

# 11. BCH Codes: Constructing them and finding the Syndrome of a Message

We have noted that a two error correcting encoding polynomial can be created by multiplying a primitive polynomial,  $p(x)$ , by another polynomial  $p_3(x)$  defined by the condition

$$\text{rem } p_3(x^3) = 0, \text{ on dividing by } p(x),$$

and codes allowing correction of 3 errors can be obtained by having a similar factor with 3 replaced by 5 as well as this one, and so on.

We will show that this is true by showing how we can find and correct errors. First we consider the question:

**How do we find polynomials  $p_3$ ,  $p_5$  and so on, to create such codes? We illustrate this procedure for 3?**

We want to find an equation of lowest degree obeyed by the remainders of the powers of  $x$  that are divisible by three. This equation will be a linear dependence among the remainders of these powers.

There is a standard way to find a linear dependence among a set of vectors. It is called row reduction.

To do it, you write down the vectors and add an appropriate multiple (0 or 1) of the first vector to each of the others so as to eliminate some one component from them all. You then add an appropriate multiple of the second vector to eliminate another component from the rest. And so on.

If your vectors are linearly dependent, eventually one of your vectors will become the (0) vector. Your equation will then be that this vector originally was the sum of the others that you added to it to get the (0) vector.

We illustrate this procedure for the primitive polynomial  $1 + x^2 + x^5$ .

The first step is to write down the remainders of the first few powers that are divisible by 3.

These can be read off from the remainder table for that polynomial, or we can construct a similar table whose successive rows represent powers that increase by 3.

Let us notice how to construct such a table. Similar tables for this and for other powers will be very useful to us soon,

We will start from the 0<sup>th</sup> power whose remainder is 1. But suppose in general that the present power has remainder  $a + bx + cx^2 + dx^3 + ex^4$ .

What happens when this remainder is multiplied by  $x^3$ ?

We get as new remainder  $ax^3 + bx^4 + c(x^2 + 1) + d(x^3 + x) + e(x^4 + x^2)$ , or

$$c + dx + (c + e)x^2 + (a + d)x^3 + (b + e)x^4.$$

We can use this fact to write a table whose successive entries are  $x^0, x^3, x^6, \dots$

However we get them we find that the remainders of the powers divisible by 3 won dividing by our primitive polynomial are

power	remainder	2	3	4	
0	1	0	0	0	0
3	0	0	0	1	0
6	0	1	0	1	0
9	0	1	0	1	1
12	0	1	1	1	0
15	1	1	1	1	1

Remainders of powers dividing by  $(1+x^2+x^5)$

Since these rows form 6 vectors in a five dimensional space, there must be a linear dependence among them. It is easy to see here that the sum of all but the power 3 is the 0 vector.

(Doing it by row reduction, we can use the 0 power remainder to remove the 0<sup>th</sup> power from the rest, the 3 power one to eliminate power 3, the power (6+3) row to eliminate the first power column, the 9+6+3+3<sup>th</sup> to eliminate the 4<sup>th</sup> power column the 12<sup>th</sup> +6<sup>th</sup> to eliminate the 2<sup>nd</sup>, and we find that the 15<sup>th</sup> is the 0<sup>th</sup> plus 3<sup>rd</sup> + 3<sup>rd</sup> +6<sup>th</sup> +9<sup>th</sup> + 6<sup>th</sup> + 12<sup>th</sup> +6<sup>th</sup>, which boils down to everything except the 3<sup>rd</sup> power.)

We deduce then that, on dividing by  $1 + x^2 + x^5$ , the remainder of

$$1 + x^6 + x^9 + x^{12} + x^{15}$$

is 0, from which we conclude that

$$p_3(x) = 1 + x^2 + x^3 + x^4 + x^5$$

is the second factor we want to create our encoding polynomial for a two error correcting code:

$$(1 + x^2 + x^5)(1 + x^2 + x^3 + x^4 + x^5).$$

Exactly the same approach can be used to obtain similar factors starting from any primitive polynomial, and powers that are multiples of any odd numbers.

You might wonder at this point, what happened to even powers? We have found an equation that has  $x^3$  as a root here. How about  $x^2$ ? Or  $x^4$ ?

We are not concerned with even powers here because they will obey the same equations in the sense used here as their square roots. Thus  $x^2$  obeys the same equation as  $x$ , and so does  $x^4$  and  $x^8$  and  $x^{16}$ .

How come?

If the remainder of  $(x^5 + x^2 + 1)$  is 0, then on squaring both sides of this equation we deduce that the remainder of  $(x^{10} + x^4 + 1)$  is also 0, which says that  $x^2$  obeys the same equation as  $x$ .

Similarly,  $x^6$  obeys the same equation as  $x^3$ .

## 11.2 Finding Two or More Errors: Step 1: Finding the Error Syndrome

Suppose we encode using an encoding polynomial  $p(x)$   $p_3(x)$ . And suppose further that the received word  $R(x)$  was garbled and has at most 2 errors. Then we will have

$$R(x) = m(x)p(x)p_3(x) + ax^{e1} + bx^{e2}$$

where  $a$  and  $b$  are each either 0 or 1 and  $e1$  and  $e2$  are the two error powers if there are two errors.

If we take the remainder of  $R(x)$  on dividing by  $p(x)$  the first or message term here will give remainder 0, so we will find the remainder of the error terms.

Similarly, if we take each power that appears in  $R(x)$  and replace it by its cube, we will create the polynomial  $R(x^3)$ , and as a result of the factor  $p_3(x)$  in the encoding polynomial, the first term above,  $m(x^3)p(x^3)p_3(x^3)$  will also have 0 remainder on dividing by  $p$ .

**This means that the remainder of  $R(x^3)$  will be that of  $ax^{3e1} + bx^{3e2}$  on dividing by  $p$ .**

To summarize, with a two error correcting code, we can find not only the sum of the remainders of the error monomials, but also the sum of the third powers of the error monomials.

In the identical way, we can find the sum of the fifth powers of the error monomials in a three error correcting code and so on.

We have to address two questions at this point.

**First, how can we actually find the remainder of  $R(x^3)$  conveniently?**

**Second, how do we use the remainders of sums of odd powers of the errors to find the errors themselves?**

The answer to the first question is exactly like the way we find the remainder of the sum of the errors in a single error correcting code or here as well.

**To find the sum of the errors we took the dot product of R with the remainder table starting from 1 in which the powers increase by 1 from row to row,**

**To find the sum of the third powers of the errors we take the dot product of R with the similar table in which the powers increase by 3 from row to row.**

We illustrate that here for the code that we have been discussing

If the third power remainder table is displayed on a spreadsheet with each remainder in a row, taking the dot product with it can be accomplished with one instruction plus copying and adding mod 2

Suppose you have constructed an error locator for the single error correcting code defined by your primitive polynomial, which means you have done something like the following.

You have listed received message in a column and put the product of the top entry in that message with the top left entry of the remainder table somewhere in its row, (having put a dollar sign on the column index of the message,) and copy down the message and across the table. Then summed mod 2 beneath the resulting columns.

This is illustrated here for the polynomial  $(1 + x^2 + x^5)$

remainder finder		remainder						taking dot product					error	mC
r	power	0	1	2	3	4	identifier	$rT$					power	
1	0	1	0				1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	2	0	1	0	0	0	0	1

	2	0	0	1	0	0	4	0	0	0	0	0	0	0	0
0	3	0	0	0	1	0	8	0	0	0	0	0	0	0	0
1	4	0	0	0	0	1	16	0	0	0	0	1	0	1	
0	5	1	0	1	0	0	5	0	0	0	0	0	0	0	
1	6	0	1	0	1	0	10	0	1	0	1	0	0	1	
	7	0	0	1	0	1	20	0	0	0	0	0	0	0	
	8	1	0	1	1	0	13	0	0	0	0	0	0	0	
1	9	0	1	0	1	1	26	0	1	0	1	1	0	1	
	10	1	0	0	0	1	17	0	0	0	0	0	0	0	
	11	1	1	1	0	0	7	0	0	0	0	0	0	0	
	12	0	1	1	1	0	14	0	0	0	0	0	0	0	
	13	0	0	1	1	1	28	0	0	0	0	0	0	0	
1	14	1	0	1	1	1	29	1	0	1	1	1	0	1	
	15	1	1	1	1	1	31	0	0	0	0	0	0	0	
	16	1	1	0	1	1	27	0	0	0	0	0	0	0	
	17	1	1	0	0	1	19	0	0	0	0	0	0	0	
	18	1	1	0	0	0	3	0	0	0	0	0	0	0	
	19	0	1	1	0	0	6	0	0	0	0	0	0	0	
	20	0	0	1	1	0	12	0	0	0	0	0	0	0	
	21	0	0	0	1	1	24	0	0	0	0	0	0	0	
	22	1	0	1	0	1	21	0	0	0	0	0	0	0	
	23	1	1	1	1	0	15	0	0	0	0	0	0	0	
	24	0	1	1	1	1	30	0	0	0	0	0	24	1	
	25	1	0	0	1	1	25	0	0	0	0	0	0	0	
	26	1	1	1	0	1	23	0	0	0	0	0	0	0	
	27	1	1	0	1	0	11	0	0	0	0	0	0	0	
	28	0	1	1	0	1	22	0	0	0	0	0	0	0	
	29	1	0	0	1	0	9	0	0	0	0	0	0	0	
	30	0	1	0	0	1	18	0	0	0	0	0	0	0	
	31	1	0	0	0	0	1								
error		pwr						2	3	1	3	3			<b>24</b>
Error rem		and id						0	1	1	1	1	<b>30</b>		

To do all this requires entering the information on the power 1 row needed to construct the remainder table,

making one entry to get the rows to be summed in making the dot product

Summing and computing mod 2;

then one entry each for the next two columns

The rest is all copying.

Here are the entries used to get the tables above, with the received message  $r$  starting in R18

r	power	0	1	2	3	4
1	0	1	0			
1	=B18+1	=G18	=C18	=MOD(D18+G18,2)	=E18	=F18

This allows construction of the remainder table for this code.

				taking dot product rT		
identifier						
	=C18*1+D18*2+E18*4+F18*8+G18*16			=A18*C18		=A18*D18
	=C19*1+D19*2+E19*4+F19*8+G19*16			=A19*C19		=A19*D19

This gives the identifier and the start of taking the dot product.

error	mC
power	
=IF(\$O\$51=H18,B18,0)	=MOD(A18+IF(\$O\$51=H18,1,0),2)
=IF(\$O\$51=H19,B19,0)	=MOD(A19+IF(\$O\$51=H19,1,0),2)

This allows identifying the error power and goes the error correction. Here O51 is the identifier of the remainder of r. The dot product is computed using (copied into all 5

=A47*C47	=A47*D47
=A48*C48	=A48*D48
=SUM(J18:J49)	=SUM(K18:K49)
=MOD(J50,2)	=MOD(K50,2)

columns of the table. O51 is the identifier of the last row here, which can be gotten by copying any row identifier from the remainder table.

Notice that you can change codes (of the same length) by changing the entries of line 19 of the remainder table and copying them down.

Now, if you have all this, you can find the identifier and power of the sum of the third powers of the message monomials by

**creating a table like the remainder table whose powers go up by 3 each time to the right of what you have already done,**

**and then copying everything that you have done except for the last error correcting column (that is, copying the dot product computation and the power locator) to the right of the original remainder table in the same places to the right of this new table.**

**You can then read off the sums of the third powers of the monomials of r as remainders,**

**by their identifiers or by their powers.**

If you wanted to correct three errors, you could add a table in which the powers go up by 5 each time and copy once more to its right, and you can find the remainders of the fifth powers of the error monomials, if your code is such that your encoded messages have the 0 for such remainders. And so on.

Essentially all the work required lies in constructing the appropriate power remainder table.

**Next we have to describe how you can use this information.**

**Exercise: Construct a spreadsheet that for a two error correcting code of the kind we are considering, given some primitive polynomial of degree 5 other than  $1+x^2+x^5$  and a received word  $r$ , produces the sum of the error monomials and the sum of their third powers, both as remainders, but also by their identifiers and by their powers.**