Example gradient use in optimizing a 1024-element Frank sequence

Figures 1 and 2 show the autocorrelations of the 1024 element Frank sequence before and after optimization using gradient descent. The first plot shows a flat weighting function, and the second plot shows a 20 dB null region around the main peak. The approach is as defined in several publications listed at the end of this document. A code listing is provided showing the generation technique. The core of the technique is the efficient equation for gradient calculation. The surrounding code generates the original Frank sequence, iteratively applies the gradient, and plots the resulting autocorrelations. In [4] optimal (though approximate) scale factor calculations are provided. These are not used here; rather a simple approximation to the scale factor is applied. A reasonable local minimum nevertheless appears to be reached quickly. Additional iterations will reduce the sidelobes further – slightly for the unweighted case, but substantially for the weighted case (which may be several dB from the true local minimum). Future work will address strategies for maximizing convergence speed, with the goal of achieving the same results with fewer iterations.

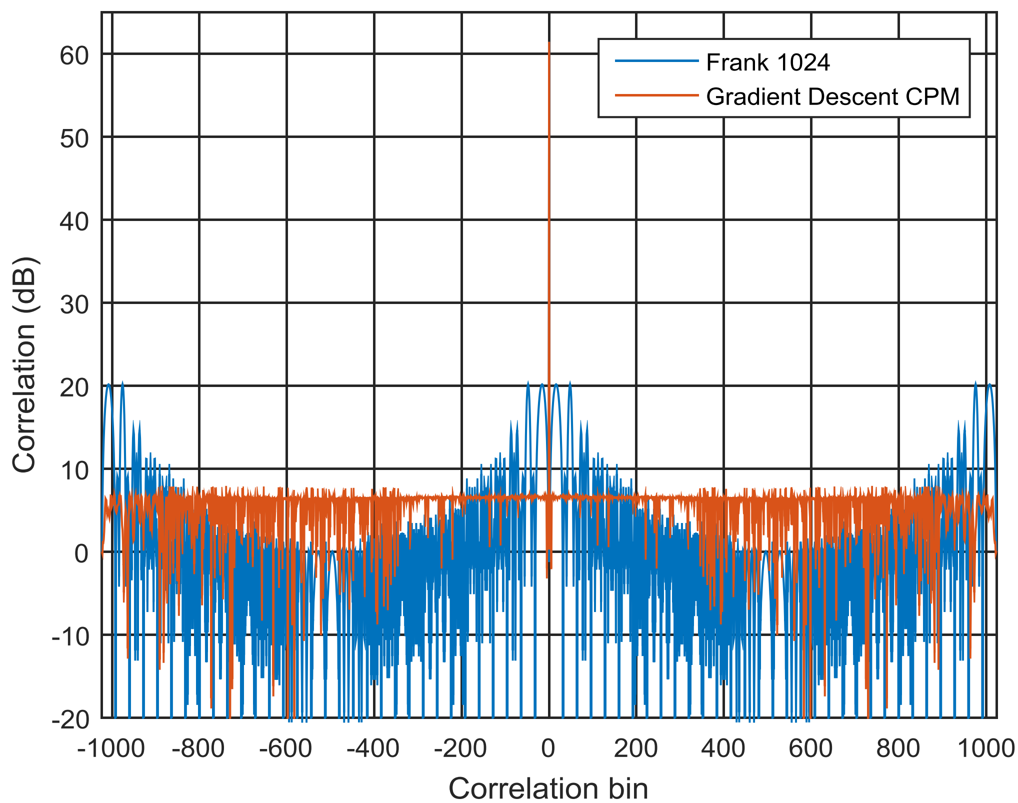

Figure 1: Frank sequence optimization

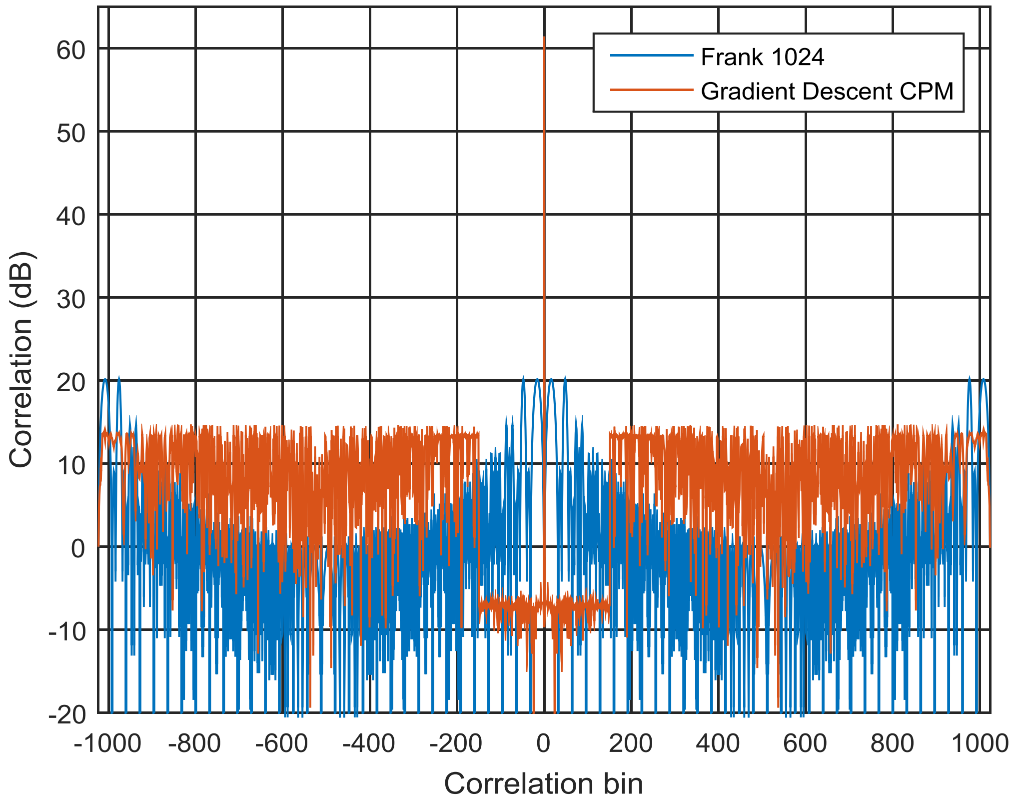

Figure 2: Frank sequence optimization with null region

```
% Example program to show application of gradient descent to optimization
% of the peak sidelobe of a 1024-element Frank sequence.  The approach is
% as described in a paper submitted to IEEE Transactions on Aerospace and
% Electronic Systems.
tic
% Generate 1024-element Frank sequence
k=0:31;
kk=k'*k;
c=exp(1i*2*pi*kk(:)/32)';
Frank1024_Start = c;

N=length(c);
zstart = conv(c,conj(c(end:-1:1)));

w=ones(1,2*N-1); % compression length is 2N-1
% Add suppression near the main peak.  Weight can be higher than 20 dB,
% but stability can be a problem when using the ad-hoc scale factor below.
%w(N-150:N+150)=10; % 20 dB null region
w(N)=0; % set peak weight to zero for numeric accuracy

for p=[1 35] % iterate for ISL then PSL
    for m=1:5000
        % Compute gradient: From equations and code sample in paper.
        fa=fft(c,2*N-1);
        x = fftshift(ifft(abs(fa).^2));
        g=w.^(2*p) .* x .* abs(x).^(2*p-2);
        gar = fftshift(ifft(fft(g,2*N-1).*fa));
        grad = -4*p*imag(c.*conj([gar(end) gar(1:N-1)]));

        % Normalize gradient to reduce numeric accuracy problems.
        grad = grad/max(abs(grad));

        % Rough estimate of scale factor for normalized gradient.  Ad-hoc
        % for this example; see [4] for a better equation.
        s=-0.05 / m^0.5;

        % Apply the gradient
        c = c .* exp(1i*s*grad);
    end
end
Frank1024_CPM=c;
zcpm = conv(c,conj(c(end:-1:1)));
psl = max(abs(zcpm(1:N-1)));
fprintf('PSL = %5.3f linear = %5.3f dB\n', psl, 20*log10(psl));

% Plot the compressions
figure;
xvals =(-N+1):(N-1);
plot(xvals, 20*log10(abs(zstart)));
hold on;
plot(xvals, 20*log10(abs(zcpm)));
xlim([-N N]);
ylim([-20 65]);
legend('Frank 1024', 'Gradient Descent CPM');
xlabel('Correlation bin');
ylabel('Correlation (dB)');
grid on
toc
```

Table 1: Sample code listing

References

1. "Efficient Optimization of the Merit Factor of Long Binary Sequences", J. M. Baden, IEEE Transactions on Information Theory, Vol. 57, No. 12, December 2011.
2. "Efficient Energy Gradient calculations for Binary and Polyphase Sequences," Baden, J.M., Davis, Mike, and Schmieder, Lance, IEEE 2015 International Radar Conference, May 11-15, 2015.
3. "Fast Gradient Descent For Multi-Objective Waveform Design", B. O'Donnell and J.M. Baden, IEEE 2016 International Radar Conference, May 2-6, 2016.
4. "Multi-Objective Sequence Design via Gradient Descent Methods" by J.M. Baden, B. O'Donnell, and Lance Schmieder, submitted to IEEE Trans. AES.