1. (a) This is the classic reciprocal-conjugate-symmetric all-zero diagram of a linear-phase FIR filter. It has zeros on the unit circle (frequency axis) for higher frequencies, but away from the unit circle for lower frequencies, so it is low pass. It has eight zeros, so it is an 8th order filter.

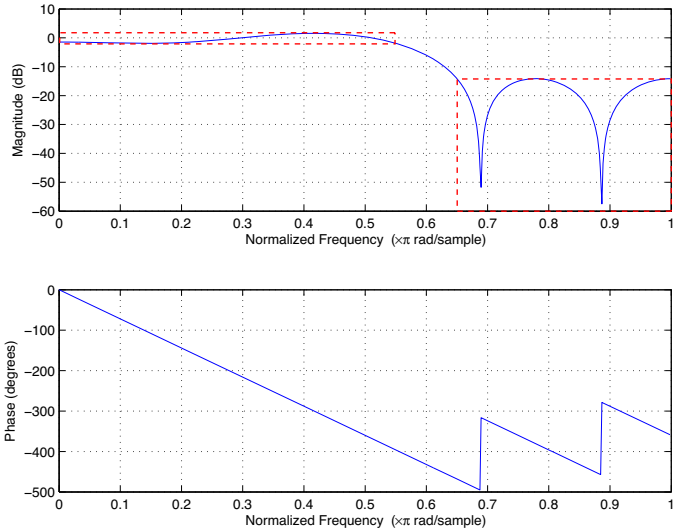
   (b) There are several ways to construct linear-phase FIR filters, including windowing the IDTFT of an ideal response, but in order to ensure a minimax-optimal filter (i.e., a filter that minimizes the maximum deviation from a piecewise-constant ideal specification), the way we know is via the Parks-McClellan approach. In fact. this filter was designed in Matlab as:

   ```
   >> h = firpm(8, [0 0.55 0.65 1], [1 1 0 0]);
   ```

   That is, it's an equiripple low-pass filter with equal-sized ripples in both pass and stop band, with a passband edge at 0.55π rad/samp, and a stopband edge at 0.65π rad/sec.

   (c) This is the simplest path by which we could convert this filter to a DT highpass:

   - Pre-warp the DT band edge $\omega_c = 0.5$ rad/samp to the CT domain via the inverse of our standard bilinear transform warp, so $\Omega+_c = \tan(\omega_c/2) \approx 0.2553$.
   - The actual magnitude and phase response are shown below:

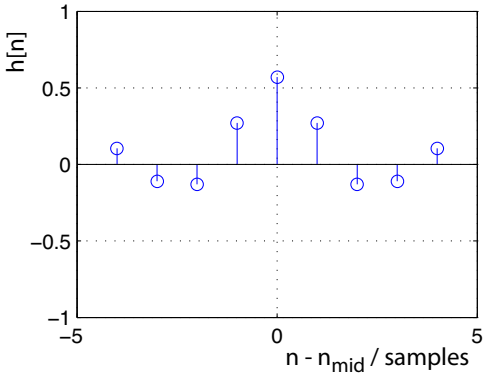


   From the zero locations in the pole-zero diagram, we know the magnitude response will go to zero at around 0.7π and 0.9π, and will have a minimum in the passband around 0.2π at the point of closest approach to the reciprocal pairs of non-unit-magnitude zeros. We can infer that it will have local maxima at $\omega = 0$ and $\omega = \pi$, although according to the alternation theorem only one of those values will have to be at the limit of the error. Here we see that the local maximum at $\omega = 0$ does not, in fact, touch the error limit.

   In this case, the maximum deviation in both passband and stopband is equal, although that's hard to tell from the pole-zero diagram. In this case, the largest gain in the passband is about $1.2\times$ (0.2 away from the target gain of 1), or about 1.6 dB, and the largest gain in the stopband is 0.2 (i.e., also 0.2 away from the target gain of 0), or about -14 dB.

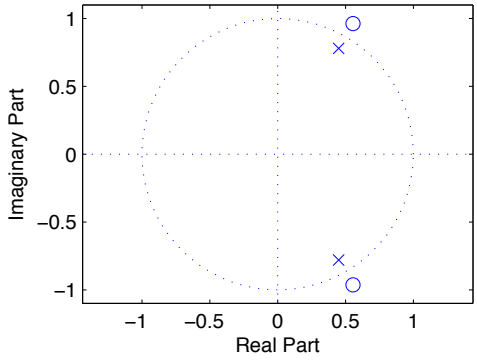
The phase response must be linear, but it would be valid to draw it as identically zero since we've been vague about the precise temporal alignment of this filter. The most usual interpretation, however, would be to treat this as a causal filter and thus the delay would be 4 samples, and the phase response will have a slope of -4. It wraps, of course, when it reaches $-\pi$, and it shows π jumps when the magnitude response goes through zero.

- The actual impulse response is shown below:

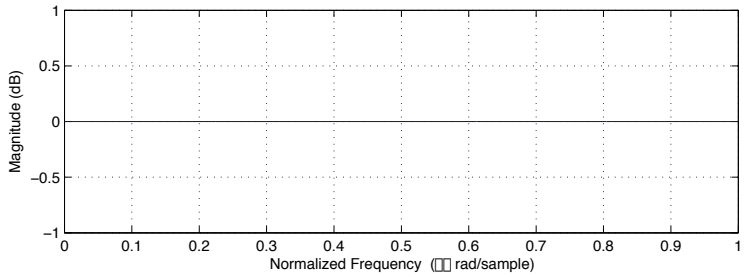


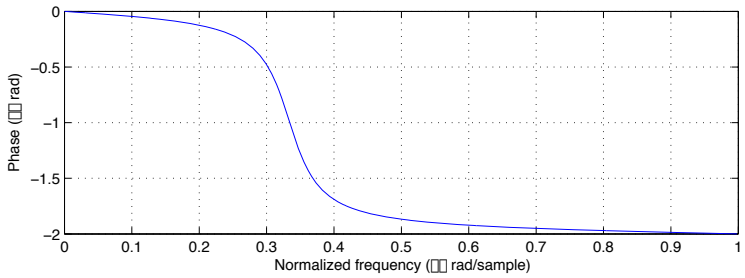
It has to be nine points long, and symmetric in time. It's going to have a vaguely sinc-like shape since it is a low-pass filter. The mainlobe width is inversely proportional to the cutoff frequency, so we could guess it was going to be fairly narrow for a filter that passes frequencies beyond $\omega = \pi/2$.

2. (a) An allpass filter has reciprocal zeros "balancing" the magnitude impact of each pole, but not the phase effects. Thus, this second order system will have zeros at the reciprocals of the pole locations:



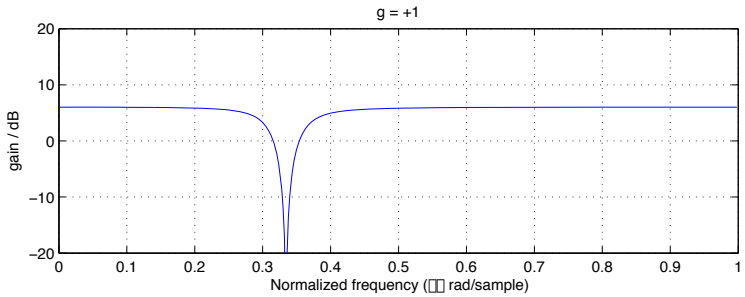
Since this is an allpass filter, its magnitude response is identically 1 (0 dB) at all frequencies. But its phase response is more complicated. By thinking about the angle subtended from the roots and how it changes as ω goes from 0 to 2π, we can understand that the angle from the zeros will change one way, then the other, but will end up the same after the complete 2π revolution. The angle from the poles, however, increases monotonically (leading to a monotonic decrease in the phase response), and is 2π larger, per pole (i.e. 4π total), by the time $\omega = 2\pi$. By symmetry, the phase response must achieve half of this overall change after half a revolution ($\omega = \pi$), thus we see a phase response going from zero at $\omega = 0$ to -2π at $\omega = \pi$:

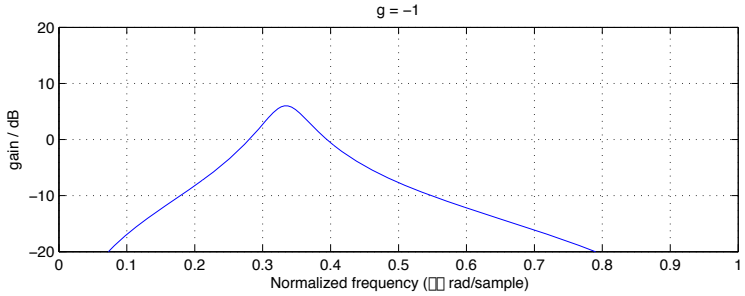




We know the phase change will be fastest for values of ω where $z = e^{j\omega}$ comes closest to the roots, so that gives us the overall shape. (I wasn't worried about the peak slope, although I guess it could be estimated).

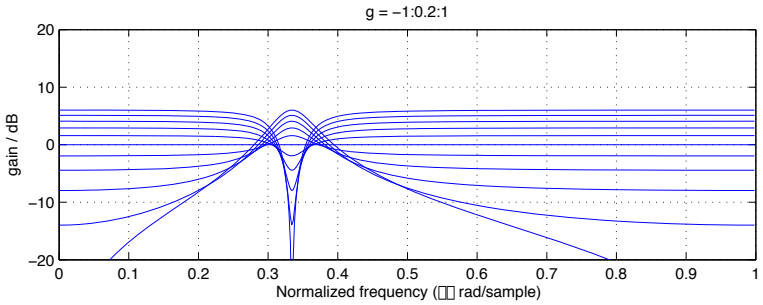
(b) The point about this structure is to notice that although the magnitude response of the direct and the allpass branches will be the same, their phases will not. In particular, since the allpass filter has a phase response of zero at $\omega = 0$ and 2π at $\omega = \pi$, it will be in phase with the direct path for those two extreme frequencies. But at frequencies close to the pole/zero pairs, the phase shift approaches π, which means it will be out-of-phase. $g = 1$ corresponds to adding the two branches; the in-phase components will reinforce (giving a 6 dB gain boost), but the out-of phase components (which, remember, both have a magnitude of 1) will cancel, creating a zero in the overall response, or a notch in the magnitude response. $g = -1$ is the opposite case, with full cancellation at the two extreme frequencies, and a 6 dB boost at the "critical" frequency defined by the roots:





(c) Clearly, for $g = 0$, the system is just a pass-through, so the gain (and phase) are flat. So from $g = 0$ to $+1$, we see a gradual deepening of the bandstop notch, and for $g = 0$ to -1, we see a gradual growth of the bandpass peak. Here, for your interest, is a superposition of a range of values for g from -1 to 1 in steps of 0.2:

```
r = 0.9;
the = pi/3;
a = [1, -2*r*cos(the), r*r];
b = fliplr(a);
for g = -1:0.2:1; ...
   [H,W] = freqz(a+g*b,a); ...
   plot(W/pi, 20*log10(abs(H))); ...
   hold on; ...
end
grid
axis([0 1 -20 20])
```



Thus, we have a filter which is continuously variable between a band-stop (notch) and a band-pass according to the value of g, all thanks to the variation in phase interactions (cancellation and reinforcement) at different frequencies.

(d) "Varying the characteristics" of the 2nd order allpass system means changing the angle and radius of the poles (which then fixes the zeros at the reciprocal locations). This, in turn, modifies the allpass phase response to have its point of maximum slope (knee) occur at different frequencies (as the pole angle varies), and have the rapidity of phase change across the knee get greater or smaller (as the radius gets closer to or further away from one). The knee frequency corresponds to the center frequency of the bandpass/bandstop response from the previous sections, and the knee "sharpness" determines the bandwidth of those characteristics, with a sharper knee (pole closer to the unit circle) corresponding to a narrower bandpass or bandstop response.

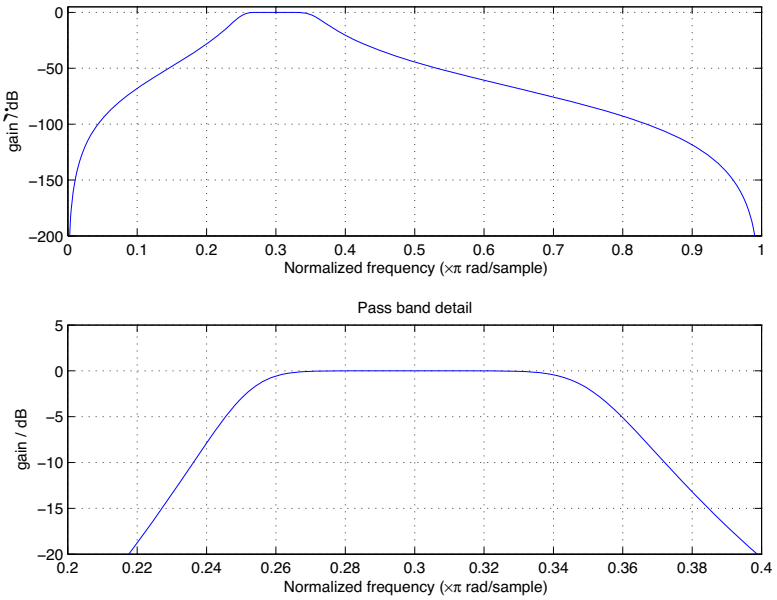
Thus, with three parameters: pole angle (0 to π), pole radius (0 to 1), and mixing gain (-1 to 1), we have a parametric filter capable of producing either bandpass or bandstop response for any frequency and of variable bandwidth. This is the basis of the "parametric eq" (equalizer) block that is the stable of a recording engineer's mixing desk. By providing a couple of these per input channel, the sounds of different instruments can be tweaked over a wide range of characteristics to make them mix together more pleasantly.

3. (a) According to the bilinear transform relations, to achieve a DT-domain band edge at $\omega = 0.25\pi$, we need to have a CT band edge at $\Omega_L = \tan(\omega_L/2) = \tan(\pi/8) \approx 0.41$. So, we have to "scale" all the prototype frequencies by 0.41, making the high-side band edge appear at $\Omega_U = 1.5\tan(\pi/8) \approx 0.62$ rad/sec. Then, when we put this through the bilinear transform, it will appear at $\omega_U = 2\tan^{-1}(\Omega_U) \approx 1.11 \approx 0.35\pi$ rad/samp.

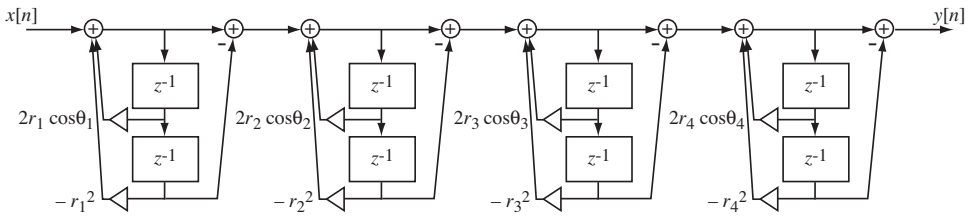
(b) The DT magnitude response will be a bandpass filter with a passband that is 3dB down at $\omega_L = 0.25\pi$ and $\omega_U \approx 0.35\pi$ rad/samp. Since it based on a Butterworth, it will be monotonic with no ripples. We can see from the s-plane pole-zero diagram that it has zeros at zero frequency; thanks to the mapping of the bilinear transform, which maps the remaining four "virtual" zeros

at infinite frequency in the CT domain to $\omega = \pi$ in the DT domain, it will also have zeros at the Nyquist frequency. Thus the plot will have minima at these two extremes and one maximum at a frequency roughly in the middle of the passband. Here's the actual plot from Matlab:

```
[B,A] = butter(4, [1 1.5], 's');
[b,a] = bilinear(B,A,0.5/tan(pi/8));
[H,W] = freqz(b,a);
subplot(211)
plot(W/pi, 20*log10(abs(H)))
axis([0 1 -200 5])
grid
xlabel('Freq (\times\pi rad/samp)')
ylabel('gain / dB');
subplot(212)
plot(W/pi, 20*log10(abs(H)))
axis([0.2 0.4 -20 5])
grid
xlabel('Freq (\times\pi rad/samp)')
ylabel('gain / dB');
title('Pass band detail');
```

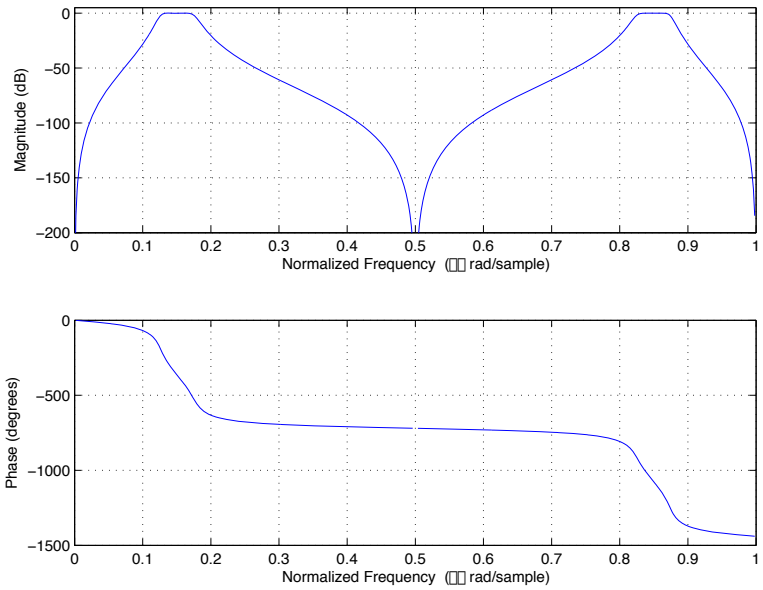


(c) This is an 8-pole, 8-zero system, so it will need to be composed of 4 second-order sections, each implementing one conjugate pair of poles and one conjugate pair of zeros. All the zeros are either at $\omega = 0$ or $\omega = \pi$. Although we haven't seen the the DT pole-zero diagram, it's easy to imagine, with a set of four poles in a little semicircle just inside the unit circle during the passband. According to the discussion of second-order section implementations in lecture 7, we want to group the poles with the closest zeros, so the two poles with the larger angle should be matched with zeros at $\omega = pi$, and the other two with $\omega = 0$, although in this case it isn't going to make much difference. Note that two zeros at $\omega = 0$ correspond to $(1-z^{-1})^2 = 1-2z^{-1}+z^{-2}$, and for $\omega = \pi$ we get $(1 + z^{-1})^2 = 1 + 2z^{-1} + z^{-2}$, so the feed-forward (zero) portions have simple parameters. Also note that if put one $\omega = 0$ zero and one $\omega = \pi$ zero together in each second-order section, we'd get $(1 + z^{-1})(1 - z^{-1}) = 1 - z^{-2}$, which would actually save us one operation in each second-order section (since the z^{-1} coefficient is zero). However, I didn't require you to note this for full marks. If the poles occur at locations $\lambda_i = r_i e^{\pm j\theta_i}$, the feedback coefficients will come from $(1 - \lambda_i z^{-1})(1 - \lambda_i^* z^{-1}) = 1 - 2\Re\{\lambda_i\}z^{-1} + |\lambda_i|^2 z^{-2} = 1 - 2r_i \cos(\theta_i)z^{-1} + r^2 z^{-2}$. When you expand this out into an actual flow diagram, the signs of the feedback coefficients flip, so the the feedback gains are actually $2r_i \cos\theta_i$ from the first delay, and $-r_i^2$ from the second delay. Hence, I was looking for a block diagram something like the one below:



(d) This is making a comb filter out of the structure; replacing every z in the system function by z^2 keeps all the same magnitude and phase values for $H(z)$ on the unit circle, but now they get repeated twice for each cycle around the unit circle, leading to a magnitude response with two bandpass regions. Note that since we are compressing the symmetric, periodic structure of the DTFT by a factor of two, the bandpass regions are actually symmetric around $\omega = 0$ and $\omega = \pi$, i.e., while the first passband is squeezed down to lie between $\pi/4/2 = \pi/8$ and $0.35\pi/2 \approx 0.18\pi$, the second appears at $\omega = \pi - 0.18\pi$ to $\pi - \pi/8$. Here's the Matlab (note that you were not asked to sketch the phase response, but freqz plotted it anyway):

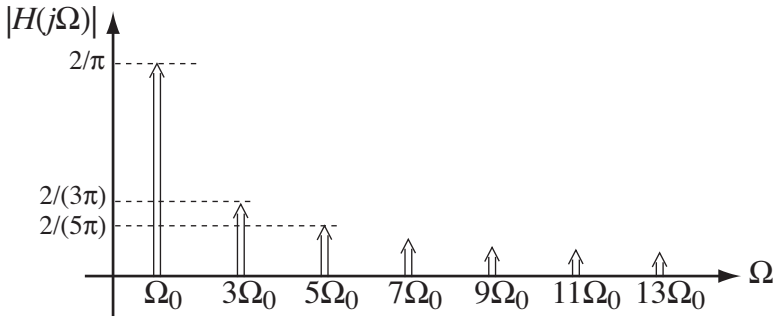
```
b = [b;zeros(1, length(b))];
b = b(:)';
a = [a;zeros(1, length(a))];
a = a(:)';
freqz(b,a)
axis([0 1 -200 5])
```



4. (a) As we saw in the introduction to the Fourier transform, a square wave consists of odd-numbered harmonics, with magnitudes decaying as $1/n$. Thus, its Fourier transform actually consists of a set of impulses at the frequencies corresponding to odd multiples of the fundamental Ω_0, with weights equal to the c_ks of the Fourier series, where

$$\begin{aligned}
c_k =& \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)e^{-j\frac{2\pi k}{\tau_0}t}dt \\
=& \frac{1}{\tau_0}\left(\int_{-\tau_0/2}^{0} -e^{-j\frac{2\pi k}{\tau_0}t}dt + \int_{0}^{\tau_0/2} e^{-j\frac{2\pi k}{\tau_0}t}dt\right) \\
=& \frac{1}{\tau_0}\left(\left[\frac{\tau_0}{j2\pi k}e^{-j\frac{2\pi k}{\tau_0}t}\right]_{-\tau_0/2}^{0} - \left[\frac{\tau_0}{j2\pi k}e^{-j\frac{2\pi k}{\tau_0}t}\right]_{0}^{\tau_0/2}\right) \\
=& \frac{1}{j2\pi k}\left(1 - e^{-j\pi k} - e^{-j\pi k} + 1\right) \\
=& \frac{1}{j\pi k}\left(1 - e^{-j\pi k}\right) = \frac{1}{j\pi k}\left(1 - (-1)^k\right)
\end{aligned}$$

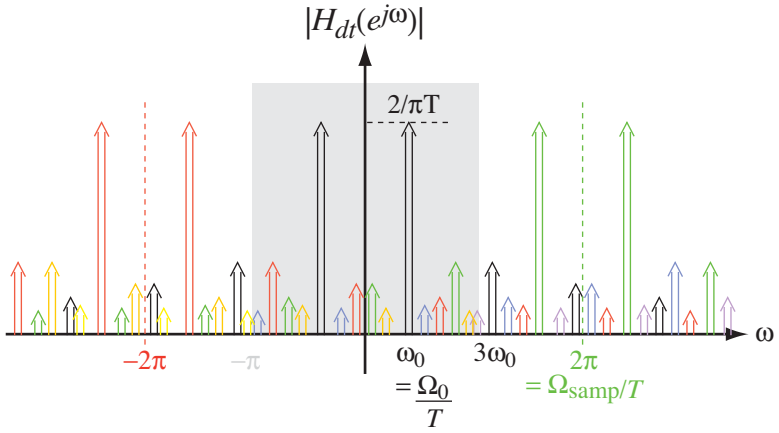
Thus, c_k is zero for all even k (when $(-1)^k$ evaluates to 1), and $|c_k| = \frac{2}{k\pi}$ for odd k, so the weights of the first few impulses in the spectrum should be around $2/\pi \approx 0.64$, $2/(3\pi) \approx 0.21$, $2/(5\pi) \approx 0.13$ etc.



(b) From lecture 11, when we convert a continuous spectrum into a discrete-time spectrum by sampling in the time domain, the resulting discrete-time spectrum is simply the superposition of a set of scaled versions of the continuous-time spectrum, thus:

$$H_{dt}(e^{j\omega}) = \frac{1}{T}\sum_{\ell} H(j\left(\frac{\omega}{T} - \ell\Omega_{\text{samp}}\right))$$

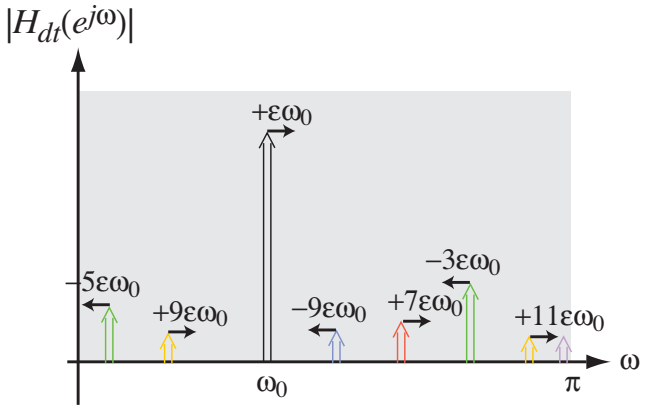
For clarity in the figure, let's assume T is a little smaller than $\tau_0/5$, so the sampling frequency falls just above the impulse at $5\Omega_0$. In the figure below, I essentially duplicated the line spectrum from above and centered it over each multiple of $\omega = 2\pi$, i.e. the sampling frequency. I gave each set of impulses a different color so you can get some idea where they come from. The first set on the positive side are green, and the first set on the negative side (centered around $\omega = -2\pi$) are in red.



The point is that, since the original square wave has lots of significant energy above the Nyquist rate ($f_{\text{samp}}/2$, it all aliases down to make a complex pattern of harmonics in the "principal range" of DT frequencies, $\omega = -\pi \ldots \pi$ (shown as the shaded box in the figure).

(c) The figure below focuses on the range $\omega = 0 \ldots \pi$, and shows how each of the harmonics in that range from the sketch above will move as the fundamental changes. The point is that, in contrast to a regular harmonic signal, where all the harmonics move in the same direction in proportion to their frequency, here most of the harmonics are aliases from one place or another, and they all

move in different directions with no clear dependence on their frequency. The result is a sound that changes in character in a distinctly unnatural and "impure" way.



(d) What I'm getting at here is that by sampling, we end up with a signal that is nothing like a band-limited square wave. We need to implement an anti-aliasing filter prior to the sampling to avoid these harmonics. The case illustrated was particularly severe because there was only one harmonic (the fundamental) within the Nyquist limit. More commonly, you would be working with a fundamental far below the Nyquist frequency, and would get many "legitimate" harmonics in your discrete time spectrum before you started getting aliases. However, the aliases will still be there, and since the harmonics die away so slowly with frequency for a signal like the squarewave that has discontinuities, they will be very significant. If you come to my music signal processing class next semester, you'll be able to hear the sound of this kind of un-anti-aliased discrete-time square-wave oscillator, and it's quite alarmingly bad.

To improve this, we could generate our continuous-time square wave, then put it through a low-pass filter to remove any energy above the Nyquist, and then sample it. But the whole point of this approach is that, to generate a discrete-time square wave on a computer, you never actually have to calculate the continuous-time signal (and how would you, anyway?). Instead, you implement $x[n] = 1$ or -1, depending on n, as a piece of conditional logic, and you might think you're done. To add anti-aliasing on top of this kind of algorithm, one approach is to figure the impulse response of your anti-alias (lowpass) filter, the sum it into a signal once every half-period, with alternating sounds, to generate an anti-aliased representation of an impulse train with alternating polarities. You can then integrate this signal to get the band-limited square wave.

However, a simpler approach is just to control a bank of phase-locked sinewave oscillators at odd multiples of the fundamental and with amplitudes set in inverse proportion to their harmonic number, and do a direct Fourier synthesis of a bandlimited square wave. Then, as Ω_0 varies, you can suppress and reintroduce harmonics as they pass above and below the Nyquist frequency, ensuring there is no aliasing.