Homebrew

More on phasing methods for SSB.

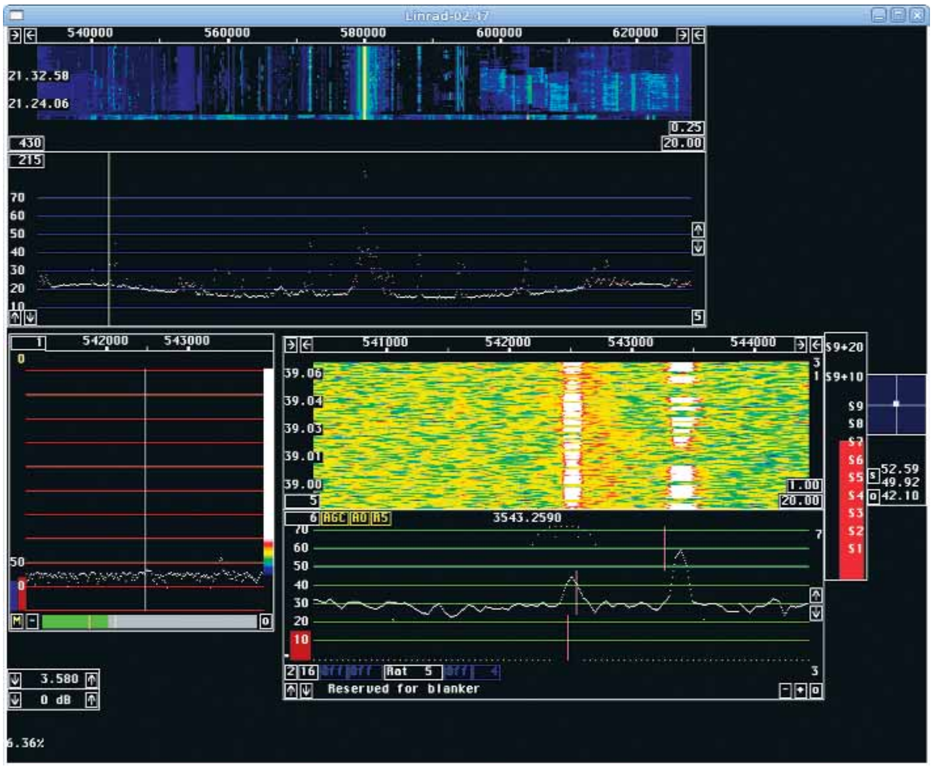


FIGURE 1: Screen capture of Linrad in action.

MORE PHASING. Following on from last month's experiments with simple phasing receivers and SDR, we will take a closer look at some of the available PC-based SDR software. We will also have a go at building some good old-fashioned analogue phasing equipment.

When most of the stages of a radio receiver or transmitter are implemented in software, it is possible to build a radio with very advanced features using very simple analogue hardware. Ideally, the only analogue circuitry employed would be limited to a low-pass filter to remove alias signals and a high performance ADC (analogue to digital converter). The RF signals would be sampled directly at the receiver front-end and converted to digital form. You can think of such a receiver as being equivalent to a PC sound card with a very, very high sampling rate and a maximum input frequency measured in MHz instead of kHz. Such receivers do exist [1] but they are still considerably more expensive than simple I/Q mixer systems. The need for high dynamic range calls for a very high performance ADC. The minimum number of bits of ADC resolution required is 10LOG(Pmax/Pmin)/6, which is equal to one bit for each 6dB of dynamic range. A high performance receiver with a dynamic range of 100dB or more would need at least 17 bits of ADC resolution.

ADCs are not perfectly linear devices. An ideal ADC would produce output values that are always linearly related to the level at the analogue input. All practical ADCs are prone to several types of error. Quantisation error, due to the limited resolution of the ADC, means that there is always at least ±0.5 LSB (least significant bit) of uncertainty for each sample. Non-linearity due to imperfections in the design and manufacture of the ADC lead to further distortion and noise. Clock jitter caused by random phase noise on the sampling clock will also add noise. ADC manufacturers usually specify error as a number of least significant bits. For example, a 16 bit ADC might only be accurate enough to produce 14 bits of accuracy and two bits of noise. This means that you will often need to choose a device with a greater resolution than a simple 6dB per bit calculation would suggest.

Some of the latest ADCs offer 16 bits of resolution at sampling rates of more than 100MHz. Claimed SNR and dynamic range figures for these devices suggest that they could provide a dynamic range of 80 - 90dB or perhaps even higher using oversampling techniques (sampling at well above the Nyquist limit, 2*f) or by parallel sampling (taking several samples at once and calculating an average sample value). It would seem that the state-of-the-art is coming closer to the point where readily available and reasonably priced ADCs will be able to match the performance of conventional receive mixers. However, there are a number of reasons why direct sampling at RF is a difficult technique to apply at frequencies of 30MHz or higher. It is not easy to make ADCs that combine high sampling rates and high resolution. You can have one or the other, but not both. Oversampling and digital post-processing can improve ADC performance, but to see a significant improvement, you will need to sample at several times the Nyquist limit. This is not hard at LF, but not so easy at HF and VHF.

There are good reasons why the image rejection I/Q mixer scheme is so popular for amateur SDR projects. High level mixers are easy to build, inexpensive and have predictable performance. RF phase shift networks (PSN) are reasonably easy to build provided that a separate PSN is used for each amateur band. If a broadband PSN is required for use in a general coverage rig, it is possible to use a pair of DDS ICs to produce accurate quadrature signals over a wide bandwidth. Some DDS devices have a pair of output DACs so that they can easily be used for this purpose. High quality ADCs designed for use in hi-fi audio equipment and PC sound cards are readily available. 16 bit devices are commonplace and 24 bit ADCs are used in high end equipment like the Delta-44 sound card. As these ICs are mass produced devices, they are available at extremely low cost.

Audio ADCs have typical maximum sampling rates of 48kHz to 192kHz. This is not much good for direct sampling at RF, but it is ideal for sampling the relatively low frequency I/Q output from an image rejection mixer. The simplest and least expensive option for a shack receiver is probably a simple I/Q front end as described last month and a standard PC to carry out the DSP functions. You don't need to have the latest and greatest PC hardware. The hardware requirements for many of the SDR programs are quite modest. A 1GHz CPU and a few hundred MB of RAM should be quite adequate for all but the most demanding applications. High-spec 24 bit sound cards are preferable to older 16 bit models, but you can expect reasonably good results, even from a fairly basic 16 bit card. Some older cards don't support full duplex operation. If you have such a card, you may need to use separate sound cards for input and output. Any recent hardware should be capable of duplex operation. All of the samples mentioned here **[2]** were recorded with the 24 bit on-board audio chip on my PC motherboard. If you have got older sound hardware without support for duplex operation, you can still experiment with SDR software by recording the I/Q signals first and then replaying the WAV file using the SDR software. Most SDR programs have support for file input.

LINRAD. Despite the name, Linrad **[3]** is a long established SDR program that runs on Microsoft Windows, Linux and other UNIX clones like FreeBSD. Linrad was originally developed for weak signal EME (moonbounce) CW reception. It is still a very powerful tool for finding very weak signals. Linrad has evolved into a general purpose digital radio design toolkit. Linrad is designed to be as hardware independent as possible. It works well as a DSP back-end for conventional receivers. I often use it to provide DSP filtering for my 2m SSB/CW receiver, which has a 2.4kHz IF bandwidth and no built-in digital filtering. Linrad makes an excellent DSP processor for simple I/Q receivers like the SoftRock or the simple receiver described in last month's Homebrew. It also works with fully digital receivers like the SDR-14 [1]. Linrad can be used as a receiver at HF, VHF and beyond. It is often used with simple 28MHz or 144MHz I/Q receivers as a back end for VHF, UHF and microwave receive converters for EME and weak signal working.

Linrad has support for several CW modes: weak signal CW, normal CW, meteor scatter CW and QRSS. It can also demodulate SSB, FM and AM. Linrad can also be used as a spectrum analyser with a RBW (resolution bandwidth) of a small fraction of 1Hz. This level of resolution is rarely achieved, even by the most expensive commercially made instruments. One interesting feature of Linrad is its ability to send either raw unprocessed or post processed digital data over a network. This means that a single aerial and receiver can be used by several operators. In the age of SDR, there is no requirement for the operator and the radio to be in the same place, or even in the same country! This might be useful for those lucky amateurs who have a holiday home in a quiet rural location. **Figure 1** shows a screen capture of Linrad in action. The baseband graph (bottom right) shows the received CW signal at 3542.5kHz and a stronger CW signal at about 900Hz above. As the stronger signal is well outside the filter passband, it causes no interference to the wanted signal.

I had intended to try another rather nice

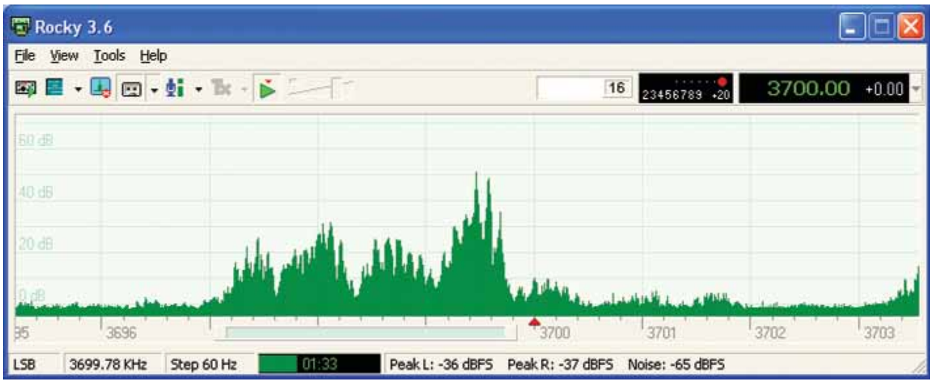


FIGURE 2: Using Rocky with my simple 80m converter, receiving an SSB signal on 3.70MHz.

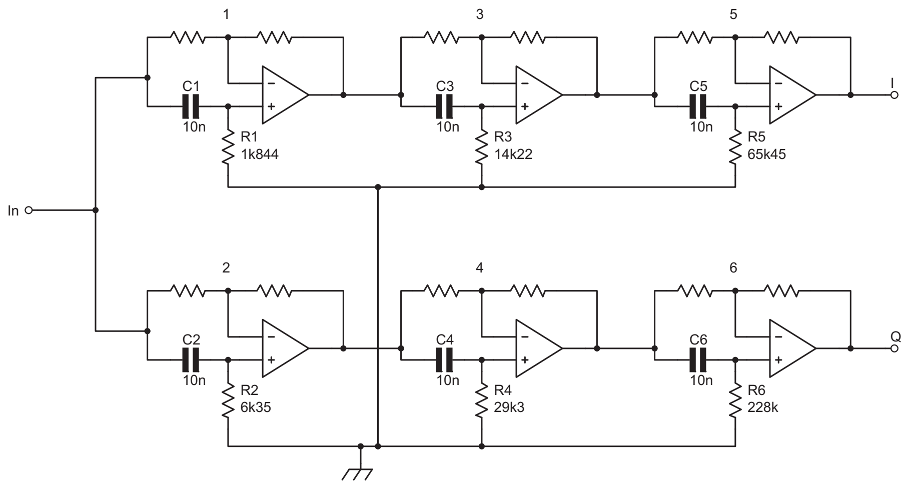


FIGURE 3: Active all-pass filter designed using QuadNet software.

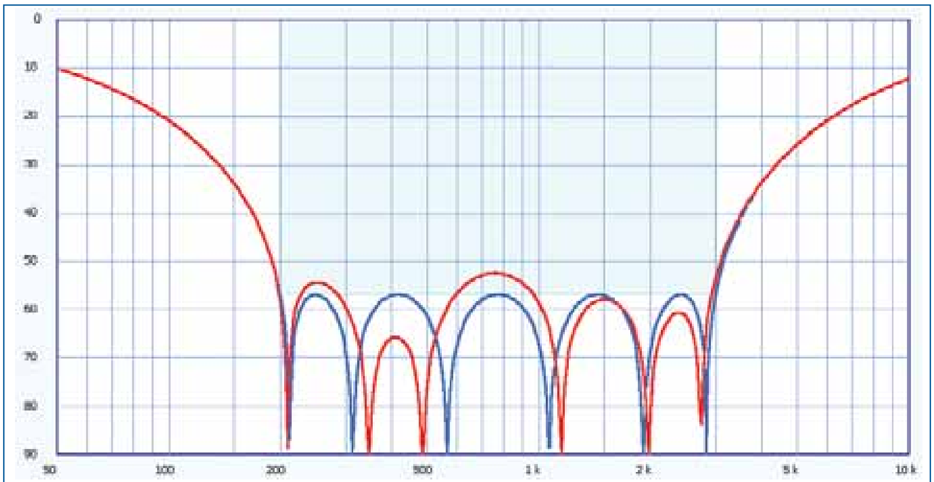


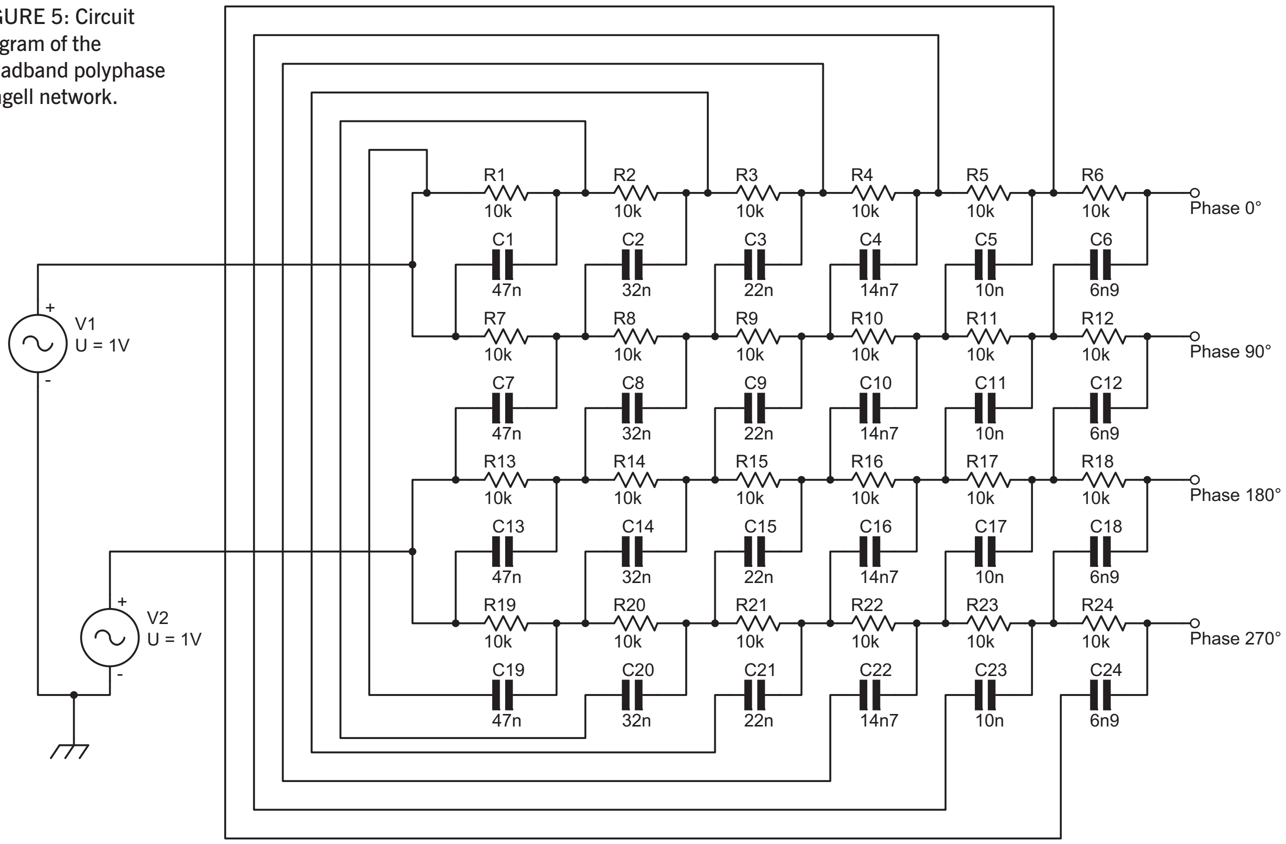
FIGURE 4: Expected opposite-sideband suppression using exact (blue) and actual (red) component values.

looking SDR console called SDR-Shell **[4]** with the simple I/Q receiver. So far, I have not managed to compile it and install it on my computer. If I ever do get it working, I will report my findings in a future issue…

WINDOWS. To test the capabilities of some of the SDR software for Windows, I borrowed a Windows XP machine and downloaded some of the more popular SDR packages for that platform. PowerSDR is the software used with the FlexRadio SDR-1000 and SDR-5000 transceivers **[5]**. Using this software with a simple I/Q receiver is a good illustration of the power of software defined radio. As so much of the functionality of these transceivers is defined in software, you can enjoy most of the powerful features of these fine radios, even when you are using a very simple I/Q front-end costing no more than £5 - £10.

I had some difficulties getting PowerSDR to play with the sound card in the Windows PC. The cheap sound card in this machine is not one of the 'supported cards' in the PowerSDR

FIGURE 5: Circuit diagram of the broadband polyphase Gingell network.



setup window. It did work, after a fashion, but the audio was a bit broken up and glitchy at times. Unusually for a Windows program, PowerSDR is free software, licensed under the GPL (Gnu General Public License). This means that the source code for the program is available to anybody who wants it. Users are allowed and encouraged to improve and modify the software. This is a very important point where SDR software for amateur radio is concerned. As more of the functions of amateur radio equipment are implemented in software, we must make sure that we have freedom to build, repair or modify our own equipment, regardless of whether it is software or hardware defined.

I also found time to do a brief test of Rocky [6], Winrad [7] and Linrad [3] for Windows. All three programs worked quite well. I was particularly impressed with the uncluttered appearance and ease of use of Rocky. It worked perfectly with the simple I/Q 80m receiver and produced excellent audio quality from SSB signals. **Figure 2** shows the Rocky program receiving a good quality SSB signal on 3.700MHz. Note that the distortion products on the opposite sideband just above 3.7MHz are 40 – 50dB down.

All of the above software was tested with the simple 80m I/Q receiver and the SoftRock-40. As I don't currently have any simple I/Q transmitting equipment, I was not able to try the transmit capabilities of any of these programs. I have got a bag of Schottky mixer diodes and a few ferrite cores, so perhaps I will build one soon.

If you would like to play with SDR without building any hardware, you should visit the Websdr project [8]. It is worth taking a look at the info page just to see the glorious dead-bug receiver circuits.

ANALOGUE PHASE SHIFT NETWORKS. A combination of a PC and simple RF hardware is probably the optimum configuration for a shack receiver or transmitter. For portable operation where a PC is not available or battery power is limited, an analogue phasing rig might be a better option. Last month, we looked at simple RC and LC phase quadrature networks. These simple circuits are ideal for generating RF quadrature signals at a single frequency or within a limited frequency range. Generating AF quadrature signals having an accurate 90° phase shift over the entire 300Hz to 3kHz range of human speech is a more difficult problem. The twisted wire quadrature hybrid can be used at audio frequencies, but you shouldn't expect to achieve more than about 20dB of opposite sideband suppression at speech frequencies, although this is still a useful amount of suppression. Putting 99% of the available power into one sideband and only 1% into the other will result in a considerable power saving compared to a suppressed carrier, double sideband transmitter. However, just 20dB of opposite sideband suppression would not really be adequate for anything other than very simple QRP equipment.

Phasing method SSB equipment was very popular with amateurs in the 1950s. Low cost phasing networks were an attractive alternative to expensive crystal and mechanical filters. Accurate measurement of resistor and capacitor values was a difficult problem at that time. These days, most of us take sub-1% accuracy for granted but, in the days before digital multimeters, accuracy of better than a few percent was not easily achieved. Several manufacturers sold pre-packaged audio phase shift networks. These devices contained a handful of close tolerance resistors and capacitors that could be used as a PSN. The best known example is the B&W model 350 2Q4 PSN. These devices still appear for sale on internet auction sites! Phasing equipment based on these networks can offer reasonable performance but, as with the twisted wire hybrid, there is a limit to the amount of opposite sideband suppression than can be achieved using such a simple network. Now that close tolerance components and accurate measuring instruments are readily available, it is quite a simple matter for the home constructor to build their own 2Q4 type networks.

The key to designing accurate wideband PSNs for audio frequencies is to use several cascaded stages of RC phase shift networks. Computer design and simulation software can be used to test more elaborate networks before the components are purchased and soldered into circuit. Wideband networks may have dozens of precision components that may have odd non-standard values which are difficult to obtain. When you build such a circuit, you would like to be reasonably sure

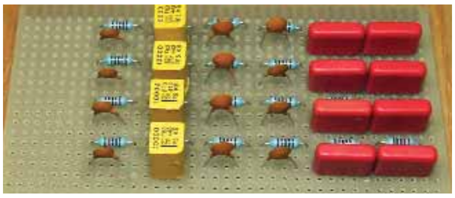


PHOTO 1: The prototype Gingell network of Figure 5 looks simple from above but has a lot of wiring and some extra components underneath.

that it will work as expected. Computer simulation will greatly improve your chances of getting it right first time.

Active filters based on opamps can make very predictable phase shift networks. One circuit that is often used in PSNs is the active all-pass network shown in **Figure 3**. This 6th order example was designed using QuadNet software from tonnesoftware.com **[9]**. This program can be used to design quadrature networks ranging from a simple 2nd order circuit to a 10th order monster with a theoretical opposite sideband suppression of 100dB. Such extreme accuracy is unlikely to be achieved in practice, but opposite sideband suppression of greater than 40dB is easily achieved and 50-60dB is possible with a carefully designed and adjusted circuit. **Figure 4** shows the expected opposite sideband suppression using the exact calculated values in blue and actual values made from series and parallel combinations of standard value resistors in red. For example, the 1.844k resistor is changed to 1.847k, made from 1.8k and 47R connected in series. This circuit configuration calls for a split rail power supply, which is slightly inconvenient for 12V battery operation.

A BROADBAND POLYPHASE NETWORK.

This month's construction project is a broadband audio PSN that can be used in a simple phasing SSB modulator or demodulator. This circuit will be used in a forthcoming simple 80m QRP SSB transmitter. The circuit is commonly known as a Polyphase network or a Gingell PSN after its inventor, M. Gingell, KN4BS. This circuit is loosely based on designs published in the current RSGB Radio Communications Handbook and several ARRL publications.

The use of six cascaded phase quadrature networks reduces the requirement for close tolerance components. This circuit can be built using standard 10% or 5% tolerance components, although 1% is of course better. The schematic is shown in **Figure 5** and a QUCS simulation plot of the 0°, 90°, 180° and 270° outputs is shown in **Figure 6**.

0/180° inputs on the left of the schematic are driven by a pair of anti-phase (180°) sine wave signals at a frequency of 500Hz. Running the simulation at frequencies of 1kHz, 1.5kHz, 2kHz and 3kHz show equally accurate phase quadrature.

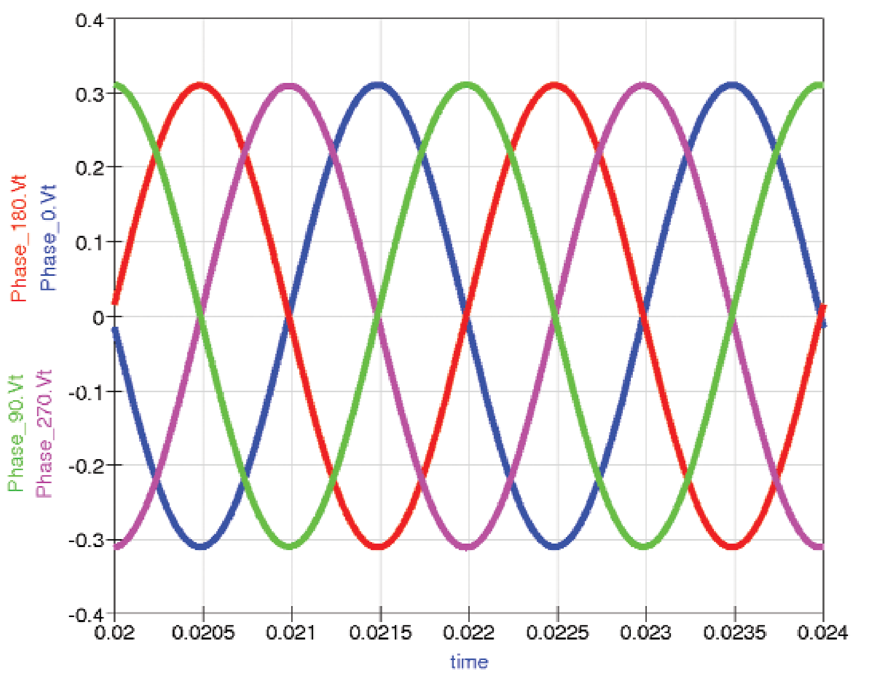


FIGURE 6: QUCS simulation output plot for the Gingell network in Figure 5.

CONSTRUCTION. The circuit was built on a strip of 0.1in matrix board. I used 1% tolerance 10k resistors just because they were readily available (Maplin 0.6W 1%). Some of the capacitors are close-tolerance types, the others are 10% types. If you can afford to spend a little bit of extra time on sorting the components, you should hand pick capacitors that don't depart too far from the recommended values. Try to ensure that the capacitors in each column are reasonably well matched to the other capacitors in the same column. The 32nF capacitor is 22nF+10nF, 14.7nF is 10nF+4.7nF and 6900pF is 4.7nF+2.2nF. **Photo 1** shows the finished circuit. It all looks very tidy from above, but the other side is not quite so orderly because of the extra capacitors that have been grafted onto the back of the board and, of course, the many wire links.

Next month: a simple SSB transmitter.

REFERENCES:

[1] www.rfspace.com/SDR-14.html
[2] http://homepage.eircom.net/~ei9gq/phasing.html
[3] www.sm5bsz.com
[4] http://ewpereira.info/sdr-shell/
[5] www.flex-radio.com
[6] www.dxatlas.com/Rocky/
[7] www.winrad.org/
[8] http://websdr.ewi.utwente.nl:8901/
[9] www.tonnesoftware.com/