ECC 2nd homework

Dongwhee Kim
Taewon Park

CRC simulation (1 / 3)

- Objective: Verify the error detection capability of CRC (Cyclic Redundancy Check)

APPENDIX III

DATA FOR SOME REPRESENTATIVE CODES

Detection Capabilities	k_{max}	$n-k$	$P(X)$	Reference
Any odd number of errors	any value	1	$1+X$	Theorem 2
Two errors, a burst of length 4 or less, 88 per cent of the bursts of length 5, 94 per cent of longer bursts*	11	4	$1+X+X^4$	Theorems 3, 5, 6
Two errors, a burst of 9 or less, 99.6 per cent of the bursts of length 10, 99.8 per cent of longer bursts	502	9	$1+X^4+X^9$	Theorems 3, 5, 6
Two bursts of length 2 or less, any odd number of errors, a burst of 5 or less, 93.8 per cent of the bursts of length 6, 96.9 per cent of longer bursts†	10	5	$(1+X+X^4)(1+X)=1+X^2+X^4+X^5$	Theorems 2, 5, 6, 7
Two bursts of combined length 12 or less, any odd number of errors, a burst of 22 or less, 99.99996 per cent of the bursts of length 23, 99.99998 per cent of longer bursts	22495	22	$(1+X^2+X^{11})(1+X^{11})=1+X^2+X^{13}+X^{22}$	Theorems 2, 5, 6, 8
Any combination of 6 or fewer errors, a burst of length 11 or less, 99.9 per cent of bursts of length 12, 99.95 per cent of longer bursts	12	11	$1+X^2+X^4+X^5+X^6+X^{10}+X^{11}$	Theorems 5, 6, and footnote 1
Any combination of 7 or fewer errors, any odd number of errors, a burst of length 31 or less, all but about 1 in 10^9 of longer bursts	992	31	$(1+X)(1+X^3+X^{10})(1+X+X^2+X^3+X^{10})(1+X^2+X^3+X^8+X^{10})$	Theorems 2, 5, 6, and footnotes 9, 12, 18

* *Note:* $1+X+X^4$ belongs to $e=15$ and $11+4=15$.
† *Note:* This is the code used in all examples.

CRC simulation (2 / 3)

1. Write CRC encode and decode codes
   - **In this assignment, polynomials are represented as arrays, like the example on the right** e.g. $x^5 + x^4 + x^2 + 1$ → [1,1,0,1,0,1]
   - **The `encode` function takes a message as input and returns a codeword**
   - **The `decode` function takes `receive` as input and returns 1 if an error is detected, otherwise 0**
   - **After writing the `encode` and `decode` functions, verify their functionality using the `test_functionality` function**

2. Implement the `get_period` function
   - Write a code to find the minimum value of **e** such that **g(x)|** $\mathbf{x^e + 1}$

3. Write the Monte-Carlo simulation code
   - Iterate for the given number of iterations (the more iterations, the more accurate the results)
   - Measure the occurrence and the detection rate of CRC for each error type
   - Refer to the `test_functionality` code for guidance
   - Note: When generating errors randomly, set the error occurrence probability to 0.5
     - Use p=0.5 in `np.random.binomial` input
   - If a specific error belongs to multiple types, count it for each type

CRC simulation (3 / 3)

- Generator polynomials : $x \quad + x^{4} + x^{2} + 1$, $x^{8} + x^{7} + x^{6} + x^{4} + x^{2} + 1$
- Display the output as shown in the illustration below (`tqdm` is an option)

```
Test the generator polynomial [1 1 0 1 0 1]
Period is 15

100%|██████████| 2000000/2000000 [01:22<00:00, 24121.11it/s]

Odd error(s)                  : 100.00% [1000879 / 1000879]
Double errors                 : 100.00% [6436 / 6436]
Burst error (length < n-k+1) : 100.00% [11752 / 11752]
Burst error (length = n-k+1) : 93.90% [8955 / 9537]
Burst error (length > n-k+1) : 96.87% [1916743 / 1978664]
```

Example : $x^5 + x^{4} + x^{2} + 1$ result