete values 0 and 1, but allow them to be continuous values between 0 and 1.

## 2 Functions of NMR and Continuous Gates

Depending on the position of the magnetisation vector at the start of signal acquisition, the time-domain NMR signal is composed of sin and cos functions, with an exponentially decaying envelope (the so-called free induction decay, FID). Accordingly, trigonometric and exponential functions are two of the continuous functions inbuilt in any NMR experiment. Most commonly, NMR signals are represented in the frequency domain. Hence, Fourier transformation gives access to, for example, the sinc function $((\sin x)/x)$ if applied to a truncated exponential decay. Fig. 3 illustrates this shift to continuous logic gates: we show the NMR implementation of NAND gates where the inputs have functional dependencies of sin/sin (Fig. 3a) and sin/sinc (Fig. 3b). Note how they have the same digital NAND gate behaviours at the corners $\{0, 1\} \times \{0, 1\}$, but very different

behaviours in between. Fig. 3c shows experimental NMR data representing the sinc function used in Fig. 3b.

Taking the step to continuous gates, the input/output mapping now applies to the $[0, 1]$ *interval* and is not as trivial as it is for the discrete logic gates. However, the NMR input parameters and output functions are known in analytical form, giving access to boolean behaviour at the corners of the two-dimensional parameter space, and continuous transitions in between.

The digital NAND gate is universal. Here we relax the constraints on the inputs, to form our continuous NAND gates. These continuous gates can serve as starting points for the optimisation of certain properties of the NAND gate itself or, alternatively, for the optimisation of circuits based on NAND gates. We show how to obtain robust NAND gates (ones that still function as digital NAND gates, even if the inputs have considerable errors), by evolving circuits of the continuous single NAND gates with sin/sin (Fig. 3a) and sin/sinc characteristics (Fig. 3b). Then we evolve circuits for a robust XOR gate, constructed from continuous simple NAND gates. Finally, we briefly address the topic of more general continuous gates based on different functions [2] and how the naturally occurring continuous NMR functions may be exploited in such circumstances. Our optimisation tool is Cartesian Genetic Programming (CGP) [3].

## 3 Evolving Robust Continuous Gates and Circuits

### 3.1 Continuous NAND Gate with sin/sin Characteristics

This continuous gate is based on the NMR parameters $\tau_{\mathrm{p}}$ (pulse duration) and $\tau_{\mathrm{d}}$ (preacquisition delay) (see Figs. 2b and 3a). It involves the following mapping of the NMR input parameters $In_1$ and $In_2$:

$$\begin{gathered} In_1, In_2 \in [0, 1] \\ In_1 = \tfrac{\tau_{\mathrm{p}}}{\tau_{\mathrm{p90}}} \quad ; \quad In_2 = 1 - \tfrac{\tau_{\mathrm{d}}}{\tau_{\mathrm{d90}}} \end{gathered} \tag{1}$$

where $\tau_{\mathrm{p90}}$ corresponds to a pulse duration causing a 90° flip of the magnetisation vector and $\tau_{\mathrm{d90}}$ is the duration of a preacquisition delay causing a 90° phase shift of the magnetisation vector in the $xy$-plane. The output of the simple sin/sin NAND gate implemented by the NMR experiment is then

$$Out = 1 - \sin\left(\tfrac{\pi}{2} In_1\right) \sin\left(\tfrac{\pi}{2} In_2\right) \tag{2}$$

Our target robust NAND gate is shown in Fig. 4a. It is a continuous gate, with discrete state *areas* which, accordingly, should represent an error-tolerant, robust gate. The sampling points used to define the fitness function for evolving this robust gate are shown in Fig. 4b. The fitness function $f$ defined over these $N$ sampling points is

$$f = \sum_{i=1}^{N} \frac{1}{1 + \left|Out_i^{\mathrm{evo}} - Out_i^{\mathrm{target}}\right|} \tag{3}$$

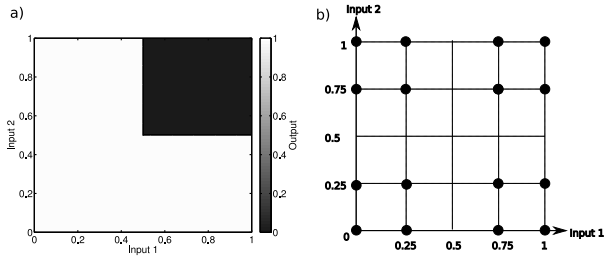



**Fig. 4.** a) Target robust NAND gate with discrete state *areas*. This is robust to errors in the inputs, yielding a correct digital NAND gate for inputs rounded to 0 or 1. b) Sampling points used in the fitness function to evolve the robust NAND gate.

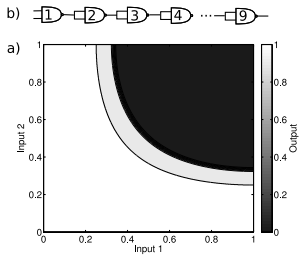



**Fig. 5.** a) Functional behaviour of the array of nine continuous sin/sin NAND gates. b) Optimisation result being a linear array of nine continuous sin/sin NAND gates.

The evolved robust NAND gate is shown in Fig. 5 (see §7 for the CGP parameters used). It displays the desired feature of well-defined, discrete state areas. The behaviour towards the centre differs from Fig. 4a, but provides no contribution to the fitness function. The evolved circuit for the robust NAND gate is a linear array of nine simple NAND gates (Fig. 5b). With increasing lengths of the NAND-gate chains, the resulting circuit for the robust gate becomes fitter. Odd length chains converge to the robust NAND gate behaviour, whereas even-length chains converge toward a corresponding robust AND gate. This is illustrated in Fig. 6. The first simple NAND gate in the chain performs the NAND operation; all the remaining gates, with their paired inputs, act as simple NOT gates. The increasing length chain converges to fitter circuits, because of the S-shaped $(1 - \sin^2 \frac{\pi}{2} x)$ form of the sin/sin gate along its $x = y$ diagonal: any value passing through a pair of simple NOT gates moves closer to being 0 or 1, and so converges to 0 or 1 as the chain of simple NOT gates lengthens. The maximum displacement of points by a single NOT gate operation towards 0 or 1 is $\approx 0.11$. This can be interpreted as a threshold for the convergence and stability of the array. Random fluctuations added numerically to every gate output in the range of $[\pm 0.1]$ do not hinder the convergence of the array (Fig. 6 last column). For rather large error values ($> 0.2$) the arrays tend to destabilise, especially for longer arrays.

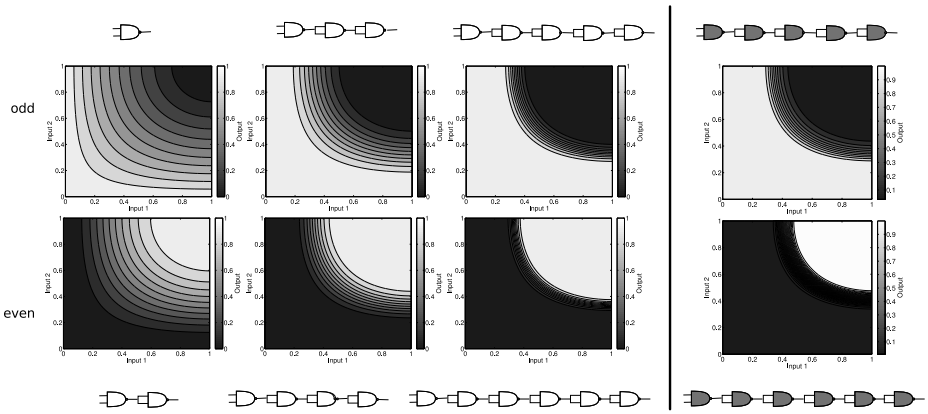



**Fig. 6.** Convergence of theoretical NAND gate arrays. Odd-numbered arrays converge toward target NAND gate (top row), even-numbered arrays (bottom row) converge toward a corresponding AND gate. The final circuit in each row displays the stability of the array convergence under erroneous signal transduction between gates, assuming random fluctuations in the range of [±0.1].

There are two possible sources for experimental imperfection and therefore imperfect gate behaviour: the accuracy by which the experimental NMR parameters $(\omega_\mathrm{p}, \tau_\mathrm{p}, \ldots)$ can be executed by the NMR hardware; and the accuracy by which the NMR spectra can be acquired and analysed (integrated in this case). A comparison shows that the fluctuations caused by the measurement and analog-digital conversion are by far the dominating factors (e.g. pulses used were of duration 2.5 ms ±50 ns [1], while fluctuations in signal intensity were $< \pm 0.5\%$).

### 3.2 Continuous NAND Gate with sin/sinc Characteristics

We now consider circuits based on the continuous simple sin/sinc NAND gate (Fig. 3b), again aiming for the target robust NAND gate with discrete state areas (Fig. 4a). Here mapping of the NMR parameters $\omega_\mathrm{p}$ (r.f. pulse frequency offset) and $\tau_\mathrm{p}$ (r.f. pulse duration) is the following

$$In_1 = \frac{\tau_\mathrm{p}}{\tau_{\mathrm{p}90}} \quad ; \quad In_2 = 1 - \frac{\omega_\mathrm{p}}{\omega_{\mathrm{p}_\mathrm{max}}} \tag{4}$$

where $\omega_{\mathrm{p}_\mathrm{max}}$ is the maximum allowed r.f. frequency offset (minimum of sinc function).

The output of the simple sin/sinc NAND gate implemented by the NMR experiment is then

$$Out = 1 - \frac{|\kappa_{\mathrm{p}90}| \sqrt{\kappa_{\mathrm{p}90}^2 \sin^2(\omega_\mathrm{eff}\tau_\mathrm{p}) + 2\omega_\mathrm{p}^2 (1 - \cos(\omega_\mathrm{eff}\tau_\mathrm{p}))}}{\omega_\mathrm{eff}^2} \tag{5}$$

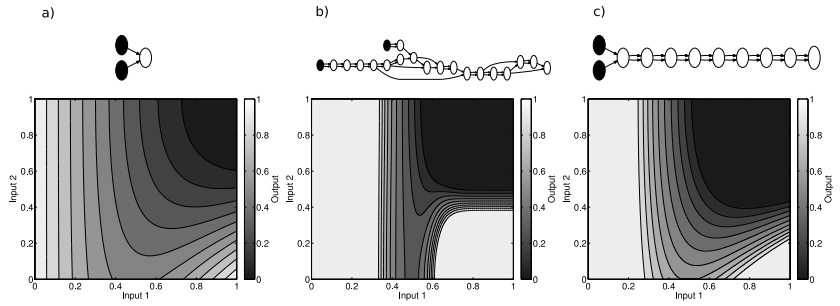



**Fig. 7.** a) Initial simple sin/sinc NAND gate with one minimum included. b) CGP evolved result. c) Array of nine simple continuous sin/sinc NAND gates.

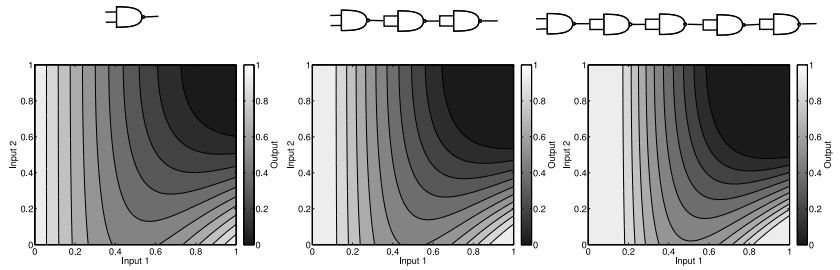



**Fig. 8.** Convergence of one-minimum sin/sinc NAND gate chains for increasing (odd-numbered) chain length.

where $\omega_{\text{eff}} = \sqrt{\omega_{\text{p}}^2 + \kappa_{\text{p90}}^2}$ assuming a perfect $\pi/2$ magnetisation flip for an on-resonance r.f. pulse of amplitude and duration $\kappa_{\text{p90}}$ and $\tau_{\text{p90}}$ respectively.

The continuous sin/sinc NAND gate is a more complicated situation because it does not display symmetry along the diagonal, in contrast to the sin/sin NAND gate. We approach evolution of a robust NAND circuit based on simple sin/sinc NAND gates in a step-wise manner.

**Gate Confined to Include only the First Minimum of the sinc Function**
To start with, we use a simple sin/sinc NAND gate confined to include only the first minimum of the sinc function (Fig. 7a).

Fig. 7b shows the CGP evolved result, a robust arrangement of discrete state areas. The evolved circuit shown at the top of Fig. 7b is more complicated than the linear chain of NAND gates previously found in the circuit based on simple sin/sin NAND gates. If we build such a linear circuit from simple sin/sinc NAND gates we do find an acceptable solution (Fig. 7c), but with slightly poorer fitness. Despite the loss of symmetry of our sin/sinc starting NAND gate, repeated application of linear chains of increasing lengths still converges to the desired behaviour (Fig. 8).

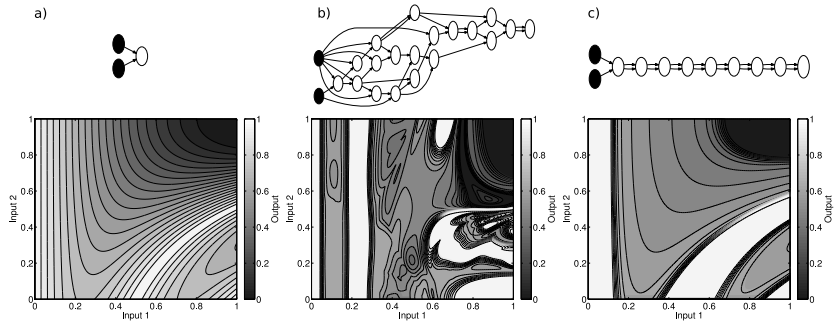



**Fig. 9.** a) Initial simple sin/sinc NAND gate with two minima included. b) CGP evolved result. c) Array of nine simple continuous sin/sinc NAND gates.

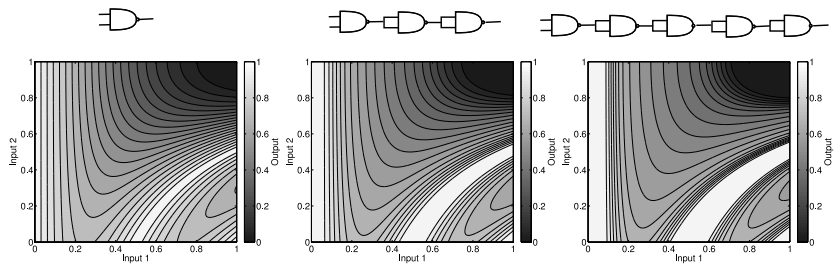



**Fig. 10.** Convergence of two-minima sin/sinc NAND gate chains for increasing (odd-numbered) chain length.

**Gate Confined to Include the Second Minimum of the sinc Function** Next, we use a simple sin/sinc NAND gate confined to include the first two minima of the sinc function (Fig. 9a).

Again, we compare the result of a CGP evolution (Fig. 9b) and the result of applying the linear array of nine simple sin/sinc NAND gates (Fig. 9c). CGP is successful in finding a solution which is fairly well optimised around the 16 sampling points (Fig. 4b), but the areas in between now display less obvious and more complicated characteristics.

The linear chain of nine simple sin/sinc NAND gates is here slightly less successful finding a good solution at and around the sampling points, but a pattern relating to the number of minima in the starting gate is emerging. With only one minimum included, there are essentially just two levels in the contour plot (Fig. 7c). Now, with two minima included, we find three distinct levels (around 0, around 0.5, and around 1; see Fig. 9c), separated from each other by steep steps. Fig. 10 shows the results of repeated application of linear arrays of simple sin/sinc NAND gates of increasing length. One can see how for the application of longer chains the terraced structure and step functions converge.

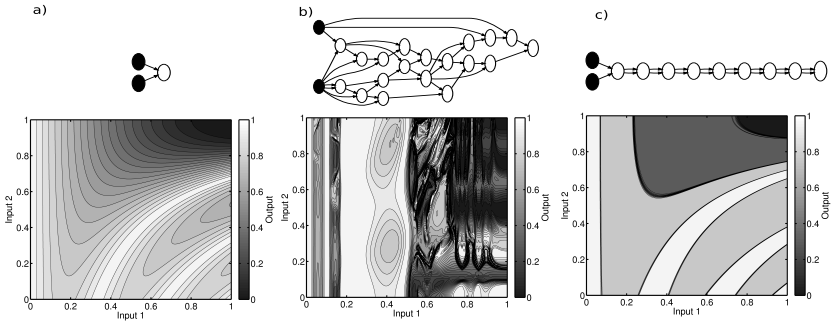



**Fig. 11.** a) Initial simple sin/sinc NAND gate with three minima included. b) CGP evolved result. c) Array of nine simple continuous sin/sinc NAND gates.

**Gate Confined to Include the Third Minimum of the sinc Function** Fig. 11 summarises the results when we include three minima of the sinc function in our starting sin/sinc NAND gate.

CGP again evolves a solution which is optimised around all 16 sampling points (Fig. 11a), but with even more complicated behaviour in between. The (unevolved) linear chain of sin/sinc NAND gates now creates four distinct levels and an overall stepped structure, but is less fit with respect to the fitness function sampling points of Fig. 4b.

From these results, we can see that continuous simple sin/sinc NAND gate can act as a good starting point for the implementation of a variety of complicated functions, simply by choosing the number of minima included appropriately for the starting continuous gate, and by defining a suitable number of sampling points.

## 4 Evolving XOR Circuits Using NAND Gates

Here we briefly demonstrate that this strategies used for evolving robust NAND circuits can also be used to obtain circuits with other functionality built from simple NAND gates. We use the continuous simple sin/sin NAND gate (Fig. 3a) as the starting point. Our target circuit is a robust XOR gate with discrete state areas (Fig. 12a), with the same 16 sampling points as before.

An XOR gate constructed from simple sin/sin NAND gates (the grey region of Fig. 12b) gives the continuous behaviour shown in Fig. 13a. If this is followed by our previously discovered strategy of a chain of simple NAND gates (Fig. 12b), we get the result shown in Fig. 13b: a robust XOR gate.

If we use CGP to evolve a solution from scratch, we get the more complicated circuit shown in Fig. 12c, with fitter continuous behaviour (Fig. 13c). Note that evolution here rediscovers the chaining strategy, and applies it to the final part of the circuit.

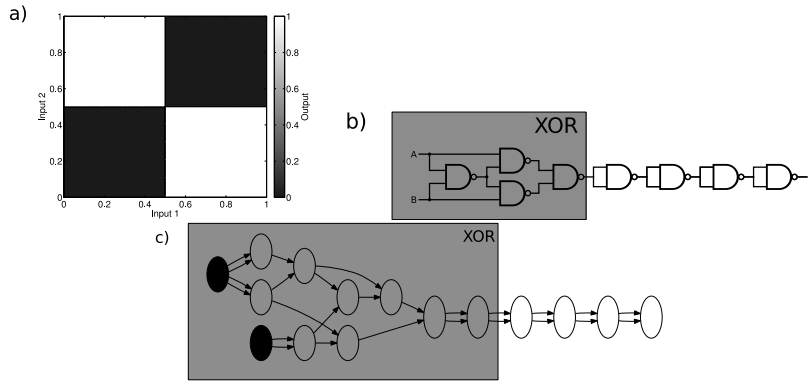



**Fig. 12.** a) The target XOR gate with discrete state areas. b) Applying the NAND-gate chain approach for optimisation. c) CGP evolved circuit.

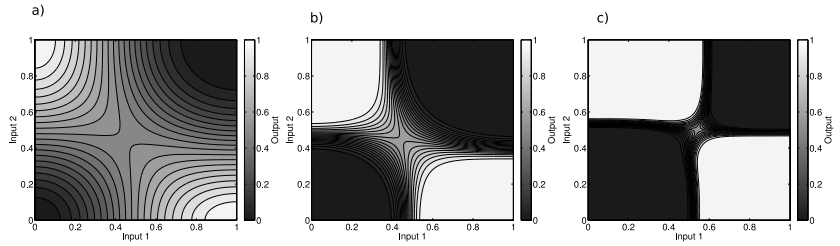



**Fig. 13.** a) XOR gate built from continuous NAND gates without optimisation. b) Result of NAND-gate chain approach. c) CGP evolved XOR gate.

## 5 Truly Continuous Gates

So far we have been using the continuous behaviour of the simple gates to implement robust, but still essentially digital, gates. In this section we use a different fitness function to evolve circuits with interesting truly continuous behaviour.

We can make boolean logic continuous on the interval [0,1] by defining $\mathrm{AND}(a, b) = \min(a, b)$ and $\mathrm{NOT}(a) = 1 - a$ (see [2]). These have the digital behaviour at the extreme values. Then $\mathrm{NAND} = 1 - \min(a, b)$ (Fig. 14a).

We start from the continuous simple sin/sin NAND gate (Fig. 14b). At first glance this seems to be a more straightforward optimisation task than for the robust gates, given that both the starting gate and the target function are continuous in nature, with a similar initial structure. Here we take a fitness function sampled over more points in the space, using a regular grid of $6 \times 6$ points.

The evolved result is shown in Fig. 14c, together with the corresponding, rather elaborate, circuit. Here the more complex circuit yields only modest improvements over the simple gate, with agreement between target and evolved function improving by about a factor 2 over that of the single simple sin/sin NAND gate. In particular, the evolved circuit does not really help to improve

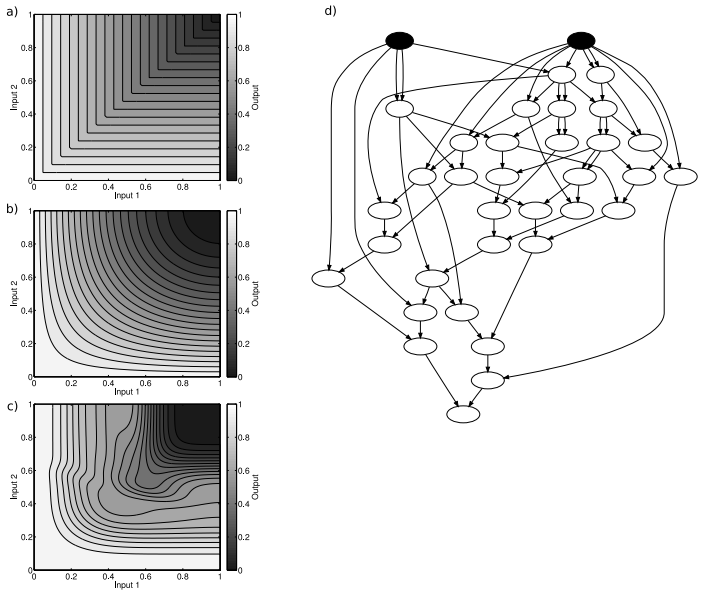



**Fig. 14.** a) The target NAND gate where NAND $= 1 - \min(a,b)$. b) The initial simple sin/sin NAND gate. c) The CGP evolved gate. d) The CGP evolved circuit (5% mutation rate, population size 500, best fitness 35.25, 10000 generations).

agreement with the most prominent feature of the target function, the sharp diagonal ridge. More work is needed to match the natural properties provided by the NMR system with the desired properties of the continuous gates.

Fig. 15 shows the truly error tolerant behaviour of the CGP evolved gate in Fig. 14c.

## 6 Conclusions and Next Steps

CGP has proved effective at evolving specific continuous circuits from the continuous simple NAND gates provided by our NMR approach. In particular, the simple sin/sinc gates can provide a rich set of disctretised behaviours.

In these experiments, neither the robust gates, nor the truly continuous gates, are inspired by the natural properties of the NMR system, but rather by mathematical abstractions. Next steps will involve investigating and exploiting what the simple NAND gates "naturally" provide.

## 7 Experimental Setup

**Evolutionary Setup.** We use a modified version of the CGP code of [4]. Our setup uses a linear topology of 60 nodes plus input and output nodes with

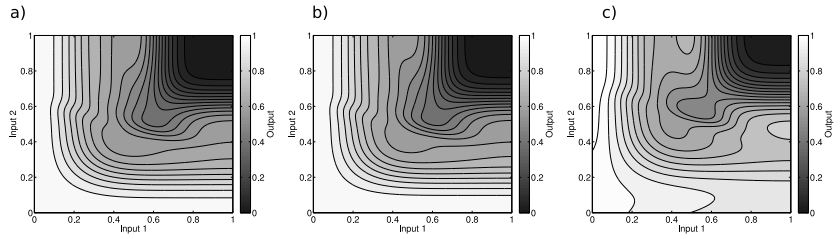



**Fig. 15.** Stability and error propagation through CGP evolved gate in Fig. 14c: with random error (a) [±0.5%]; (b) [±1%]; (c) [ ±10%]

the maximum number of level-back connections. Optimum results used between nine and 33 nodes. The mutation rate during evolution was varied between 0.5% and 50%, where rates between 5% and 10% performed best. Populations of 50/500 were evolved for 10000 generations. Results presented are the best of 10 evolutionary runs.

**NMR Spectroscopy.** $^1$H NMR spectra of 99.8% deuterated $CHCl_3$ (Aldrich Chemicals) were recorded on a Bruker Avance 600 NMR spectrometer, corresponding to a $^1$H Larmor frequency of −600.13 MHz. On-resonant 90° pulse durations were 2.5 ms and recycle delays 3 s.

Hardware limitations [1]: duration of r.f. pulses accurate to ±50 ns; pulse rise and fall times 5 ns and 4 ns respectively; pulse amplitude switched in 50 ns with a resolution of 0.1 dB; phases are accurate to ±0.006 degree and switched $< 300$ ns; r.f. range is 3–1100 MHz with a stability of $3 \cdot 10^{-9}$/day and $1 \cdot 10^{-8}$/year and a resolution of $< 0.005$ Hz. Frequency switching is $< 300$ ns for 2.5 MHz steps and $< 2\mu$s otherwise. Main source of experimental errors is integration error due to limited digitisation resolution, 0.5% maximum.

## Acknowledgements

We gratefully acknowledge the Leverhulme Trust for supporting this work. We thank Shubham Gupta, IIT Mumbai, India, for his cooperation in the initial stages of this work, supported by the TRANSIT project (EPSRC grant EP/F032749/1), and John Clark, York for continued discussions and comments.

## References

1. Butler, E.: NMR hardware user guide version 001. Tech. rep., Bruker Biospin GmbH, Rheinstetten, Germany (2005)
2. Levin, V.: Continuous logic - i. basic concepts. Kybernetes 29, 1234–1249(16) (2000)
3. Miller, J.F., Thomson, P.: Cartesian genetic programming. In: 3rd European Conference on Genetic Programming. LNCS, vol. 1802, pp. 121–132. Springer (2000)
4. Miller, J.: Cartesian genetic programming source code (July 2009), http://sites.google.com/site/millerjules/professional
5. Roselló-Merino, M., Bechmann, M., Sebald, A., Stepney, S.: Classical computing in nuclear magnetic resonance. Int. J. of Unconventional Computing 6(3–4) (2010)
6. Stepney, S.: The neglected pillar of material computation. Physica D: Nonlinear Phenomena 237(9), 1157–1164 (July 2008)