

Generation of a Coherent Dispersed Pulse by Simulation

K. B. Deshpande * and S. W. Ellingson

August 6, 2009

Contents

| | | |
|----------|----------------------------------|----------|
| 1 | Introduction | 2 |
| 2 | Methodology | 2 |
| 3 | Example | 5 |
| A | Appendix: Simulation Code | 9 |

*Bradley Department of Electrical & Computer Engineering, 432 Durham Hall, Virginia Polytechnic & State University, Blacksburg, VA 24061 USA. E-mail: kshitija@vt.edu

1 Introduction

In this report, we describe a method of generating a coherent dispersed pulse by simulation. A pulse from a source is dispersed as it travels through the interstellar medium (ISM). Additionally, it is broadened to durations on the order of milliseconds to seconds due to scattering by inhomogeneities in the ISM (See for example [1]). A pulse simulated to exhibit properties that represent these propagation effects can serve as a good test for radio transient search software. We simulate a dispersed, scattered-broadened Crab giant pulse (CGP) in the Eight-meter-wavelength Transient Array (ETA) data format, and use it to verify the “toolchain” developed to analyze the data from the ETA [2]. We describe the method in Section 2 (“Methodology”), and present an illustration of CGP simulation in Section 3 (“Example”).

2 Methodology

The dispersed pulse can be modeled as a “chirp” signal (where frequency is a function of time) in the time domain. It is easy to generate a dispersed pulse in the frequency domain, given knowledge of the dispersion delay at each frequency in the band of interest. The time domain signal can be then obtained by inverse Fourier transform. This procedure is described below in detail.

A pulse traveling through the interstellar medium (ISM) is dispersed; that is, the higher-frequency components of the pulse arrive earlier in time than the lower frequency ones. The dispersive delay is given by [3]:

$$\tau = \frac{\text{DM}}{\alpha_0 \nu^2} \quad [\text{s}] \quad (1)$$

where ν is instantaneous frequency in Hz, $\alpha_0 = 2.410 \times 10^{-16}$, and the dispersion measure (DM) has units of pc cm^{-3} . The shortest dispersion delay τ_1 will occur at the highest

frequency ν_1 in the band of interest. The dispersion delay for other frequencies ν relative to the delay at the highest frequency is given as $t_r(\nu) \triangleq \tau - \tau_1$. Thus,

$$\begin{aligned}
 t_r(\nu) &= \tau - \frac{\text{DM}}{\alpha_0 \nu_1^2} \\
 &= \frac{\text{DM}}{\alpha_0 \nu^2} - \frac{\text{DM}}{\alpha_0 \nu_1^2} \\
 &= \frac{\text{DM}}{\alpha_0} \left(\frac{1}{\nu^2} - \frac{1}{\nu_1^2} \right). \tag{2}
 \end{aligned}$$

Let us first consider the case of simulating a dispersed ‘‘impulse’’ as sketched in a ‘‘magnitude spectrogram’’ (intensity plot displaying the magnitude of spectrum as a function of both time and frequency) in Figure 1. Even though we start with the frequency domain signal, the final goal is to obtain the time domain signal. Let the magnitude of each value in Figure 1 be

$$\begin{aligned}
 a(t, \nu) &= 1 && \text{for } t_r(\nu) - \frac{dt}{2} < t \leq t_r(\nu) + \frac{dt}{2} \\
 &= 0 && \text{otherwise}
 \end{aligned} \tag{3}$$

where dt is equal to the sample period in the time-frequency domain. In other words, dt is the resultant time resolution after performing N -point fast Fourier transform (FFT) on the time domain data with a time resolution of δt . Thus, $dt = N\delta t$. In Figure 1, the vertical dashed line represents the frequency spectrum at time t' . The inverse Fourier transform of the spectrum at time t' gives the corresponding time domain signal for an interval of length dt .

Let us now consider the additional effect of scatter-broadening. To incorporate this effect, a Gaussian-shaped pulse can be implemented in the Fourier domain using the following equation.

$$a(t, \nu) = \exp \left(-\frac{(t - t_r(\nu))^2}{t_{1/2}^2 \beta} \right) \tag{4}$$

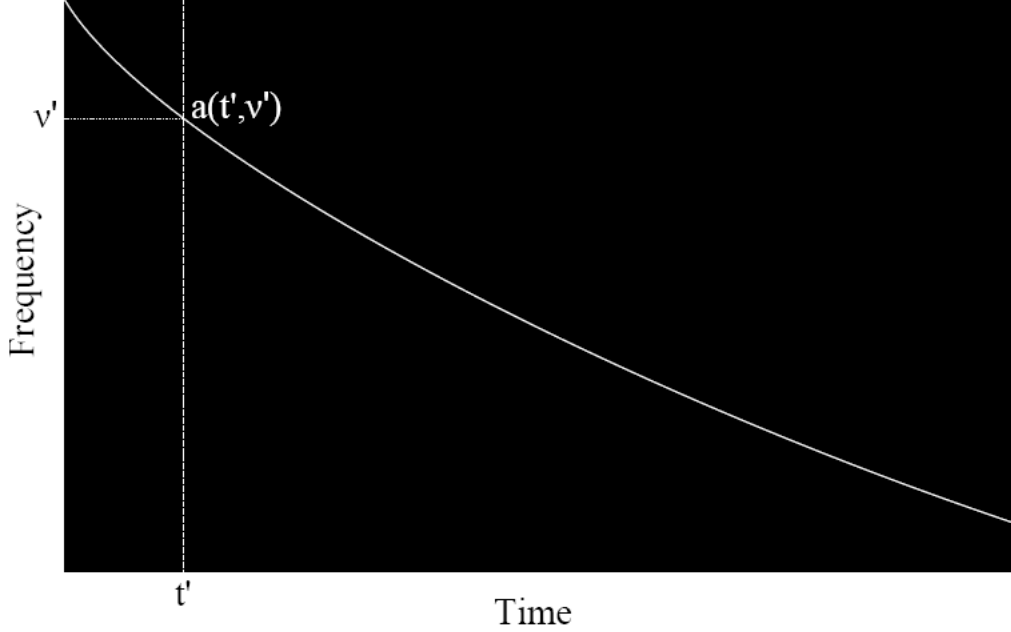


Figure 1: A sketch of a magnitude spectrogram showing a dispersed impulse in the time-frequency domain as a white solid line. Color code: black = 0, white = 1.

where $t_{1/2}$ is the full width at half maximum (FWHM) of this Gaussian-shaped pulse and β is a constant. At half maximum, the numerator in the exponent will be equal to $t_{1/2}^2$. Thus, from Equation 6 we have

$$\frac{1}{2} = \exp\left(-\frac{t_{1/2}^2}{t_{1/2}^2\beta}\right). \quad (5)$$

Solving this equation, we find $\beta = 1/\ln 2$. Substituting β in Equation 6, we obtain

$$a(t, \nu) = \exp\left(-\frac{(t - t_r(\nu))^2 \ln 2}{t_{1/2}^2}\right) \quad (6)$$

The procedure of pulse generation in this case is the same as that described in the previous case; except now with $a(t, \nu)$ from Equation 6, we get a dispersed, scattered-broadened pulse.

3 Example

To illustrate the dispersed pulse in the time-frequency domain, a raw spectrogram containing only a dispersed “impulse” is shown in Figure 2. Next, a dispersed CGP (DM = 56.791 pc cm⁻³) of FWHM = 1 s is generated in the ETA data format described in detail in [4], and is shown in the raw spectrogram of Figure 3. The signal-to-noise ratio (S/N) of this pulse after dedispersion is ~ 20 at a time resolution of 8.738 ms as seen in the dedispersed time series in Figure 4. Analysis of a simulated 7.8σ CGP similar to the pulse in Figure 3 embedded in a small dataset is done using the ETA toolchain, and is discussed in Section 4.1 of [2].

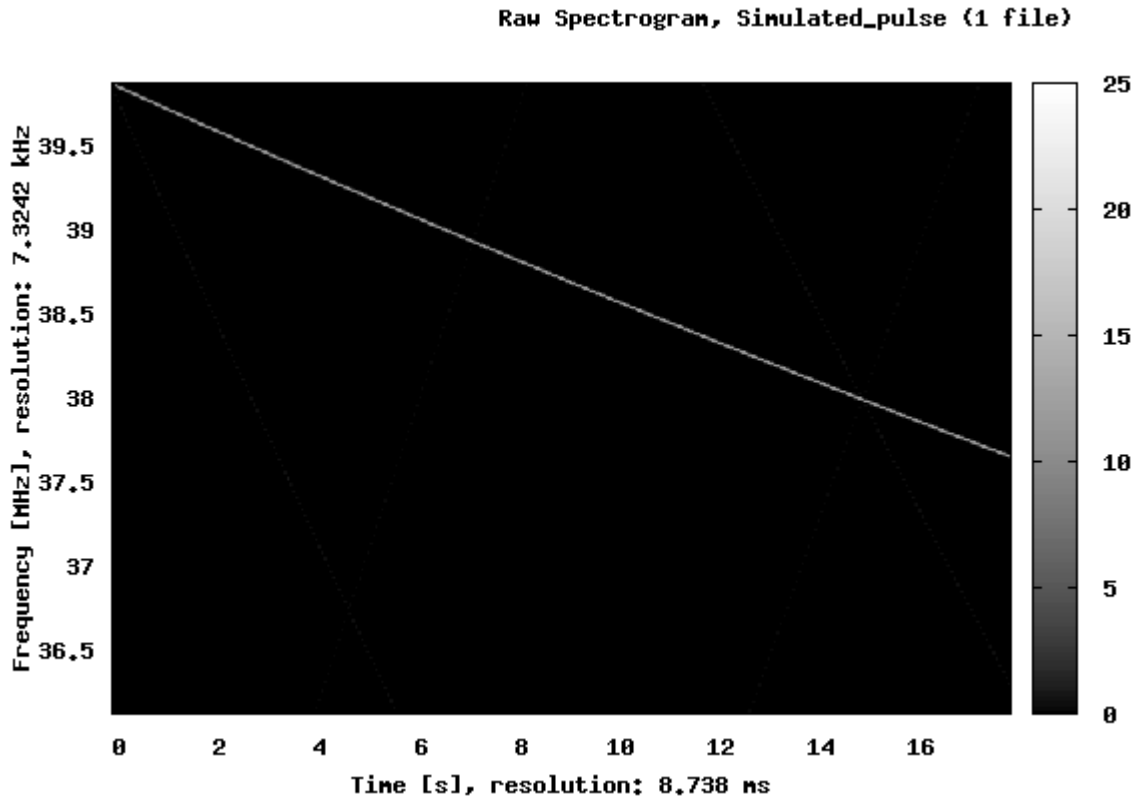


Figure 2: Spectrogram of a simulated impulse dispersed at 56.791 pc cm⁻³ without noise. Intensity of each point on the plot represents the power spectral density (PSD) in arbitrary units. The pulse spans from 39.875 MHz to 37.675 MHz in 17.9 s, which is the same result found theoretically using Equation 2.

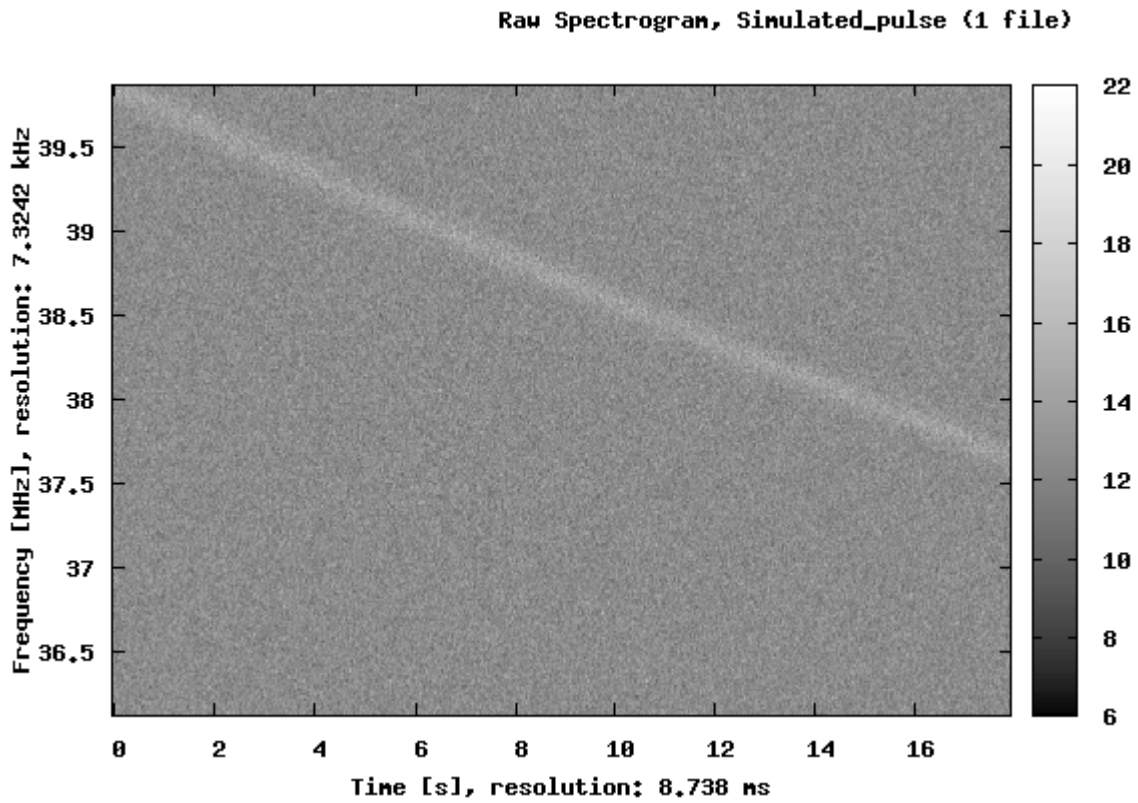


Figure 3: Same as Figure 2, except now FWHM = 1 s and noise is added.

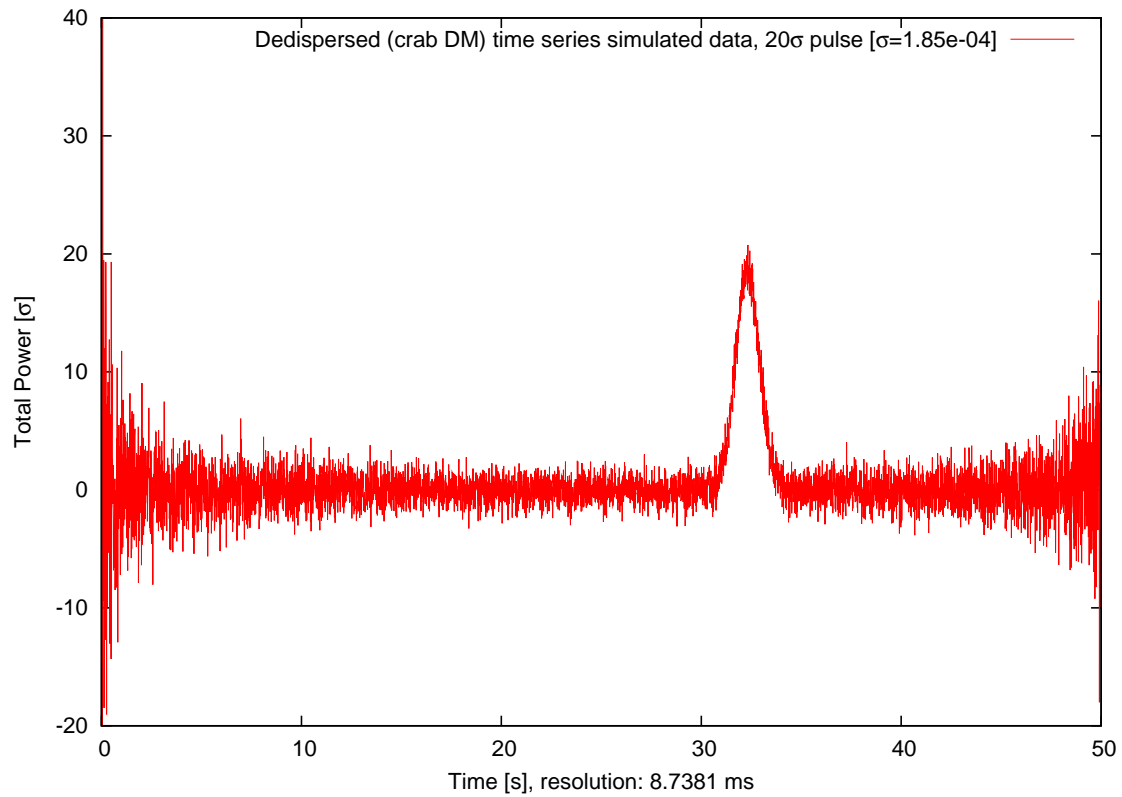


Figure 4: Dedispersed time series (at Crab DM) of the pulse from Figure 3. Fewer samples are averaged at the ends of the dedispersed time series, resulting in the larger variance at the edges.

References

- [1] J. M. Cordes *et al.*, “The Brightest Pulses in the Universe: Multifrequency Observations of the Crab Pulsar’s Giant Pulses,” *Astrophysical Journal*, vol. 612, pp. 375 – 388, September 2004.
- [2] K. B. Deshpande and S. W. Ellingson, “ETA Targeted DM Pulse Search Toolchain,” Internal Report, Virginia Polytechnic Institute and State University, July 2009, available online: <http://www.ece.vt.edu/swe/eta/>.
- [3] D. Lorimer and M. Kramer, *Handbook of Pulsar Astronomy*, Cambridge University Press, 2005.
- [4] C. Patterson *et al.*, “Searching for Transient Pulses with the ETA Radio Telescope,” *ACM Trans. Reconfigurable Technology & Systems*, vol. 1, no. 4, p. 19, January 2009.

A Appendix: Simulation Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MATLAB Code to generate a coherent dispersed pulse %%
% Developed by S.W.Ellingson, June 2009                %%
% Modified by K.B.Deshpande, July 2009                %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;
more off;
% user-selectable parameters
DM = 56.791; % [pc cm-3]
FC = 38.0e+6; % [Hz] center frequency of passband (as received)
FS = 7.5e+6; % [samples per second]
T = 17.8957; % [s] duration of simulation
LFFT = 1024; % size of FFT
p_amp = 0.65; % amplitude of the pulse in time domain
t_half = 1; % the FWHM of the pulse
% constants
alpha = 2.410e-16; % for DM in [pc cm-3], time in [s], and frequency in
% [Hz]
beta = 1/log(2);
% derived parameters
dt = 1/FS; % Time domain sample period
df = FS/LFFT; % FFT bin width
dt2 = LFFT*dt; % FFT period (how often we get a vector of LFFT output
% samples)
% more initializations
fbin = [ 0 : df : FS/2 - df , -FS/2 : df : -df ]; % [Hz] FFT bin center
% frequencies (at baseband)
fbin = fftshift(fbin); % fftshift into low-high
% order
f1 = FC + max(fbin)/2; % [Hz] start frequency
% (center freq of pulse at start of simulation)
t1 = DM/(alpha*(f12)); % [s] reference delay for
% start freq (f1)
tbin = (DM./(alpha*((FC+fbin-df/2).^2))) - t1; % [s] delay relative to
% t1 for lowest frequency in each bin
tbin = [ tbin (DM/(alpha*((FC+fbin(LFFT)+df/2).^2))) - t1 ]; % adding one
% more value to simplify algorithm below
num_scl = 1000;
X = zeros(1,LFFT);
```

```

x = zeros(1,LFFT);
xs = zeros(1,num_scl*LFFT);
t = 0; % [s] initialize sim time
kmax = ceil(T/dt2); % number of FFT input blocks to process
fp = fopen('pulse.out','w');
k = 0;
k1 = 0; %count the number of FFT blocks to write
while k<kmax, % loop over FFT input blocks
    k=k+1;
    k1= k1+1;
    b = 1; % count bins
    for f = fbin, % loop over FFT bins
        t2a = tbin(b+1); % delay for max frequency in this bin
        t2b = tbin(b); % delay for min frequency in this bin
%% Gaussian shaped pulse
        X(b) = exp(-(t-t2a)^2/(beta*t_half^2));
%% Impulse
        % X(b) = 0;
        % if ((t>t2a) & (t<=t2b)), % are we in this time-frequency cell?
        % X(b) = 1;
        % end
        b = b + 1;
    end % for f
    t = t + dt2; % update time
    X = fftshift(X); % FFT shift into screwy FFT order
    x = ifft(X)*LFFT; % Inverse FFT gives time domain
    xs(:,(k1-1)*LFFT+1:k1*LFFT)=x;
    if (k1 > num_scl)
        disp([t k]);
        k1 = 1;
        max_xs = max(abs(xs));
        for l=1:max(length(xs)),
% Write in ETA format
            xr = round(10*(p_amp*real(xs(l))/max_xs+randn()));
            xi = round(10*(p_amp*imag(xs(l))/max_xs+randn()));
            fwrite(fp,bitshift(xi,1),'schar',0,'ieee-be');
            fwrite(fp,bitshift(xr,1),'schar',0,'ieee-be');
        end % for l
    end %for k1
end % while k
fclose(fp);
return;

```