

纠错编码入门 Project-3

1. 目标：实现卷积码编码和 BCJR 译码。
2. 测试：BPSK 调制（即 $0 \rightarrow +1$; $1 \rightarrow -1$ ）和 AWGN 信道。
3. 基本类的情况说明如下

```
class CConvCodec
{
public:
    CConvCodec();
    virtual ~CConvCodec();
//卷积码的维度 k，长度 n，以及校验数目 n-k
    int m_codedim;
    int m_codelen;
    int m_codechk;
    double m_coderate;
//卷积码有 m_len_input 个输入 m_len_output 个输出
    int m_len_input;           //length of the inputs
    int m_num_input;           //number of the inputs
    int m_len_output;          //length of the outputs
    int m_num_output;          //number of the outputs

//Trellis 中的状态和边的数目
    int m_num_state;           //number of the states
    int m_num_edge;           //number of the edges
//下面参数用于描述卷积码的网格图
//编码和译码均基于 Trellis
//Trellis 从文件读入
    int *m_left_vertex;
    int *m_in_label;
    int *m_out_label;
    int *m_right_vertex;
    int m_len_trellis;         //length of the trellis
//下面参数用于 BCJR 译码算法
    double **m_alpha;
    double **m_beta;
    double **m_gamma;
    void Malloc(int len_uu, int code_no, char *file_name);
    void Free();
    void Encoder(int *uu, int *cc);
    void SoftInHardOut(double *in_bit_probs, int *uu_hat);
};
```

4. 例子

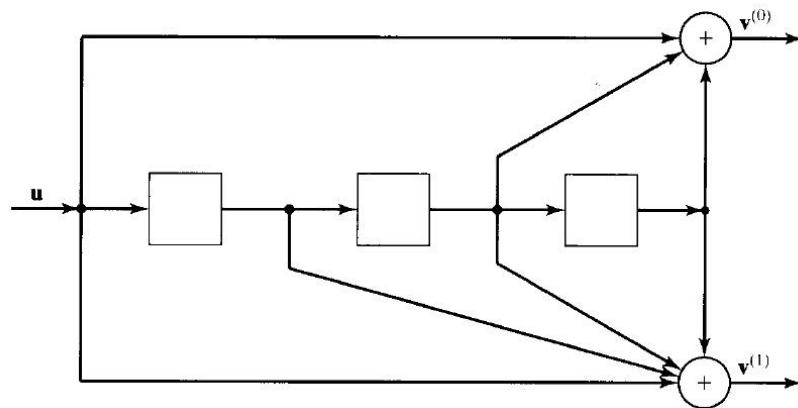


FIGURE 11.1: A rate $R = 1/2$ binary nonsystematic feedforward convolutional encoder with memory order $m = 3$.

对于上图定义的卷积码，我们有，

```
m_num_state = 2*2*2 = 8;
```

```
m_len_input = 1;
```

```
m_num_input = pow(2, m_len_input) = 2;
```

```
m_len_output = 2;
```

```
m_num_output = pow(2, m_len_output) = 4;
```

```
m_num_edge = m_num_state * m_num_input = 16;
```

如果对输入信息序列的每 $N = 1000$ 个比特截断进行编码且不归零处理，则

```
m_len_trellis = N/m_len_input;
```

```
m_codedim = N = 1000;
```

```
m_codelen = m_len_trellis * m_len_output = 2000;
```

```
m_codechk = m_codelen - m_codedim = 1000;
```

5. 译码

```
void SoftInHardOut(double *in_bit_probs, int *uu_hat);
```

中 `in_bit_probs` 包含每个比特的概率，`uu_hat` 为译码输出的序列。

对于 BPSK + AWGN 信道，对于第 t 个比特 `uu[t]`，如果对应收到的 `yy[t]`，则有

```
if (yy[t] > 0)
```

```
    in_bit_probs[t] = 1.0 / (1.0 + exp(-2.0*yy[t]/var));
```

```
else
```

```
    in_bit_probs[t] = 1.0 - 1.0 / (1.0 + exp(2.0*yy[t]/var));
```

这里的 `var` 为高斯噪声的方差。

注：需采用前面的调制映射方式。

6. 提交

代码：需提交全部**源代码**以及可运行文件(.o 也可)。

报告：在提交的报告中需给出测试的卷积码的**编码图**和对应的**性能曲线**，至少测试两个卷积码且误比特率至少到 10^{-6} 。

对于每一个卷积码，需测试不同的 N 对卷积码性能的影响，如 N=100, 500 和 1000。

为确保仿真数据真实可靠，建议对于任意一个信噪比点至少仿真 1000 个比特错误。

提交至 zhaoday@mail2.sysu.edu.cn