1 GSW… The Wiener Filter

This is one of the most fundamental results in all of communications engineering. It can be used to design optimum receive filters, optimum beam-forming patterns for smart antennas, optimum combination schemes for diversity receivers and many other useful things. I'll take this slowly, since it's important.

First of all, what is a Wiener filter? It's the optimum linear combination of several different versions of a signal, all corrupted by noise and interference. In this case 'optimum' means that the filter minimises the mean square difference between the desired signal and the output from the filter[1]. I'll start with the simplest example I can think of.

1.1 A Simple Example: Receive Diversity

Perhaps the simplest possible example of a problem that can be solved using a Wiener filter is a radio receiver with two antennas. Consider a system consisting of one transmitter and two receivers, where each receiver receives a copy of the same transmitted signal $s(t)$, but through a channel with a different gain (g_1 for receiver one and g_2 for receiver two), and with a totally independent noise contribution ($n_1(t)$ and $n_2(t)$ respectively). What is the best linear[2] way to combine the outputs of the two receivers $r_1(t)$ and $r_2(t)$ to produce the best possible estimate of the transmitted signal?

The scenario is illustrated in the figure below:

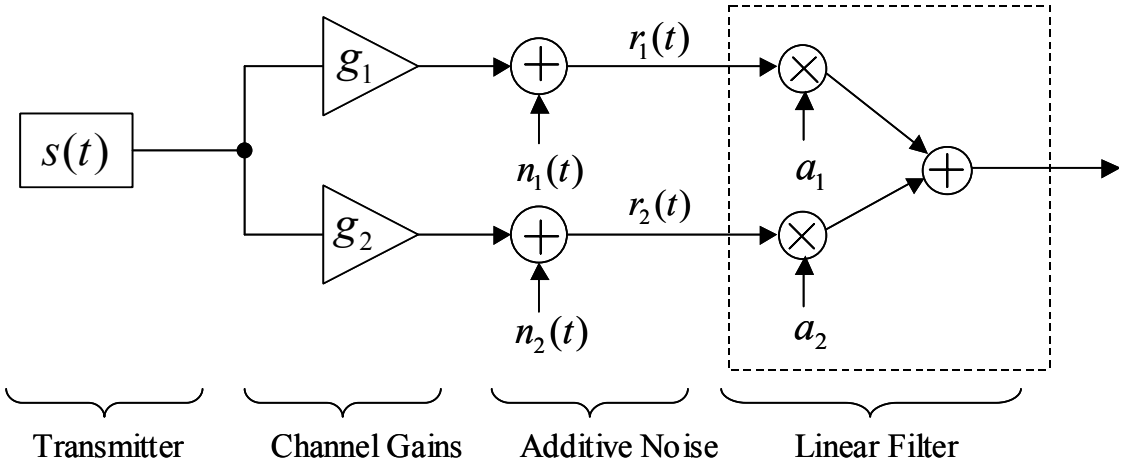

Figure 1-1 A Two-Branch Diversity Receiver with Combiner

[1] In the most general case this is not quite the same thing as minimising the number of errors when digital modulation schemes are being used, but it's very close, and the maths is a lot easier than trying to work out the real optimum filter.

[2] By 'linear' I mean that I'm only allowed to add multiples of the two received signals together. I can't add a constant, or the square of either received signal, or use any other function. I'm looking for something of the form $Ar_1(t) + Br_2(t)$, where A and B are constants.

First, I need to define what I mean by 'best possible'. For the Wiener filter, 'best possible' means minimising the mean square of the difference between the combined receive signal and the transmitted signal, in this case:

$$\text{mean square error} = \mathrm{E}\left\{e^2\right\} = \mathrm{E}\left\{\left(s(t)-\left(a_1 r_1(t)+a_2 r_2(t)\right)\right)^2\right\} \tag{0.1}$$

and for this reason, Wiener filters of this type are sometimes called *minimum mean square error* (MMSE) filters.

This is a minimisation problem with two independent variables (a_1 and a_2), so we can solve it by differentiating this expression with respect to each variable in turn, in each case setting the result equal to zero to look for a turning point, which will be the minimum we're looking for[3].

Then, noting that we can reverse the order of the differentiation and the averaging[4], the differentiation is just the function of a function:

$$\frac{d\mathrm{E}\left\{e^2\right\}}{da_1} = \mathrm{E}\left\{2\left(s(t)-\left(a_1 r_1(t)+a_2 r_2(t)\right)\right)\left(-r_1(t)\right)\right\} \tag{0.2}$$

and setting the differential equal to zero for a turning point gives:

$$\mathrm{E}\left\{s(t)r_1(t)-a_1 r_1^2(t)-a_2 r_2(t)r_1(t)\right\}=0 \tag{0.3}$$

Then using the result that the mean of the sum of random variables is equal to the sum of the means, and re-arranging this in terms of a_1 gives:

$$\begin{aligned}\mathrm{E}\left\{a_1 r_1^2(t)\right\} &= \mathrm{E}\left\{s(t)r_1(t)\right\}-\mathrm{E}\left\{a_2 r_2(t)r_1(t)\right\}\\ a_1 &= \frac{\mathrm{E}\left\{s(t)r_1(t)\right\}-\mathrm{E}\left\{a_2 r_2(t)r_1(t)\right\}}{\mathrm{E}\left\{r_1^2(t)\right\}}\end{aligned} \tag{0.4}$$

Doing an exactly similar calculation to find the value of a_2 that gives a minimum value for the mean square error gives the result:

$$a_2 = \frac{\mathrm{E}\left\{s(t)r_2(t)\right\}-\mathrm{E}\left\{a_1 r_1(t)r_2(t)\right\}}{\mathrm{E}\left\{r_2^2(t)\right\}} \tag{0.5}$$

---

[3] It can't be a maximum, since the expression is a quadratic in terms of a_1 and a_2, which means there is only one turning point, and clearly, as a_1 or a_2 tends to plus or minus infinity, the error term will tend towards positive infinity. The only turning point we get in both cases must therefore be a minimum. There's more discussion about this point later.

[4] See the chapter on "WYNTKA Random Variables" if you're not sure about this: both differentiating and taking the mean are linear operations, so the order doesn't matter.

and that gives two equations in two unknowns, which we can solve by substituting equation (0.5) into equation (0.4), and after a lot of tedious algebra, we get:

$$a_1 = \frac{\mathrm{E}\{s(t)r_1(t)\}\mathrm{E}\{r_2^2(t)\} - \mathrm{E}\{s(t)r_2(t)\}\mathrm{E}\{r_2(t)r_1(t)\}}{\mathrm{E}\{r_1^2(t)\}\mathrm{E}\{r_2^2(t)\} - \mathrm{E}\{r_1(t)r_2(t)\}^2} \tag{0.6}$$

This is a general result, true for a lot of different scenarios, including some cases with interference as well as just noise (we haven't yet considered whether there is any correlation between the noises $n_1(t)$ and $n_2(t)$). However, here the only signal impairments are the different gains of the two channels from the transmitter to the receivers, and the uncorrelated noise. If we write the received signals $r_1(t)$ and $r_2(t)$ in terms of the transmitted signal, the gains of the channels and the noise, we get:

$$\begin{aligned} r_1(t) &= g_1 s(t) + n_1(t) \\ r_2(t) &= g_2 s(t) + n_2(t) \end{aligned} \tag{0.7}$$

and since in this case the noise is uncorrelated between the two receivers, and also uncorrelated with the signal, we can write:

$$\mathrm{E}\{r_2(t)n_1(t)\} = \mathrm{E}\{s(t)n_1(t)\} = \mathrm{E}\{r_1(t)n_2(t)\} = \mathrm{E}\{s(t)n_2(t)\} = 0 \tag{0.8}$$

and substituting equations (0.7) into equations (0.8) gives:

$$\begin{aligned} \mathrm{E}\{r_1^2(t)\} &= g_1^2\mathrm{E}\{s^2(t)\} + \mathrm{E}\{n_1^2(t)\} \\ \mathrm{E}\{r_2^2(t)\} &= g_2^2\mathrm{E}\{s^2(t)\} + \mathrm{E}\{n_2^2(t)\} \\ \mathrm{E}\{r_1(t)r_2(t)\} &= g_1 g_2 \mathrm{E}\{s^2(t)\} \\ \mathrm{E}\{s(t)r_1(t)\} &= g_1\mathrm{E}\{s^2(t)\} \\ \mathrm{E}(s(t)r_2(t)) &= g_2\mathrm{E}(s^2(t)) \end{aligned} \tag{0.9}$$

Now if we let $S = \mathrm{E}\{s^2(t)\}$ be the mean power in the transmitted signal (assuming it is not a function of time), and write the mean noise powers at the two receivers as $N_1 = \mathrm{E}\{n_1^2(t)\}$ and $N_2 = \mathrm{E}\{n_2^2(t)\}$ respectively, then we can substitute in S, N_1 and N_2, and get:

$$a_1 = \frac{g_1 S(g_2^2 S + N_2) - g_1 g_2^2 S^2}{(g_1^2 S + N_1)(g_2^2 S + N_2) - g_1^2 g_2^2 S^2} \tag{0.10}$$

$$a_1 = \frac{g_1 S N_2}{g_1^2 S N_2 + g_2^2 S N_1 + N_1 N_2} \tag{0.11}$$

and following through the same steps for a_2 (or just noting that by symmetry we can swap all the '1's and '2's in the subscripts), we get:

$$a_2 = \frac{g_2 SN_1}{g_1^2 SN_2 + g_2^2 SN_1 + N_1 N_2} \tag{0.12}$$

(note the denominator is the same in the expression for a_1 and a_2, it's just the numerator that changes). That might seem like a lot of algebra, but remember that was about the simplest possible case of interest. Most interesting cases have a lot more than two different versions of the received signal, and the noise in the inputs may be correlated. We're going to need a different way to do these calculations. However, before we talk about that, there are a couple of points of interest about even this 'simplest' result.

1.1.1 An Interesting Special Case: $g_2 = 0$

If one of the receive signals contains none of the signal at all (say $g_2 = 0$), you might expect that the contribution of this signal to the final best estimate would be zero. After all, including any amount of this 'signal' in the final output would only add more noise. You would be right: $g_2 = 0$ does imply that $a_2 = 0$.

What about a_1? If $g_2 = 0$ then:

$$a_1 = \frac{g_1 SN_2}{g_1^2 SN_2 + g_2^2 SN_1 + N_1 N_2} = \frac{g_1 SN_2}{g_1^2 SN_2 + N_1 N_2} = \frac{g_1 S}{g_1^2 S + N_1} \tag{0.13}$$

which is not, as you might have expected, equal to $1 / g_1$. Surely if a channel has a loss of, say, 3 dB, and there is only one receiver, then the optimum receiver would have a gain of 3 dB?

Well, no. If there wasn't any noise, then this would be true: set N_1 to zero in the above equation, and you'll find that a_1 does indeed equal $1 / g_1$. But as soon as there is any noise, the optimum gain of the receiver is less than you would expect. The problem is that any gain in the receiver is amplifying the noise too, and that's increasing the total amount of noise in the output of the filter. In terms of the mean-square error, you're actually better off having slightly less signal, and less noise.

Another way of thinking about this: suppose that as well as $g_2 = 0$, g_1 was very small, so there was very little signal arriving at the receiver at all. Having a very large value of a_1 would then mostly just amplify up the noise, resulting in a very large minimum square error. You'd be better off with a value of $a_1 = 0$: then the output of the filter would be zero, and the minimum square error would just be equal to the expectation value of the signal power, S. A large value of a_1 could give a minimum mean square error much larger than that[5].

In this case, the output of this Wiener filter is:

[5] This is one time when the optimum filter (from the minimum bit error point of view) and the optimum filter (from the minimising the mean square error point of view) are different. Having a zero gain would be a good idea from the MMSE point of view, but it guarantees a bit error rate of 50%.

$$\begin{aligned} a_1\left(g_1 s(t)+n_1(t)\right) &= \frac{g_1 S\left(g_1 s(t)+n_1(t)\right)}{{g_1}^2 S+N_1} \\ &= \frac{{g_1}^2 S}{{g_1}^2 S+N_1} s(t)+\frac{g_1 S}{{g_1}^2 S+N_1} n_1(t) \end{aligned} \tag{0.14}$$

There's something interesting here too: in the case where $s(t)$ has a constant value (say, s) and $n_1(t)$ has a zero mean, the expectation (mean) value of the output of this filter would be:

$$a_1\left(g_1 s(t)+n_1(t)\right) = \frac{{g_1}^2 S}{{g_1}^2 S+N_1} s \tag{0.15}$$

which is less than s. We could say that this minimum mean square error filter is *biased*. It tends to underestimate the value of any signal component.

(Setting $a_1 = 1 / g_1$ in these circumstances gives

$$a_1\left(g_1 s(t)+n_1(t)\right) = s+\frac{1}{g_1} n_1(t) \tag{0.16}$$

and while this is an unbiased receiver since the mean value of the output is now equal to s, it has a larger mean square error.)

1.1.2 Signal to Noise Ratio at the Filter Output

There is a particularly neat result for the signal to noise ratio at the output of a Wiener filter. The input signal to noise ratio at the first receiver $SNR_1 = {g_1}^2 S / N_1$. At the second receiver, it's $SNR_2 = {g_2}^2 S / N_2$. At the output of the filter, it's:

$$SNR_{out} = \frac{\mathrm{E}\left\{\left(a_1 g_1 s(t)+a_2 g_2 s(t)\right)^2\right\}}{\mathrm{E}\left\{\left(a_1 n_1(t)+a_2 n_2(t)\right)^2\right\}} \tag{0.17}$$

and since the noise is uncorrelated, the expectation value of the product $n_1(t)\, n_2(t)$ is zero, and this gives:

$$SNR_{out} = \frac{\left(a_1 g_1+a_2 g_2\right)^2 \mathrm{E}\left\{s^2(t)\right\}}{\mathrm{E}\left\{\left({a_1}^2 {n_1}^2(t)+{a_2}^2 {n_2}^2(t)\right)\right\}} \tag{0.18}$$

Substitute in for S, N_1, N_2, a_1 and a_2 (and going through yet more tedious algebra) produces the result:

$$
\begin{aligned}
SNR_{out} &= \frac{\left(g_1 SN_2 g_1 + g_2 SN_1 g_2\right)^2 S}{\left(g_1 SN_2\right)^2 N_1 + \left(g_2 SN_1\right)^2 N_2} \\
&= \frac{\left(g_1{}^2 N_2 + g_2{}^2 N_1\right)^2 S}{\left(g_1{}^2 N_2 + g_2{}^2 N_1\right) N_2 N_1} \\
&= \frac{g_1{}^2 S}{N_1} + \frac{g_2{}^2 S}{N_2} \\
&= SNR_1 + SNR_2
\end{aligned}
\tag{0.19}
$$

Or in words: the signal to noise ratio at the output of the Wiener filter is the sum of the signal to noise ratios at the inputs to the filter. This simple result remains true for any number of receivers, provided the noises and interference at each receiver input are uncorrelated.

1.1.3 Uniqueness and Convex Surfaces

The previous analysis found one minimum point, one value for a_1 and a_2 which gave a minimum for $\mathrm{E}\{e^2\}$. How do we know that this is the global minimum? Perhaps there is somewhere else, some other values of a_1 and a_2 that give another turning point indicating another minimum that is lower than the value derived above?

Well, there isn't. There never is. This is another beauty of the approach of taking the minimum mean square error: the error function is *quadratic* in each of the unknown variables (the filter coefficients a_n). In other words, if I hold everything constant except one of the variables, say a_1, I can write the mean square error as:

$$
\begin{aligned}
\mathrm{E}\left\{e^2\right\} &= \mathrm{E}\left\{\left(s(t) - \left(a_1 r_1(t) + a_2 r_2(t)\right)\right)^2\right\} \\
&= a_1{}^2\,\mathrm{E}\left\{r_1{}^2(t)\right\} - a_1\,\mathrm{E}\left\{2\,r_1(t)\left(s(t) - a_2 r_2(t)\right)\right\} + \mathrm{E}\left\{\left(s(t) - a_2 r_2(t)\right)^2\right\}
\end{aligned}
\tag{0.20}
$$

This is in the form:

$$
\mathrm{E}\left\{e^2\right\} = A a_1{}^2 + B a_1 + C \tag{0.21}
$$

which is a quadratic function of a_1: the highest power is a_1 squared. It looks like a parabola, and it only has one turning point. Since the term A is the expectation value of a square, it must be positive, and this means the turning point is a minimum. Thinking about this in another way, consider the second derivative of the mean square error term:

$$
\frac{d^2\,\mathrm{E}\left(e^2\right)}{da_1{}^2} = 2A = 2\,\mathrm{E}\left\{r_1{}^2(t)\right\} \tag{0.22}
$$

The expectation value of the square of any quantity is always positive, so the of change of the gradient of the graph of $\mathrm{E}(e^2)$ against a_1 will be constant, and positive, all the time, everywhere. The same is true for a_2, and if there were more versions of the received signal to consider, the same would be true for the filter gains of all the others as well.

A three-dimensional plot of the mean square error as a function of a_1 and a_2 produces a graph like this:

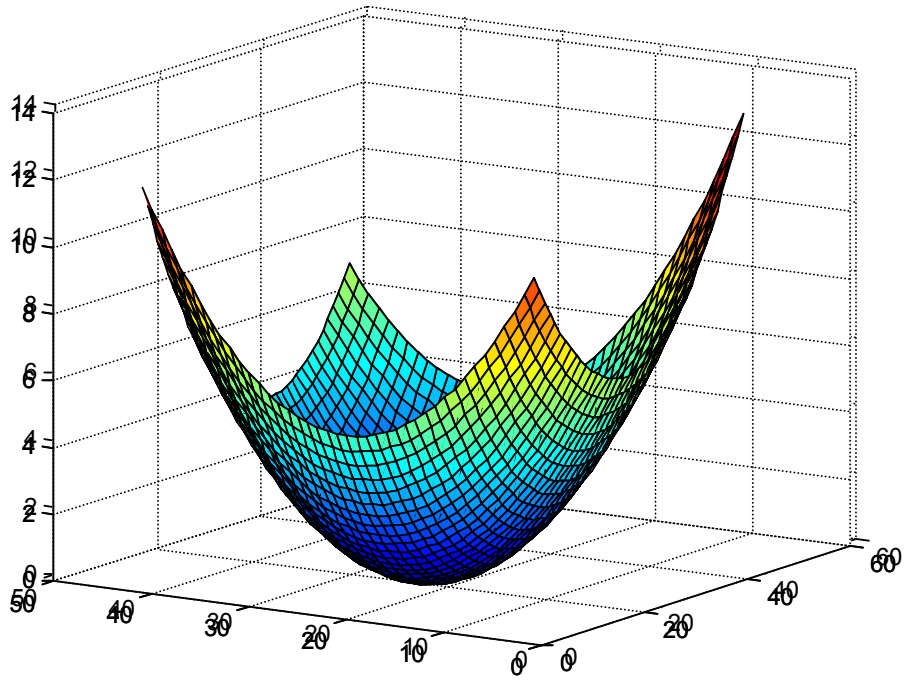

Figure 1-2 A Convex Three-Dimensional Quadratic

which is known as a convex surface, it can only have one global minimum.

1.1.4 Is the Wiener Filter Really the Best One to Use?

Good question. It's certainly the optimum filter at minimising the mean square error of the received signal using a linear combination of the inputs, but this is not necessarily what you want. For example, consider again the case where $g_2 = 0$, so effectively there is just a single input, and the signal is corrupted by Gaussian noise.

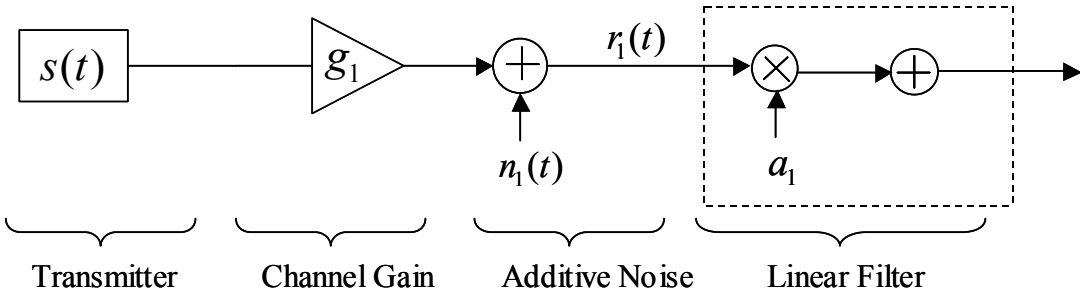

Figure 1-3 The Wiener Filter with Only One Input

Following through the analysis as before gives the result:

$$a_1 = \frac{g_1 S}{g_1^2 S + N_1} \tag{0.23}$$

Now suppose that this is a digital transmission scheme, where the signal s(t) has a value of one when a '1' bit is being transmitted, and a value of zero when a '0' bit was being transmitted.

After the filter, a decision is made on whether the transmitted bit was a '0' or a '1' depending on whether the filtered received signal is greater or less than 0.5. The task of the filter is to minimise the probability of a bit error.

The output of the filter has a mean value of zero when a '0' bit is being transmitted, and a mean value of $a_1 g_1$ when a '1' is being transmitted. In both cases, there is also noise with a variance of N_1 added, so the standard deviation of the noise in the output of the filter is:

$$\sigma = a_1 \sqrt{N_1} \tag{0.24}$$

The probability that there is a bit error when a '0' is being transmitted is the probability that the noise has a value greater than 0.5:

$$p_e(0) = Q\left(\frac{0.5}{\sigma}\right) \tag{0.25}$$

and the probability that there is a bit error when a '1' is being transmitted is the probability that the noise has a value less than $(0.5 - a_1 g_1)$:

$$p_e(1) = 1 - Q\left(\frac{0.5 - a_1 g_1}{\sigma}\right) \tag{0.26}$$

If '0' bits and '1' bits are equally likely to be transmitted, this gives a mean probability of bit error of:

$$p_e = \frac{p_e(0) + p_e(1)}{2} = \frac{1}{2}\left(1 - Q\left(\frac{0.5 - a_1 g_1}{a_1\sqrt{N_1}}\right) + Q\left(\frac{0.5}{a_1\sqrt{N_1}}\right)\right) \tag{0.27}$$

and differentiating this with respect to a_1 gives:

$$\frac{dp_e}{da_1} = \frac{1}{2}\left(\frac{0.5}{{a_1}^2\sqrt{2\pi N_1}}\exp\left(-\frac{0.5^2}{2{a_1}^2 N_1}\right) - \frac{0.5}{{a_1}^2\sqrt{2\pi N_1}}\exp\left(-\frac{(0.5 - a_1 g_1)^2}{2{a_1}^2 N_1}\right)\right) \tag{0.28}$$

Setting this to zero to find the 'optimum' value of a_1 to use then gives:

$$0.5^2 = (0.5 - a_1 g_1)^2 \tag{0.29}$$

so either $a_1 = 0$, which can't be the minimum we're looking for, since it would make the bit error rate = 0.5 all the time, or:

$$a_1 = \frac{1}{g_1} \tag{0.30}$$

That's not the MMSE result given by the Wiener filter. In practice the difference is not that important: plotting the difference in bit error rates as a function of the value of g_1 for a given value of noise (in this case a standard deviation of one) gives:

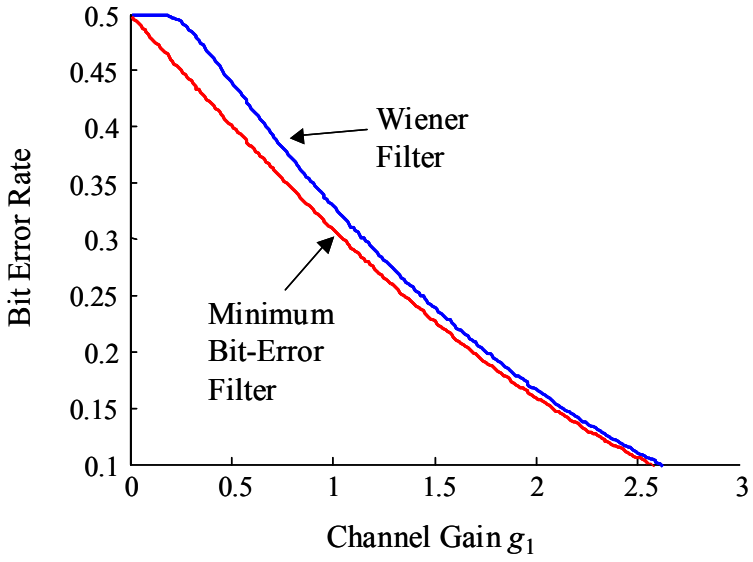

Figure 1-4 Bit Error Rates of Wiener and Optimum Filter

For the range of values where the difference is noticeable, there are so many errors that the communication scheme is most likely completely useless anyway. While it's relatively simple to calculate the optimum minimum bit-error-rate filter for this case, in general this calculation is much more difficult than calculating the Wiener filter, so with greater numbers of inputs, it's much easier (an therefore cheaper) to use the Wiener filter instead.

1.2 The General Case

In general, we don't have just two versions of a signal available to the filter to combine, we can have any number N of them. This results in N simultaneous equations in the N unknown gains to solve, and we have to start using matrix notation.

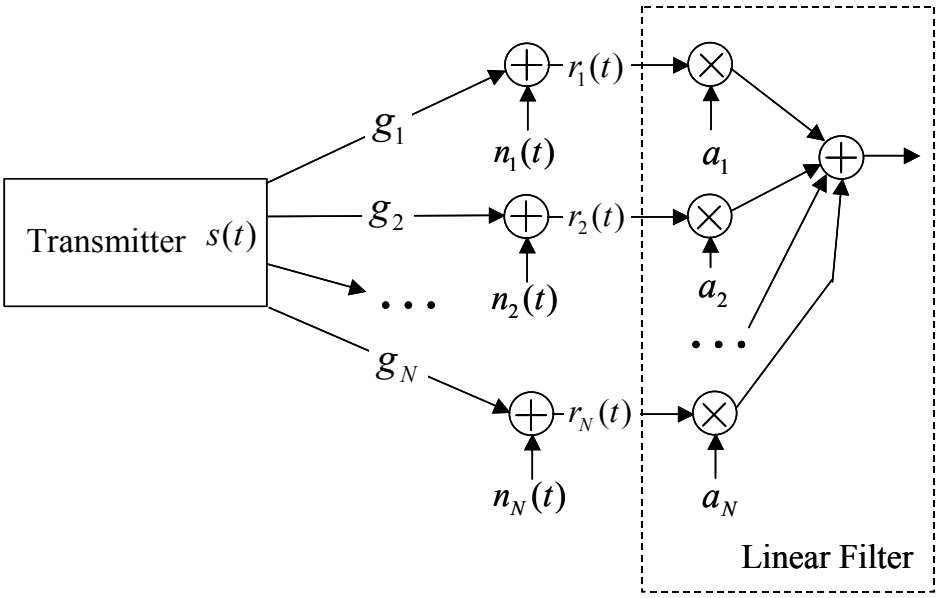

Figure 1-5 An *N*-Branch Diversity Receiver with Combiner

Suppose the i^{th} receiver gets a received signal of $g_i s(t)$, adds noise $n_i(t)$ and has a filter gain of a_i. Then, the total mean square error at the output of the filter is:

$$\text{mean square error} = \mathrm{E}\left\{e^2\right\} = \mathrm{E}\left\{\left(s(t)-\sum_{i=1}^{N} a_i r_i(t)\right)^2\right\} \tag{0.31}$$

We can proceed as before to differentiate this to find the turning points, in this case to find the optimum value for one particular filter co-efficient a_j, we differentiate with respect to a_j:

$$\frac{d\mathrm{E}\left\{e^2\right\}}{da_j} = \mathrm{E}\left\{2\left(s(t)-\sum_{i=1}^{N} a_i r_i(t)\right)\left(-r_j(t)\right)\right\} \tag{0.32}$$

and for a turning point (which must again be a minimum), this must be equal to zero, so:

$$\mathrm{E}\left\{s(t)r_j(t)\right\} = \mathrm{E}\left\{r_j(t)\sum_{i=1}^{N} a_i r_i(t)\right\} \tag{0.33}$$

This leads to a set of N simultaneous equations, one for each possible value of j. We can write these in matrix notation, by first defining a cross-correlation vector **P** with elements of the expectation value of the product of the transmitted signal $s(t)$ and the received signals $r_j(t)$:

$$\mathbf{P}_j = \mathrm{E}\left\{s(t)r_j(t)\right\} \tag{0.34}$$

and an autocorrelation matrix **R**, with elements of the expectation value of the product of each received signal with all of the other received signals:

$$\mathbf{R}_{j,i} = \mathrm{E}\left\{r_j(t)r_i(t)\right\} \tag{0.35}$$

Post-multiplying this matrix **R** with the vector **a** (which has elements of the filter co-efficients a_j), gives a vector with elements:

$$\mathbf{R}\,\mathbf{a} = \sum_{i=1}^{N} \mathrm{E}\left\{r_j(t)r_i(t)\right\} a_i = \mathrm{E}\left\{r_j(t)\sum_{i=1}^{N} a_i r_i(t)\right\} \tag{0.36}$$

So we can simply write all the simultaneous equations we're trying to solve as:

$$\mathbf{P} = \mathbf{R}\,\mathbf{a} \tag{0.37}$$

and provided we can invert this matrix, we can equally easily write the solution we want as:

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{P} \tag{0.38}$$

This is known as the Wiener-Hopf equation, and it's one of the most important results in signal processing. All we have to do to solve it is work out all the cross-correlations between the input signals (and hence produce the vector **P**) and all the auto-correlations between all the different versions of the received signals (and hence the matrix **R**), and then invert the matrix **R**. Of course, we don't usually have all that information, and even if we did, this involves a lot of calculations, and takes a lot of processor power to do.

1.2.1 Working out P and R

At this point you could raise a very fair objection. It's simple enough for a filter to work out the autocorrelation matrix **R**: all it has to do is multiply every input to the filter by every other input, and calculate the average value of each of the products.

But how can a filter at the receiver work out the cross-correlation vector **P** if it doesn't know what the transmitted signal was? Two techniques are used to solve this problem: either a set of known symbols are transmitted by the transmitter to allow the receiver to work out **P** and hence 'train' the equaliser (these are known as *training sequences* or *pilot symbols*), or the receiver has to try and work it out using some iterative technique: knowing the set of possible transmit symbols, it tries to minimise the error between the received signal and the nearest possible transmit symbol.

(In fact, due to the time it takes to invert the matrix **R**, iterative techniques are often used in practice that avoid the need to calculate the matrix **R** and the cross-correlation vector **P**. See the chapters on the LMS and RLS algorithms for more details.)

1.3 Complex Signals, Complex Losses and Complex Gains

If we're going to use all this theory with passband communication signals using the equivalent baseband technique, we need to extend the result to consider complex signals, complex channel losses and complex filter coefficients. We also have to slightly modify what we mean by 'mean square error'.

Since the signal and the output of the filter are now both complex numbers, just taking the square or the error term would result in another complex number. What the Wiener filter does in these circumstances is to try and minimise the square of the magnitude of the difference between the transmitted signal and the filter output: that corresponds to the square of the length of the vector separating the two points on the Argand diagram:

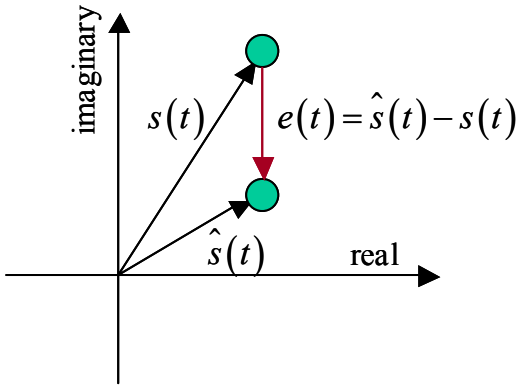

Figure 1-6 Complex Errors between Complex Signals

Since the square of the magnitude of a complex number z is (z z*) where z* indicates the complex conjugate of z, the expression we have to minimise is now:

$$\mathrm{E}\left\{e^2\right\} = \mathrm{E}\left\{\left(s(t)-\sum_{i=1}^{N}a_i r_i(t)\right)\left(s^*(t)-\sum_{i=1}^{N}a_i^* r_i^*(t)\right)\right\} \qquad (0.39)$$

Deriving the complex form of the Wiener-Hopf equation is slightly more difficult than for the real case, since we can't simply differentiate with respect to a complex number: we've got a term in a_i^*, and you can't differentiate the complex conjugate of a number with respect to the complex number, the answer depends on which direction you're moving[6].

In these cases we have to consider both the real and imaginary components of a_i separately. Let $a_j = x_j + jy_j$, and first differentiate with respect to the real component x_j. This is the differentiation of a product, so:

$$\frac{d\mathrm{E}\left(e^2\right)}{dx_j} = \mathrm{E}\left\{\left(s(t)-\sum_{i=1}^{N}a_i r_i(t)\right)\left(-r_j^*(t)\right)+\left(s^*(t)-\sum_{i=1}^{N}a_i^* r_i^*(t)\right)\left(-r_j(t)\right)\right\} \quad (0.40)$$

and for a turning point,

$$\mathrm{E}\left\{r_j(t)s^*(t)+r_j^*(t)s(t)\right\} = \mathrm{E}\left\{r_j(t)\sum_{i=1}^{N}a_i^* r_i^*(t)+r_j^*(t)\sum_{i=1}^{N}a_i r_i(t)\right\} \quad (0.41)$$

If I do exactly the same thing except differentiating with respect to y_j, I get:

$$\frac{d\mathrm{E}\left\{e^2\right\}}{dy_j} = \mathrm{E}\left\{\left(s(t)-\sum_{i=1}^{N}a_i r_i(t)\right)\left(+j\,r_j^*(t)\right)+\left(s^*(t)-\sum_{i=1}^{N}a_i^* r_i^*(t)\right)\left(-j\,r_j(t)\right)\right\} \quad (0.42)$$

which when looking for a turning point (to find a minimum value), gives:

$$\mathrm{E}\left\{r_j(t)s^*(t)-r_j^*(t)s(t)\right\} = \mathrm{E}\left\{r_j(t)\sum_{i=1}^{N}a_i^* r_i^*(t)-r_j^*(t)\sum_{i=1}^{N}a_i r_i(t)\right\} \quad (0.43)$$

These results can be written in a very convenient way using the identities:

$$x^*y+xy^* = x^*y+\left(x^*y\right)^* = 2\Re\left\{x^*y\right\} \quad (0.44)$$

$$x^*y-xy^* = x^*y-\left(x^*y\right)^* = 2\Im\left\{x^*y\right\} \quad (0.45)$$

which when applied to equations (0.41) and (0.43) gives:

$$\Re\left\{E\left\{r_j(t)s^*(t)\right\}\right\} = \Re\left\{E\left\{r_j(t)\sum_{i=1}^{N}a_i^* r_i^*(t)\right\}\right\} \quad (0.46)$$

[6] Let the complex number $z = x + jy$, where x and y are both real. Then $z^* = x - jy$. Moving a small distance along the real axis gives a new value of $z^* = (x + \delta x) - jy$, so that the change in z is δx, and dividing this by the distance travelled gives 1. However, moving a small distance along the imaginary axis gives a new value of $z^* = x - j(y + \delta y)$, so that the change in z is $-j\delta y$, and dividing this by the distance travelled gives –1. The function $f(z) = z^*$ is said to be 'non-differentiable'.

$$\Im\left\{E\left\{r_j(t)s^*(t)\right\}\right\} = \Im\left\{E\left\{r_j(t)\sum_{i=1}^{N} a_i^* r_i^*(t)\right\}\right\} \tag{0.47}$$

and if both the real and imaginary parts of these expressions are equal, then clearly:

$$E\left\{r_j(t)s^*(t)\right\} = E\left\{r_j(t)\sum_{i=1}^{N} a_i^* r_i^*(t)\right\} \tag{0.48}$$

If we modify the definition of the correlation vector **P** such that[7]:

$$\mathrm{P}_j = \mathrm{E}\left\{r_j^*(t)s(t)\right\} \tag{0.49}$$

and of the autocorrelation matrix **R** such that[8]:

$$\mathrm{R}_{j,i} = \mathrm{E}\left\{r_j^*(t)r_i(t)\right\} \tag{0.50}$$

Then multiplying this matrix **R** with the vector **a** (which has elements of the filter co-efficients a_j), gives a vector with elements:

$$\mathbf{R}\,\mathbf{a} = \sum_{i=1}^{N} \mathrm{E}\left\{r_j^*(t)\, r_i(t)\right\} a_i = \mathrm{E}\left\{r_j^*(t)\sum_{i=1}^{N} a_i r_i(t)\right\} \tag{0.51}$$

So we can write all the simultaneous equations we're trying to solve as:

$$\mathbf{P}^* = \left(\mathbf{R}\,\mathbf{a}\right)^* \tag{0.52}$$

and again, provided we can invert this matrix, we can write the solution we want as:

$$\mathbf{a} = \left(\mathbf{R}^{-1}\mathbf{P}\right) \tag{0.53}$$

If all the signals, channel gains and filter gains are real, then this gives exactly the same form as the original Wiener-Hopf equation shown above[9].

---

[7] This is entirely reasonable, since for complex signals, the correlation is usually defined as:

$$\phi_{ab}(\tau) = \int_{-\infty}^{\infty} a^*(t)\,b(t+\tau)\,dt$$

[8] This is entirely reasonable as well – see the last footnote.

[9] Although note that if I'd defined the cross-correlation vector the other way around, so that:

$$\mathrm{P}_j = \mathrm{E}\left\{s^*(t)r_j(t)\right\}$$

the Wiener-Hopf equation would have been:

[continued on next page…]

1.4 The Principle of Orthogonality

One of the interesting things about the optimum solution indicated by the Wiener-Hopf equation is that the error is orthogonal to all the inputs to the filter. By orthogonal, I mean that the expectation value of the product of the error and the received signal is zero. Since for real signals and gains:

$$\text{error} = s(t) - \sum_{i=1}^{N} a_i r_i(t) \tag{0.54}$$

and we get from equation (0.32) that for a turning point:

$$\frac{d\mathrm{E}\left\{e^2\right\}}{da_j} = \mathrm{E}\left\{2\left(s(t) - \sum_{i=1}^{N} a_i r_i(t)\right)\left(-r_j(t)\right)\right\} = 0 \tag{0.55}$$

we can quickly deduce that for the minimum error:

$$\mathrm{E}\left\{e(t) r_j(t)\right\} = 0 \tag{0.56}$$

and this is true for all the inputs to the filter. With complex signals, channel gains and filter gains, the result becomes:

$$\mathrm{E}\left\{e^*(t) r_j(t)\right\} = 0 \tag{0.57}$$

(for proof, see the problems).

1.5 Discrete Wiener Filters

In real life, signal processing operation such as correlations and filtering are done by DSP processors with sampled versions of the input signals, not time-continuous versions. This doesn't make a great deal of difference in terms of the derivation of the Wiener filters. All that happens is that any integration over time (to find the expectation values) is replaced by a summation over the samples available.

The cross-correlation vector between the inputs to the filter and the desired signal becomes:

$$\mathrm{P}_j = \sum_n r_j^*(n) s(n) \tag{0.58}$$

and the autocorrelation matrix between the various inputs to the filter becomes:

$$\mathrm{R}_{j,i} = \sum_n r_j^*(n) r_i(n) \tag{0.59}$$

$$\mathbf{a} = \left(\mathbf{R}^{-1}\mathbf{P}^*\right)$$

Everything else is the same.

1.6 Tutorial Questions

1) A transmit signal encodes a binary '0' as a voltage of one volt, and a binary '1' as a voltage of two volts. The signal is sent down a cable of gain g, and the equivalent input noise at the receiver has a mean-square value of N V^2. What is the optimum gain of the receiver in the minimum least-squares case?

(Assume the probability of transmitting a '1' and a '0' are equal, and the noise is uncorrelated with the data. Try and work this out from first principles.)

2) What if the voltages transmitted by the transmitter were zero volts for a '0' and one volt for a '1'? Would the optimum gain of the receiver change? How about if the voltages were 0.4 V for a '0' and 2.2 V for a '1'?

3) Go back to the voltage levels of question 1: one volt for a '0' and two volts for a '1'. If you had a circuit after the receiver that treated all incoming signals above 1.5 V as a '1', and all signals below 1.5 V as a '0', and you were trying to minimise the probability of a bit error, what should the gain of the receiver be?

Is this the same as the gain of the receive filter that minimises the mean-square error? If not, is the Wiener filter always the best solution to use in practice?

4) Prove that the principle of orthogonality holds for complex signals and filter gains.

5) The solution for $\mathbf{x}$ that minimises the amplitude of the vector $\mathbf{y} - \mathbf{A}\mathbf{x}$ is given by the normal equation: $\mathbf{x} = (\mathbf{A}^H\mathbf{A})\mathbf{A}^H\mathbf{y}$. The Wiener filter is the solution for the vector of taps $\mathbf{a}$ that minimises the expectation value of the expression $s(t) - \mathbf{a}^{\mathrm{T}}r(t)$.

Are these two results related? If so, show how, and how the form of the solution of the Wiener filter could be derived from the normal equation.

6) What is the minimum mean bit-error-rate filter for a transmission scheme that sends a value of –1 for a data '0', and a value of +1 for a data '1', when the receiver picks up two versions of the signal with uncorrelated noise with a standard deviation of one, and the channel gains to the two receivers are 1.5 and 0.8 respectively.

What error rate results in this case, and how much better is this than using the Wiener filter in the same situation?