

How the BumbleBee Radar Works

The Samraksh Company, September 2015

The [BumbleBee](#), named after the [world's smallest bat](#), is a micro-power pulsed Doppler radar with complex output that provides high-precision, high-resolution information about a target's relative motion and direction. Software can adjust the detection focus from a near-field range of 0.2 to 1 m, out to a far-field range of 1 to 10 m. The BumbleBee only consumes about 12 mA, so when using typical 1.5v AA alkaline batteries with a [capacity of 2400 mA hours](#), it can run at 100% duty cycle for about 8 days. It is easily interfaced to mote-scale devices such as the [eMote .NOW](#) or the [Arduino](#).

1 The Limitations of Low Power

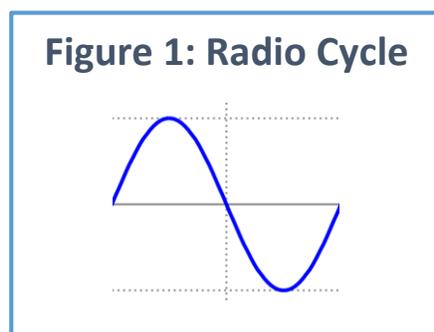
[Pulse-Doppler radar](#) can measure the range of a target by measuring the elapsed time between a pulse of radio energy (the reference) and its reflection (the reflection), and can measure target velocity using the [Doppler effect](#) by the frequency shift of the reflection.

The problem for a low-power application like ours is that the speed of light is so large that measuring the pulse and reflection times and frequencies accurately requires a lot more computing power than a battery can sustain for very long. Light takes about [1 nanosecond to travel 1 foot](#). To measure the round-trip time to a foot of resolution would require a clock running with a speed of at least 0.5 GHz (2 nanosecond cycles). That's 0.5 GHz rather than 1 GHz because a change of one foot in distance to the target is two feet for a round trip. So we only need a resolution of 2 nanoseconds.

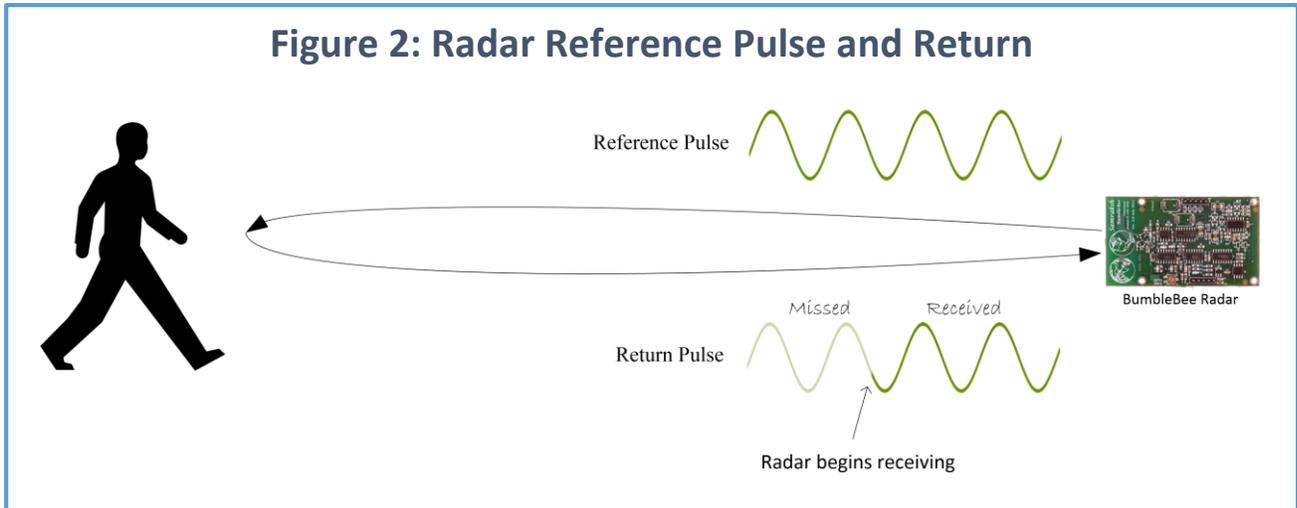
Nevertheless, this puts us into the PC range, with corresponding power requirements. By contrast, the Arduino UNO R3 has a 16 MHz clock, about 3% that speed. This would give us about 30 feet of resolution. We wouldn't know much more than that the target is in range of the radar, and even at Arduino speeds we wouldn't be able to sustain battery power for very long. We need to do better and at less power.

2 Relative Distance

What we'll do is give up on absolute distance and see if we can measure a different kind of distance: relative distance; as we'll see later, we can also get velocity. Our point of reference is the radar *pulse* that's sent out: a radio wave that's transmitted for some period of time. The pulse is a *sine wave* in which the amplitudes form a regular, repeating pattern. Each repeat is called a *cycle*. Here is an example of one cycle in a sine wave, beginning and ending when the amplitude is zero.



The radar broadcasts a sine wave for a short period, producing a *reference pulse*, pauses a bit and then starts receiving the *return pulse* that has been reflected off some target. While paused, some of the return may have come in, so when the radar does start receiving, it will in general be somewhere in the middle of the reflected reference.



In the figure above, the radar has transmitted the reference pulse, paused, and then started receiving the reflected return. Notice that part of the reference had already arrived before the receiver turned on; this is the lighter part on the left. Hence the return that's actually received is the darker part which begins partway into the reflected return. In this case, the return begins part way into a cycle.

Suppose we know where in a cycle the return pulse begins. What can we do with it? The BumbleBee transmits at 5.8 GHz—that's 5.8 billion cycles per second. Since the transmitted pulse is traveling at the speed of light, we can calculate that each cycle is about 5.6 cm, called the *wavelength*. In the figure, the pulse starts about 1/8 into a cycle which is a distance of 1/8 of a wavelength. So the pulse begins $1/8 * 5.6 = 0.7$ cm from the start of the cycle.

Now suppose the target moves slightly farther away, so that the return is delayed and now begins 1/4 into the same cycle. In this case, the pulse begins 1.4 cm from the start of the cycle. How far has the target moved? Because of the round-trip effect, it will be half of the difference. This means the target has moved $(1.4 - 0.7) / 2 = 0.35$ cm.

What can we do with this information? If we go through the transmit-wait-receive process fast enough, we can track the motion of the target with respect to the radar, and do so with a high degree of accuracy and precision. Notice however that we don't know what cycle the target is in. In our illustration above, we could shift the return a full cycle to the left or the right and get the same result. Each cycle represents a 2.8 cm interval of target displacement ($5.6 \text{ cm} / 2$ because of the round trip). Since we can't tell which cycle the target is in, we can't tell which interval it is in. Collectively, the intervals a target might be in are called the *ambiguity intervals*.

But knowing where the target is within a cycle can be put to good advantage. So long as the target is within range of the radar, we can track the amount of motion and a computer that is processing the BumbleBee's output can distinguish oscillating motion (such as a bush blowing back and forth in the

wind) from displacement (motion in one direction). We can, with enough post processing on a computer, even do such things as tell whether the target is walking, running, skipping or dancing a jig.

3 How the BumbleBee Works

Here we're going to get in to details of how the BumbleBee does its job. We aren't going to be getting into hardware details and we aren't going to be using a lot of math. The goal instead is to give you a clear idea of what's going on.

3.1 Getting the Basic Information

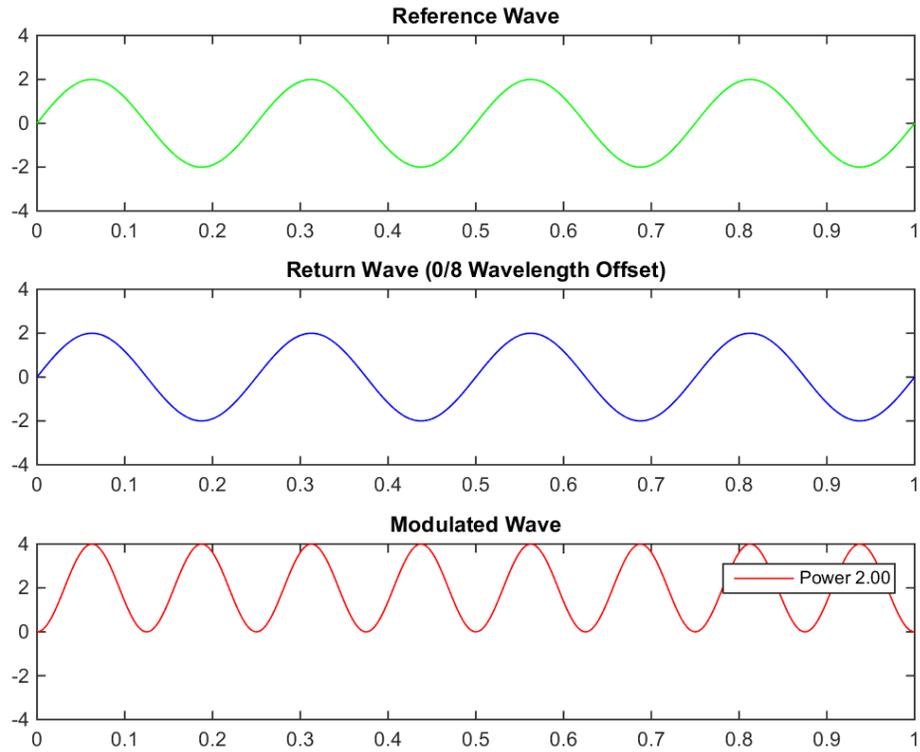
Our first job is to figure out where in a cycle the return begins. We can do this fairly easily up to a point provided we don't care what direction the target is moving and the size of the target doesn't vary. In the next section we'll see how we can resolve these limitations.

At this point, we have the reference pulse and the return pulse to work with. An important thing to keep in mind is that the reference is a sine wave and so is the return. They differ only by the offset and length of the return.

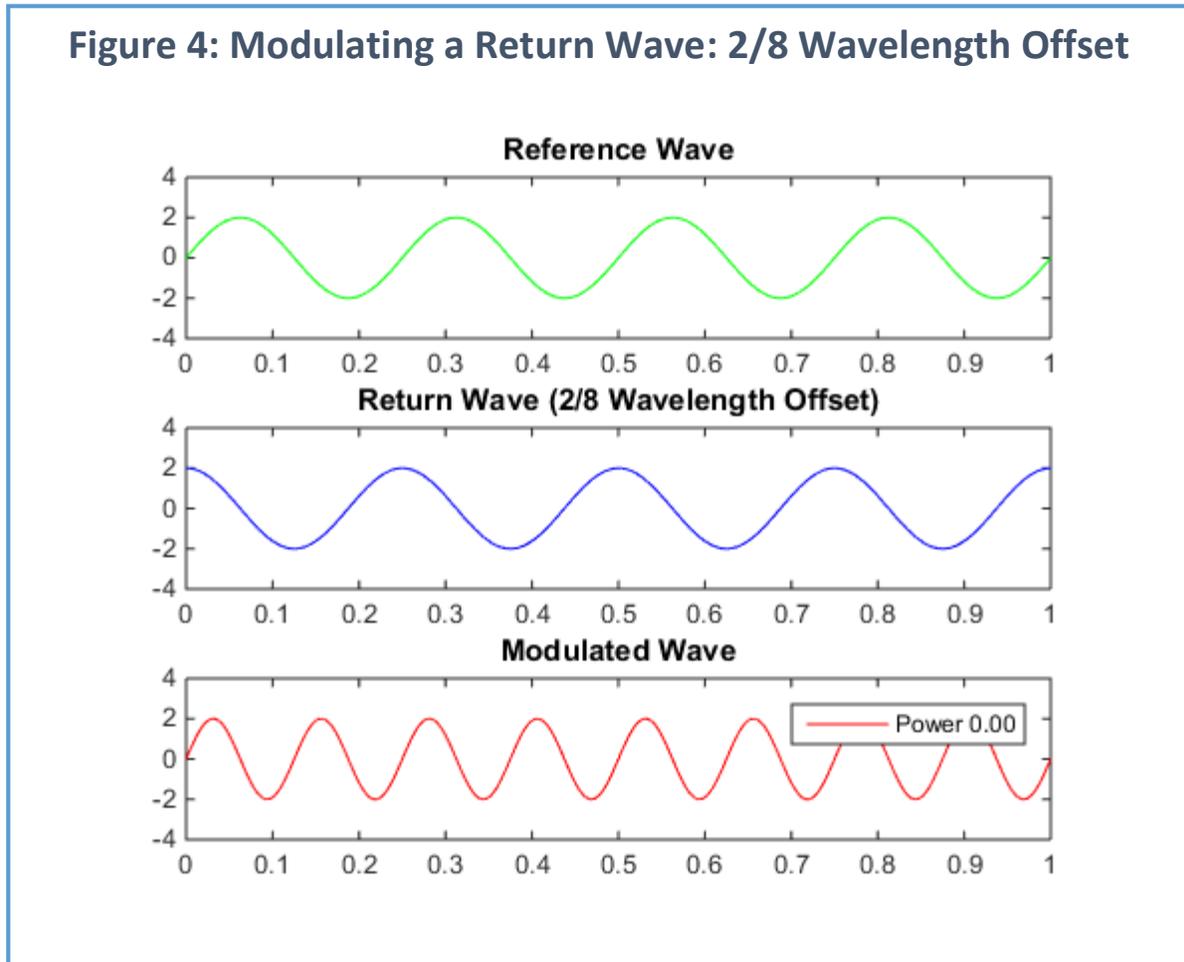
The first thing we do is multiply the wave of the return pulse (or, when unambiguous, just *return wave* or *return*) by the wave of the reference pulse (*reference wave* or *reference*), a process called *amplitude modulation*; see <http://www.dspguide.com/ch10/5.htm> for example. This will produce a new wave whose amplitude changes according to the return's offset. The figures below shows the effect of modulation. In Figure 3 the return is not offset. As you can see in the modulated wave, the amplitudes are all non-negative. If this surprises you, keep in mind that we're multiplying the waveforms, not

adding them. Hence, for example, if the reference value is -2, the return value is also -2 so the product is 4.

Figure 3: Modulating a Return Wave: Zero Offset



In Figure 4 the return is offset by $2/8$ wavelength (one quarter or 90 degrees). In this case, the modulated wave is symmetric around zero.



We can characterize these two modulated waves according to their *average power* (or just *power*). This is also known as the DC value, and is calculated by averaging the amplitudes of the wave. Hence the modulation of a reference wave with a 0-offset return wave yields an average power of 2. Modulation of the same reference wave with a $2/8$ -offset return wave yields an average power of 0. We can use this information to help determine the offset of the return and hence the relative position of the target.

Incidentally, you may have noticed that the modulated wave is the same length as the reference wave. We did this in order to make the presentation clearer. In reality the wave will be truncated, depending on when the receiver started to receive it. This will affect the average power but only slightly. Any number of whole cycles can be truncated without any change, so the variability will be with respect to the truncation of the final (partial) cycle. If a wave has enough cycles, this difference will be insignificant.

Now suppose we send out multiple pulses and calculate the power of the return. As the target moves we will see the values change. This lets us know that the target is moving but as it turns out, we can't tell what direction it's moving. Figure 5 shows the power for returns with offsets in one-eighth wavelength increments. Note that the values are symmetric around $4/8$, so for example, the value for $1/8$ is the same as for $7/8$. So if in one sample we get a power value of -2.00 and in the next -1.41, we cannot tell if

the target is moving towards or away from the radar. Nor can we tell how much it's moving: if one power value is 2.00 and the next is -1.41, the target might have moved 3/8 or 5/8 of a wavelength.

Figure 5: Power Values for One Wavelength

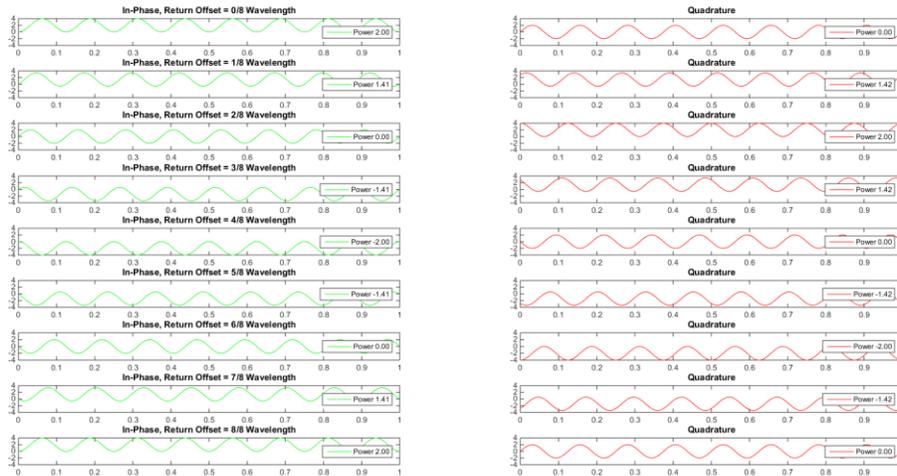
Wavelength Offset	Power
0/8	2.00
1/8	1.41
2/8	0.00
3/8	-1.41
4/8	-2.00
5/8	-1.41
6/8	0.00
7/8	1.41
8/8	2.00

3.2 Resolving Target Direction, Distance and Size

So the power value lets us know about movement but not direction or amount of movement. The trick to resolving the issue of this is to once again modulate the return but with a displaced reference. Imagine that we have two perfectly isolated radars side-by-side such that the operation of one does not interfere with the operation of the other. The second radar is set back slightly so that it's exactly one eighth of a wavelength farther than the first; because of the round-trip effect, the total path will be one quarter of a wavelength farther. Suppose both radars broadcast and receive at exactly the same times. Then the return for the second radar will be the same as that of the first except that it will be offset one quarter of a wavelength.

The results of the first radar we call In-Phase (I) and the second Quadrature (Q) (because it is offset one quarter wavelength). If we take readings for a target that moves in 1/8 wavelength steps between readings, we'll get the I and Q modulations and power values shown in Figure 6.

Figure 6: I-Q Modulation & Power @ 1/8 Wavelength Offsets

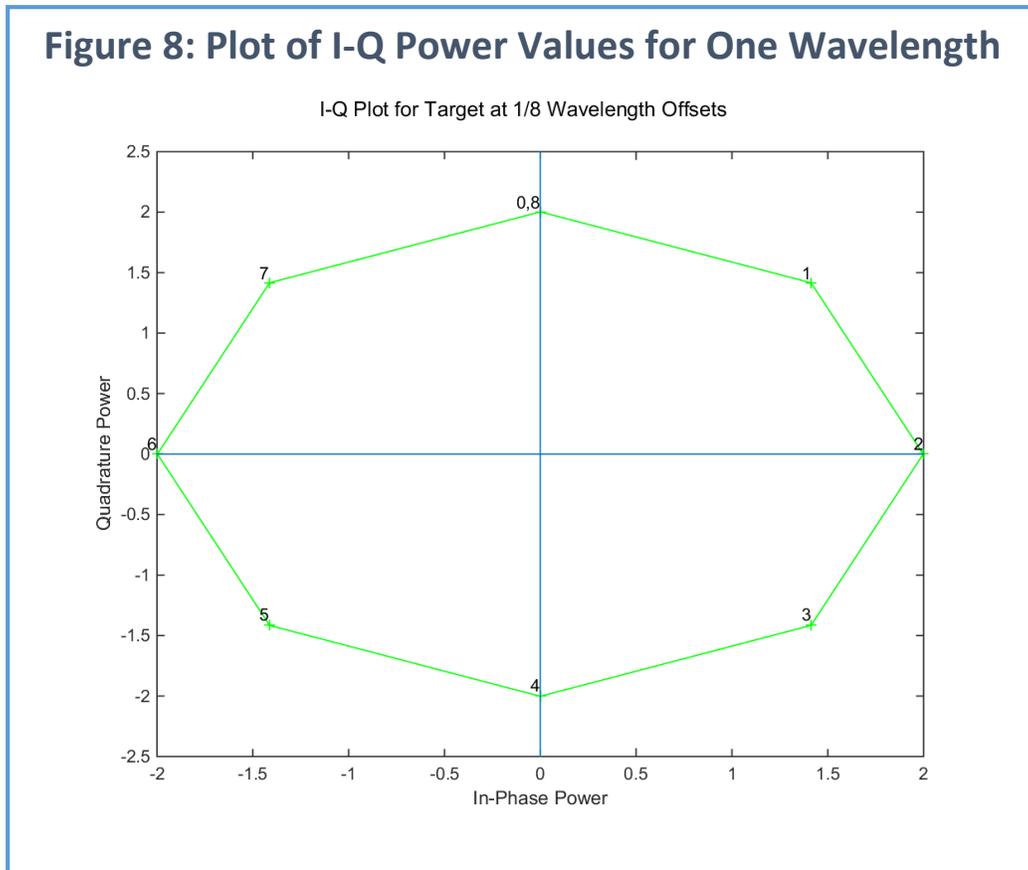


Note that if we take I and Q together, the symmetry is broken. As you can see in Figure 7, each I-Q value pair is unique. At the boundaries, 0/8 and 8/8, the values are the same as we'd expect.

Figure 7: I & Q Power Values for One Wavelength

Wavelength Offset	I Power	Q Power
0/8	2.00	0.00
1/8	1.41	1.42
2/8	0.00	2.00
3/8	-1.41	1.42
4/8	-2.00	0.00
5/8	-1.41	-1.42
6/8	0.00	-2.00
7/8	1.41	-1.42
8/8	2.00	0.00

There's something else interesting about this. Not only do the value pairs repeat from one cycle to the next, within a cycle they form a polygon shape with the vertex points in order. In Figure 8 the value pairs go clockwise from 0/8 offset through 7/8 offset and then back at 8/8 offset.

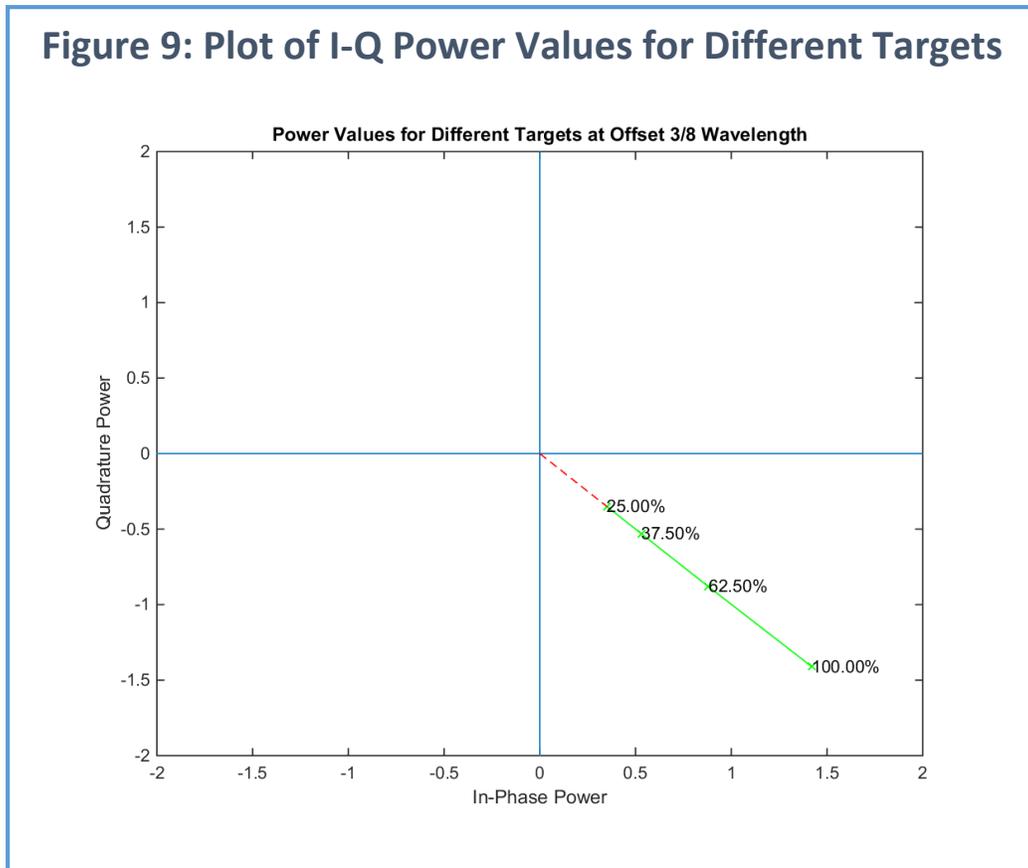


As the target moves away from the radar, the return takes longer to arrive so more of the return wave prefix will be dropped and the offset will increase. So clockwise I-Q power value pair sequences means that the target is moving away while counter-clockwise means it is moving closer. (By the way, we plot the In-Phase value on the X axis and the Quadrature power on the Y axis. This is a standard convention, which we follow here.)

Now we can tell with excellent fidelity and precision both the direction of target movement as well as distance traveled between samples.

It turns out we've also solved another problem. Hitherto we've assumed that the target is of a fixed size. Actually, it's not the size that matters so much as the *radar cross-section*, the profile presented to the radar. The power of the return wave will vary according to the cross-section of the target. So a thin, wide target (a piece of sheet metal, for instance) will have a large return power when facing the radar but a small return power when turned sideways.

The way we're measuring I-Q values lets us avoid the cross-section issue entirely. In Figure 9 we have the I-Q power values varying from a weak target that presents a small cross-section (25% of the highest power that could be returned) to a strong target that presents a large cross-section (100%). Note that these values form a straight line to the origin, so in each case the angle is the same. Thus, no matter the strength of the target, we can tell the direction of motion and the distance traveled by looking at the relative angles.



Actually getting the I and Q returns with two carefully placed radars would be tricky. Fortunately, we don't need to do that. Recall that modulation is the product of the outgoing reference wave and that of the return. The process we've just described makes a one-quarter offset by having a second, carefully positioned radar in order to delay the return. Suppose instead we were to use the same return as on original radar but instead offset the reference. Since in both cases the reference waves are the same except for an offset, the net result is the same as having two radars. And that, in fact, is what the BumbleBee does. So when it receives a return signal, it modulates against the reference and against a reference that's been offset by one-quarter wavelength. This gives us the power values for I and Q.

Now we have a way to estimate the amount of motion—the displacement over time. As we noted earlier, if the angle between two consecutive I-Q lines it is clockwise, the target's movement is towards the radar and otherwise it is away. In Figure 8, a target movement resulting in consecutive measurements for point 7 on the graph followed by point 1 is counter-clockwise and therefore the target motion is away from the radar. Conversely, a move point 1 followed by point 7 is clockwise so the target motion is towards the radar. The amount of motion can be calculated as follows, given consecutive I-Q values of P1 and P2. We do the calculation using degrees, but you can use radians instead.

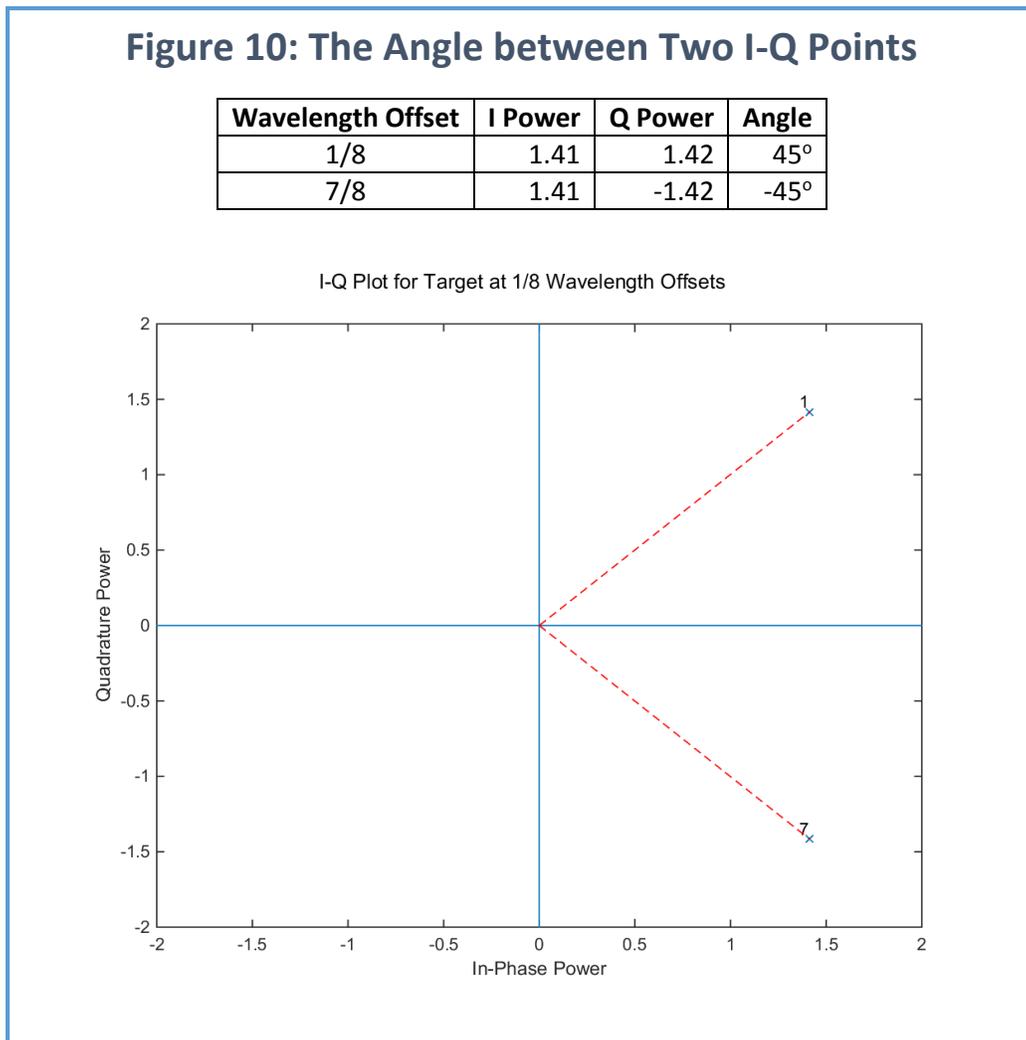
- Assuming a line from the origin to each point, calculate the angle for each point using the [atan2 function](#). This function gives an inverse tangent in all 4 quadrants; that is, it gives the angle in the range of $(-180^\circ, 180^\circ]$.

BumbleBee Radar Tutorial

- Using those angles, calculate the angle between them by subtracting the angle of P2 from that of P1. This will be positive for counter-clockwise, negative for clockwise.
- Get the ratio of the angle to that of a circle, 360° .
- Multiply the ratio by the wavelength, 5.2 cm.
- Divide that by 2 to account for the round-trip.

This gives the distance traveled between samples. The clockwise vs. counter-clockwise change lets us give the number a positive sign (for movement away from the radar) or a negative sign (towards the radar). By summing up these distance values you can tell at any time the distance of the target relative to when you started keeping track. To emphasize, this doesn't give us absolute distance from the radar but only with respect to some starting point.

To illustrate, let's pick points 1 and 7 from Figure 8 and calculate the angle as shown in Figure 10.



Using the atan2 function, we get the values shown for angle of each point. The difference between the angles of point 7 and point 1 is $-45^\circ - 45^\circ = -90^\circ$. Taking the absolute value of the ratio, $|-90^\circ / 360^\circ| = 0.25$. The motion of the target therefore is $0.25 * 5.2 \text{ cm} / 2 = 0.65 \text{ cm}$. Since the angle between the points is clockwise, the motion is towards the BumbleBee.

If we sample the BumbleBee periodically over time, we can determine with high accuracy a target's direction, speed and distance moved.

3.3 Eliminating Stationary Objects

Our discussion has assumed that there's nothing in the BumbleBee's field of view except the target. In reality, of course, there will be many other things such as hills, boulders, buildings, parked cars and the like. These are stationary but are still reflecting return waves back to the BumbleBee. We need to filter these out so that we're only dealing with the return from the target.

Suppose we have one moving target and one stationary object. The return pulse received by the BumbleBee will be a composite wave in which the returns of each are added together. Since the stationary object isn't moving, its return wave will always be the same while the moving target's wave will change. So if we can isolate the return wave we can subtract it from the composite and just be left with the target's wave. Doing this directly would be hard at low power so instead we do it indirectly. We take the I power values over a sliding window of time and calculate the average. This gives us the I power of the stationary target in isolation and we can subtract it from the current composite I power to get the target's I power. And, of course, we do the same thing with the Q power values.

As an example, we suppose a static item whose return is at $3/8$ cycle offset. A target moves away from the radar in $1/8$ cycle increments. To keep it simple, we sample 8 times per second. In Figure 11 we have the results.

The first pair of columns gives the I and Q power values for the static item. Because it's not moving, these don't change. The next triple of columns gives the offset and the I and Q power values for the target in isolation; that is, what they would be if there were no static item. The static item and the isolated target waves are added to give the composite wave, whose I and Q power values are shown in the next two columns.

The composite means for the composite wave power values are calculated for the sliding window comprising the current sample power values and those of the preceding second. The filtered target power values are those of the composite minus the composite mean.

The filtered target power values differ from the isolated target: they're lower unless zero. But what matters are not the actual I and Q power values but rather the angle they form. As we see in the last columns, the angles formed by the filtered target values will converge over time to be the same as those formed by the target in isolation. Initially they will differ but within about a second they will be very close, which means we can accurately calculate motion and direction of the target in the presence of stationary items.

Figure 11: Filtering out Stationary Items

Samp	Static Item		Isolated Target			Composite		Composite Mean		Filtered Target		Angles (degrees)	
	I Pwr	Q Pwr	Offset	I Pwr	Q Pwr	I Pwr	Q Pwr	I Pwr	Q Pwr	I Pwr	Q Pwr	Isolated Target	Filtered Target
0	0.77	1.85	0/16	2	0	2.77	1.85	2.77	1.85	0	0	0	0
1	0.77	1.85	1/16	1.85	0.77	2.61	2.61	2.69	2.23	-0.08	0.38	23	101
2	0.77	1.85	2/16	1.41	1.41	2.18	3.26	2.52	2.57	-0.34	0.69	45	116
3	0.77	1.85	3/16	0.77	1.85	1.53	3.7	2.27	2.85	-0.74	0.84	68	131
4	0.77	1.85	4/16	0	2	0.77	3.85	1.97	3.05	-1.21	0.79	90	147
5	0.77	1.85	5/16	-0.77	1.85	0	3.7	1.64	3.16	-1.64	0.54	112	162
6	0.77	1.85	6/16	-1.41	1.41	-0.65	3.26	1.31	3.18	-1.96	0.09	135	177
138	0.77	1.85	138/16	-1.41	-1.41	-0.65	0.43	0.74	1.9	-1.39	-1.47	-135	-133
139	0.77	1.85	139/16	-0.77	-1.85	0	0	0.74	1.89	-0.74	-1.89	-112	-111
140	0.77	1.85	140/16	0	-2	0.77	-0.15	0.74	1.88	0.03	-2.03	-90	-89
141	0.77	1.85	141/16	0.77	-1.85	1.53	0	0.74	1.86	0.79	-1.86	-68	-67
142	0.77	1.85	142/16	1.41	-1.41	2.18	0.43	0.75	1.85	1.43	-1.42	-45	-45
143	0.77	1.85	143/16	1.85	-0.77	2.61	1.08	0.77	1.85	1.85	-0.77	-23	-23
144	0.77	1.85	144/16	2	0	2.77	1.85	0.78	1.85	1.99	0	0	0
145	0.77	1.85	145/16	1.85	0.77	2.61	2.61	0.79	1.85	1.82	0.76	23	23
146	0.77	1.85	146/16	1.41	1.41	2.18	3.26	0.8	1.86	1.38	1.4	45	45
147	0.77	1.85	147/16	0.77	1.85	1.53	3.7	0.81	1.88	0.72	1.82	68	68
148	0.77	1.85	148/16	0	2	0.77	3.85	0.81	1.89	-0.04	1.96	90	91
149	0.77	1.85	149/16	-0.77	1.85	0	3.7	0.8	1.9	-0.8	1.8	112	114
150	0.77	1.85	150/16	-1.41	1.41	-0.65	3.26	0.79	1.91	-1.44	1.35	135	137
151	0.77	1.85	151/16	-1.85	0.77	-1.08	2.61	0.78	1.91	-1.86	0.7	157	159
152	0.77	1.85	152/16	-2	0	-1.23	1.85	0.77	1.91	-2	-0.07	180	-178
153	0.77	1.85	153/16	-1.85	-0.77	-1.08	1.08	0.75	1.91	-1.84	-0.83	-157	-156
154	0.77	1.85	154/16	-1.41	-1.41	-0.65	0.43	0.74	1.9	-1.39	-1.47	-135	-134
155	0.77	1.85	155/16	-0.77	-1.85	0	0	0.74	1.89	-0.74	-1.89	-112	-111
156	0.77	1.85	156/16	0	-2	0.77	-0.15	0.74	1.87	0.03	-2.03	-90	-89
157	0.77	1.85	157/16	0.77	-1.85	1.53	0	0.74	1.86	0.79	-1.86	-68	-67
158	0.77	1.85	158/16	1.41	-1.41	2.18	0.43	0.75	1.85	1.43	-1.42	-45	-45
159	0.77	1.85	159/16	1.85	-0.77	2.61	1.08	0.77	1.85	1.85	-0.77	-23	-23
160	0.77	1.85	160/16	2	0	2.77	1.85	0.78	1.85	1.99	0	0	0

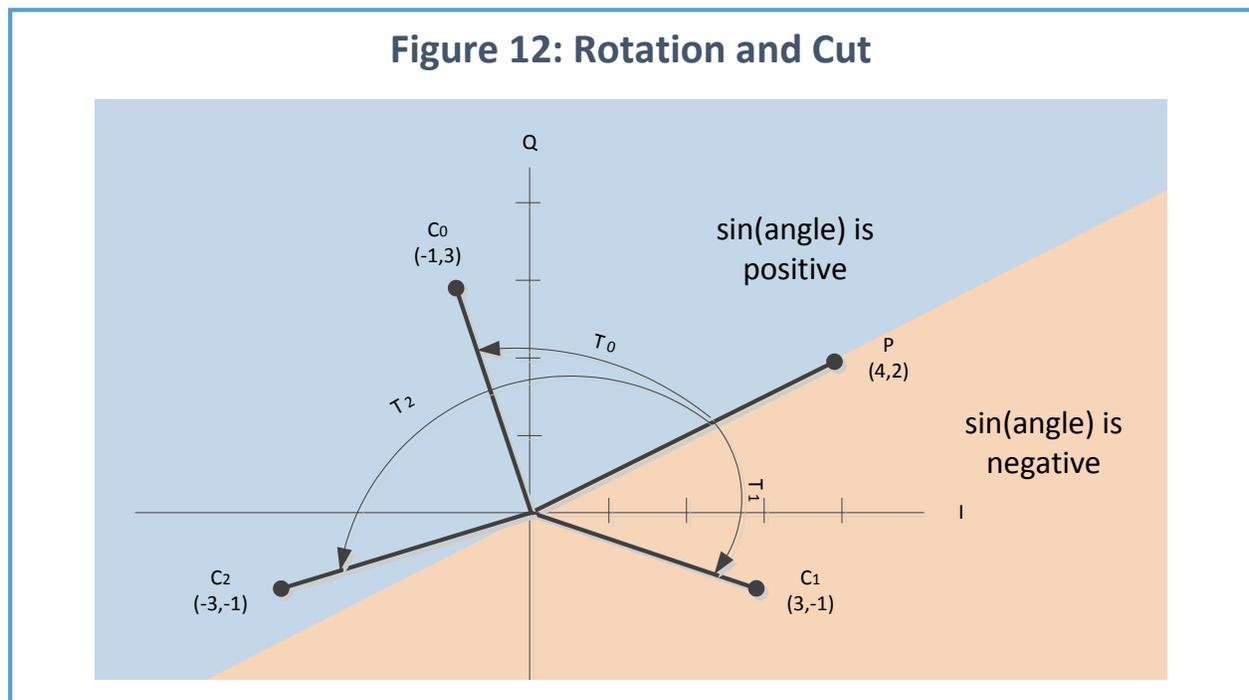
3.4 Effect of Noise and Multiple Targets

We'll cover noise and multiple targets briefly. Radio environments can be noisy—a WiFi router nearby, for example—and this can cause interference. However, when modulated by the reference wave, the mean of these signals is usually zero so they wind up having no net effect, just as for stationary targets.

Sometimes there can be two targets in range of the BumbleBee. Usually one is stronger than the other so the dominant signal will prevail. It is often possible, using signal processing techniques that we don't cover here, to separate the I-Q values and consider each target separately.

4 Cut Detection

Now that we know how to detect motion and direction, we'll look at a way to do this efficiently. In what we've discussed so far we have to calculate and subtract angles for samples. This can give us fairly precise measurement of motion but it takes a fair amount of computing power for each sample. If we relax our requirements somewhat we can get useful information at much less cost. What we do is make use of the fact that every time a target moves $\frac{1}{2}$ wavelength, the graph of the I-Q sample pairs for that wavelength go 360° , one full rotation. So we focus instead detecting times when a consecutive pair of samples crosses the negative I axis; each such crossing is called a *cut* and can be positive or negative. (The selection of that particular axis is customary but arbitrary.) In Figure 11, the sample pair (P, C_2) makes a cut in the counter-clockwise (positive) direction for target movement towards the BumbleBee.



If we were to swap the two samples, then (C_2, P) would be a cut in the clockwise (negative) direction for movement away from the BumbleBee. Each full cut represents 2.6 cm of movement by the target relative to the BumbleBee so by doing cut analysis for successive pair sample pairs we can track direction and relative distance.

Notice that we're losing a bit of information since, in a series of samples, the first cut and the last cut will be less than full. But with enough samples, this doesn't matter much. Why not just go ahead and keep track of sample-by-sample differences? There are two reasons. First, if we're sampling 250 times per second (which is a reasonable rate), we only have 4 milliseconds to do the calculation; depending on the computer that's doing the processing, that might take too long. Second, if we can switch to low

power when not processing, a more efficient processing algorithm allows more time to be spent in low power mode and that translates directly to longer battery life.

4.1 Getting the Direction of Motion

So what we need is a fast way to do cut analysis. First a bit of trig (don't let this dismay you): if we know the angle between two vectors, the sine of the angle will be positive if the direction is counter-clockwise, negative if clockwise and 0 if we can't tell (the angle is either 0° or 180°). In Figure 11, $\sin \theta_0$ is positive while $\sin \theta_1$ is negative. See <https://www.mathsisfun.com/algebra/trig-four-quadrants.html> for more about this.

As a side note, in the case of a 0° angle, no motion has taken place and we don't care. In the case of a 180° angle—half a rotation—there is motion but it's rather a lot. If the sample rate is 250 then it means that the target has moved 1.8 cm (half a wave) in 4 microseconds, or about 16 km/hr, which gives us an upper limit on the speed of the target that we can reliably track.

How do we get the sine of the angle efficiently? Again, we can simplify. We don't really need to know the sine itself, just whether it's positive or negative. So now we can leverage the following fact: the magnitude of the cross-product of two vectors equals the product of the lengths of each vector times the sine of the angle between them. In mathematical notation,

$$|P \times C| = \|P\| \|C\| \sin \theta$$

where P is the previous sample and C is the current one. In our case, the length of each vector is non-negative so the product of the two is non-negative. If the right-hand side is positive, the sine must be positive and similarly for negative. So all we have to do now is compute the cross-product.

For more information about this, see

- <https://www.youtube.com/watch?v=Tesvs6xCWZA>
- https://www.khanacademy.org/math/linear-algebra/vectors_and_spaces/dot_cross_products/v/proof-relationship-between-cross-product-and-sin-of-angle

To summarize where we are:

- If the cross-product is positive then the sine of the angle from the previous to the current sample is positive, rotation is counter-clockwise and the motion of the target is towards the BumbleBee.
- If the cross-product is negative then the sine of the angle is negative, rotation is clockwise and the motion of the target is away from the BumbleBee.

4.2 Checking for a Cut

So to check for a cut we have to do check if the negative I axis has been crossed. A cut happens if:

- The rotation is counter-clockwise, P.Q is negative and C.Q is positive. This is a cut representing motion of the target towards the BumbleBee.
- The rotation is clockwise, P.Q is positive and C.Q is negative. This is a cut representing motion away from the BumbleBee.

BumbleBee Radar Tutorial

To determine the direction of rotation, we can calculate the sign of the cross product, whose formula is as follows:

$$|P \times C| = (P.Q * C.I) - (P.I * C.Q)$$

This will either be positive (counter-clockwise rotation), negative (clockwise rotation) or zero. As earlier, we ignore the zero case. Note that avoid the use of trig functions, making this an efficient calculation.

4.3 Examples

Let's see how this works out for the values in Figure 11. P is the same in each case.

Previous		Current			Cross Product	Rotation direction	Cut?
P.I	P.Q		C.I	C.Q			
2	4	C ₀	-1	3	-14	Counter-clockwise	No
		C ₁	3	-1	10	Clockwise	No
		C ₂	-3	-1	-2	Counter-clockwise	Yes

Here,

- (P, C₀) doesn't cross the I axis at all.
- (P, C₁) crosses the I axis, but it is the positive side.
- (P, C₂) crosses the negative I axis. The direction is counter-clockwise, P.Q = 4 is positive and C₂.Q = -1 is negative.

Hence in the last case, there is a cut indicating motion towards the BumbleBee.

5 Displacement Detection

Now that we know how to determine cut and direction, we can detect displacement which, as we mentioned earlier, is motion in one direction, filtering out other kinds of motion such as a bush blowing back and forth. The approach is simple. Over a window of time—typically one second—sum up the cuts. Motion towards the radar is negative, away is positive. So if we add them up for a second, a sufficiently large absolute value (positive or negative) will indicate displacement. A person, for example, that's walking towards the radar will generate a succession of negative cuts. A bush nearby can be also generating cuts but they will be a mixture of positive and negative as the bush blows back and forth: these will tend to cancel each other out. So we can set a minimum cumulative cuts threshold of, say, 9 to detect displacement. If the sum of the cuts over the window has an absolute value of at least 9 then displacement occurs.

There is a complication however: we have to deal with environments in which there's nothing around so there is no return. In this case, the BumbleBee will produce results that are affected primarily by the noise in the signal. This can result in low values that indicate cuts and falsely detect displacement. To handle this, there is a second parameter that can be used: noise threshold. If a noise threshold value of, say, 7 is used, then the absolute values of both the I and the Q must be at least 7 before evaluating whether there is a cut or not.

The values for minimum cumulative cuts and noise threshold that are appropriate depend in part on such things as the resolution of the ADC used to read the BumbleBee's output and the degree to which the BumbleBee's output voltage range is matched by the ADC's input range. They will also depend on the environment around the BumbleBee. Placement in a bare environment (such as a large open dirt field) will result in more false alarms from noise than a residential environment that has bushes, trees, a building and the like to provide a return. As you work with the BumbleBee in your particular

environment, you'll need to experiment with different values of minimum cumulative cuts and noise thresholds to find a balance between false positives (false alarms) and false negatives (failure to detect displacement).

6 Summary

We've covered the basics of Doppler pulsed radars and have shown how to use the BumbleBee to detect the amount of target motion and direction by using the in-phase and quadrature power values. To simplify motion detection we use the idea of a cut to track the number and direction of cuts that represent rotation cuts on the graph. The BumbleBee is a low-power device and, with the optimizations of cut processing, direction and amount of motion can be done with low power on a mote-scale device.

Together, a BumbleBee paired with a low-power mote (microcontroller) can be used for an extended period of time to detect and track motion in a setting where batteries are required.