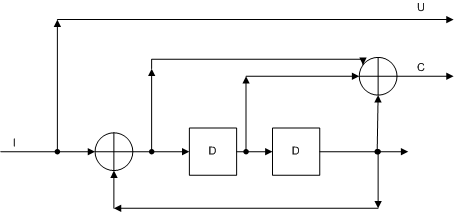
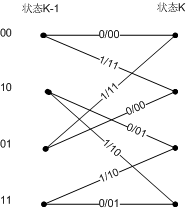
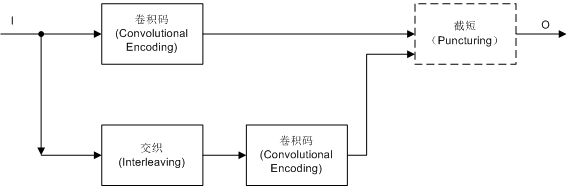
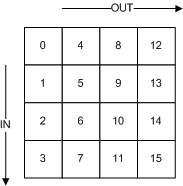
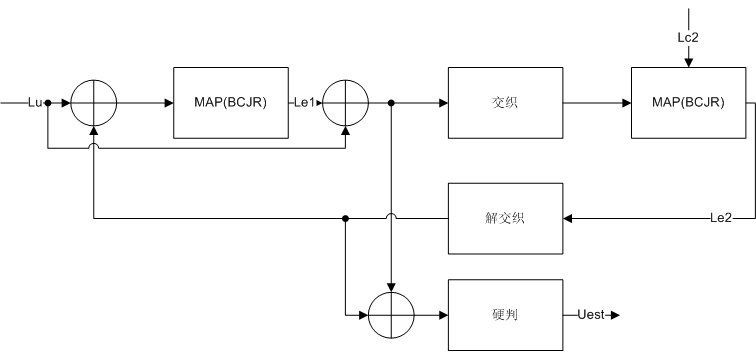
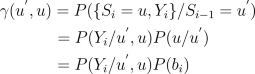
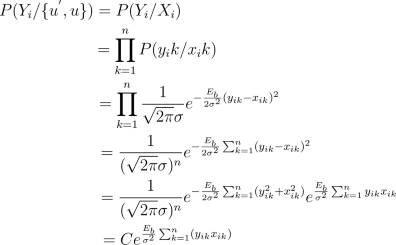
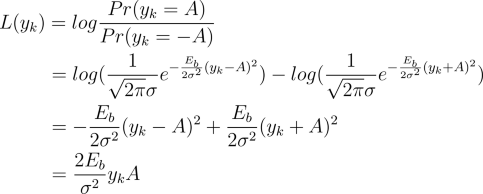
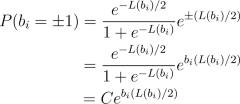
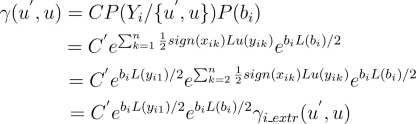
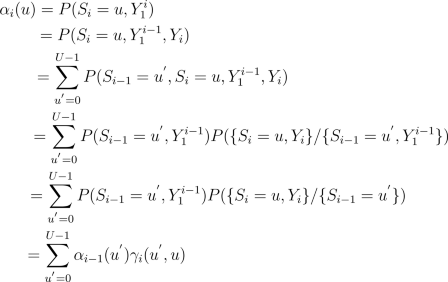
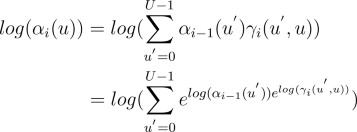
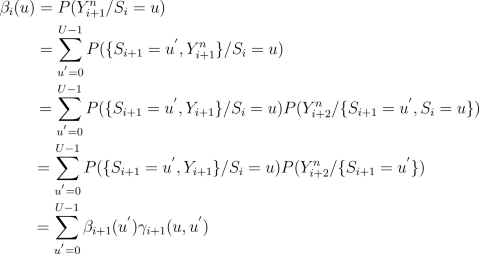
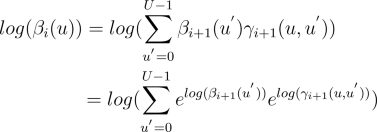
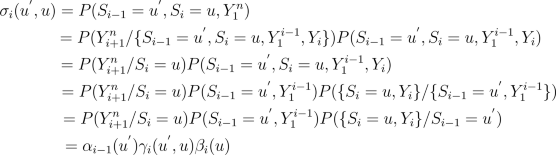
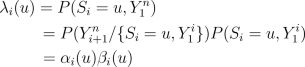
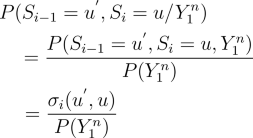
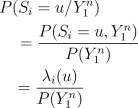
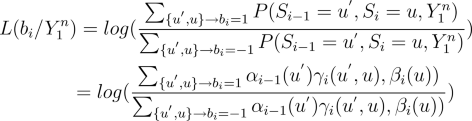
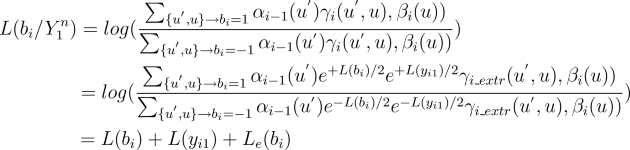
**BCJR算法--一个例子**

[程序下载](http://www.feiyilin.com/downloads/BCJR.zip)

**[参考文献]**

1. Jorge, C.M. and Patrick G.F., Essentials of Error-control Coding, John Wiley &; Sons, Ltd,2006
2. [Implementing a MAP Decoder for cdma2000 ETurbo Codes on a TMS320C62x E DSP Device](http://focus.ti.com/lit/an/spra629/spra629.pdf)
3. 公式绘图 <http://www.sitmo.com/latex/>
4. **1 概述**
5. 上面参考文献中关于BCJR算法的内容讲解的非常详细，这个例子希望能有助于你更好的理解其中的内容。
6. 卷积码
7. 
8. 卷积码状态转移图
9. 
10. Turbo码编码
11. 
12. 这个例子中采用的交织图案
13. 
14. 截短的矩阵为
15. P1 = [1 1;
16. 1 0]
17. P2 = [0 0;
18. 0 1]
19. 也就是对于第一路卷积码的两路输出:
20. 信息位：全部保留
21. 校验位：只保留序号为（0，2，4，...)
22. 对于第二路卷积码的两路输出：
23. 信息位：全部去掉，因为这些信息位可以用第一路的信息位经过一个交织器来得到
24. 校验位：只保留序号为（1，3，5，...)
25. Turbo码解码
26. 
27. 输入，信息位和校验位
28. U = [+0.213 -0.371 +0.139 +0.514 +0.539 -0.422 +1.533 +1.457 +0.323 +2.028 -0.414 +1.482 -1.701 -0.175 -0.862 -0.918];
29. C = [+0.364 0.000 +1.818 0.000 +0.388 0.000 +0.267 0.000 +1.103 0.000 +3.560 0.000 +0.893 0.000 -2.049 0.000];
30. C2= [ 0.000 -0.351 0.000 +1.646 0.000 -2.587 0.000 -1.678 0.000 -0.170 0.000 -1.003 0.000 -1.306 0.000 -0.492];
31. 计算LLR(log-likelihood ratio)
32. 认为信道为AWGN信道N(0,1.2)，即SNR = E/N0 = 1/(2\*1.2^2)
33. Lu = log((1/sqrt(2\*pi\*1.2)\* exp(-(U -1).^2/(2\*1.2\*1.2)))./(1/sqrt(2\*pi\*1.2)\* exp(-(U +1).^2/(2\*1.2\*1.2))));
34. Lc = log((1/sqrt(2\*pi\*1.2)\* exp(-(C -1).^2/(2\*1.2\*1.2)))./(1/sqrt(2\*pi\*1.2)\* exp(-(C +1).^2/(2\*1.2\*1.2))));
35. Lc2 = log((1/sqrt(2\*pi\*1.2)\* exp(-(C2-1).^2/(2\*1.2\*1.2)))./(1/sqrt(2\*pi\*1.2)\* exp(-(C2+1).^2/(2\*1.2\*1.2))));
36. 初始化外信息为0
37. Lu\_ex = zeros(1,length(U));
38. 循环迭代
39. for m =1:4
40. Lu1 = Lu + Lu\_ex;
41. Lu\_new = Turbo\_BCJR(Lu1,Lc,0);
42. Lu\_ex = Lu\_new-Lu1;
43. Lu2 = Lu + Lu\_ex;
44. Lu\_est = Lu2;
45. Lu2 = Turbo\_Interleaver(Lu2);
46. Lu\_new = Turbo\_BCJR(Lu2,Lc2,0);
47. Lu\_ex = Lu\_new-Lu2;
48. Lu\_ex = Turbo\_Interleaver(Lu\_ex);
49. Lu\_est = Lu\_est + Lu\_ex;
50. end
51. function [out] = Turbo\_Interleaver(in)
52. out = reshape(reshape(in,4,4)',1,16);
53. end
54. function [Lu\_new] = Turbo\_BCJR(Lu, Lc,end\_s)
55. FRAME\_LENGTH = length(Lu);
56. **2 gamma**
57. 计算分支度量（bench metric），就是分别计算每个时刻k输出的概率
58. 在这个例子中，就是计算输出为(0，0)(0，1)(1，0)(1，1)的LLR概率，这是因为：
59. gamma值的定义：
60. 
61. latex 代码：
62. \gamma(u^{'},u)=P(\{S\_i=u,Y\_i\}/S\_{i-1}=u^{'}) \\
63. = P(Y\_i/{u^{'},u})P(u/u^{'}) \hspace{12 mm} \\
64. =P(Y\_i/{u^{'},u})P(b\_i) \hspace{17 mm}
65. 对于AWGN信道，其中：
66. 
67. latex 代码：
68. P(Y\_i/\{u^{'},u\}) = P(Y\_i/X\_i) \hspace{60 mm} \\
69. =\prod\_{k=1}^{n} P(y\_ik/x\_ik) \hspace{51 mm} \\
70. =\prod\_{k=1}^{n} \frac{1} {\sqrt{2\pi} \sigma} e^{-\frac{E\_b} {2\sigma^2} (y\_{ik} - x\_{ik})^2} \hspace{35 mm} \\
71. =\frac{1} {(\sqrt{2\pi} \sigma)^n} e^{ -\frac{E\_b} {2\sigma^2} \sum\_{k=1}^{n} (y\_{ik} - x\_{ik})^2} \hspace{27 mm} \\
72. =\frac{1} {(\sqrt{2\pi} \sigma)^n} e^{ -\frac{E\_b} {2\sigma^2} \sum\_{k=1}^{n} (y\_{ik}^2 + x\_{ik}^2)} e^{\frac{E\_b} {\sigma^2} \sum\_{k=1}^{n}y\_{ik}x\_{ik}} \hspace{5 mm} \\
73. =C e^{\frac{E\_b} {\sigma^2} \sum\_{k=1}^{n} (y\_{ik}x\_{ik})} \hspace{47 mm}
74. 由于输入数据的LLR值为：
75. 
76. latex 代码：
77. L(y\_k)=log{ \frac{Pr(y\_k=A)} {Pr(y\_k=-A)}} \hspace{70mm} \\
78. =log( \frac{1} {\sqrt{2\pi} \sigma}e^{ -\frac{E\_b} {2\sigma^2}(y\_k-A)^2})-log( \frac{1} {\sqrt{2\pi} \sigma}e^{-\frac{E\_b} {2\sigma^2}(y\_k + A)^2}) \hspace{13 mm} \\
79. =-\frac{E\_b} {2\sigma^2}(y\_k-A)^2 + \frac{E\_b} {2\sigma^2}(y\_k + A)^2 \hspace{44 mm} \\
80. =\frac{2E\_b} {\sigma^2}y\_kA \hspace{86mm}
81. 所以：
82. http://www.feiyilin.com/images/BCJR_eq_3.png
83. latex 代码：
84. P(Y\_i/\{u^{'},u\})=Ce^{ \frac{E\_b} {\sigma^2} \sum\_{k=1}^{n}y\_{ik}x\_{ik}} \hspace{11 mm} \\
85. =Ce^{\sum\_{k=1}^{n} \frac{1} {2}sign(x\_{ik})L(y\_{ik})}
86. 所以
87. http://www.feiyilin.com/images/BCJR_eq_4.png
88. latex 代码：
89. \gamma(u^{'},u)=P(Y\_i/\{u^{'},u\})P(b\_i) \hspace{6 mm} \\
90. =Ce^{ \frac{1} {2}sign(x\_{ik})Lu(y\_{ik})}P(b\_i)
91. 对于这个例子而言，在每一个状态，输入{0，1}的概率相等都是1/2，所以上式中最后面一项为一个常数，所以可以不用带入计算：
92. http://www.feiyilin.com/images/BCJR_eq_5.png
93. latex 代码：
94. \gamma^{'}(u^{'},u)=P(Y\_i/\{u^{'}, u\}) \hspace{9 mm} \\
95. =e^{\sum\_{k=1}^{n} \frac{1} {2}sign(x\_{ik})L(y\_{ik})} \\
96. http://www.feiyilin.com/images/BCJR_eq_6.png
97. latex 代码：
98. log(\gamma^{'}(u^{'},u))=\sum\_{k=1}^{n} \frac{1} {2}sign(x\_{ik})L(y\_{ik})
99. 
100. latex 代码：
101. P(b\_i=\pm1)=\frac{e^{-L(b\_i)/2}} {1 + e^{-L(b\_i)}}e^{\pm(L(b\_i)/2)} \\
102. =\frac{e^{-L(b\_i)/2}} {1 + e^{-L(b\_i)}}e^{b\_i(L(b\_i)/2)} \\
103. =Ce^{b\_i(L(b\_i)/2)} \hspace{16 mm}
104. 
105. latex 代码：
107. \gamma(u^{'},u)=CP(Y\_i/\{u^{'}, u\})P(b\_i) \hspace{33 mm} \\
108. =C^{'}e^{\sum\_{k=1}^n \frac{1} {2}sign(x\_{ik})Lu(y\_{ik})}e^{b\_iL(b\_i)/2} \hspace{16 mm} \\
109. =C^{'}e^{b\_iL(y\_{i1})/2}e^{\sum\_{k=2}^n \frac{1} {2}sign(x\_{ik})Lu(y\_{ik})}e^{b\_iL(b\_i)/2} \\
110. =C^{'}e^{b\_iL(y\_{i1})/2}e^{b\_iL(b\_i)/2} \gamma\_{i\\_extr}(u^{'}, u) \hspace{14 mm}
111. 其中，第一项是常数，可以不用带入计算；第二项之所以把k=1单独提出来，是因为这一项正好是接收到的系统信息，区分与后面的外信息；第三项是可以认为是常数，因为认为在每个时刻输入的数据取值{1 0}的概率相等，可以不带入计算；最后一项是外信息，用来和另外一个BCJR译码器交换信息。所以有用的项只留下上面公式5中的gamma'
112. 对于这个例子而言，每一个转移路径上有两个输出（U，C），这两个输出的可能值都是{+1，-1}，从公式5可以，看出m00 = -m11 ，m01 = -m10
113. m11 = (Lu+Lc)/2;
114. m10 = (Lu-Lc)/2;
115. **3 alpha**
116. 计算前向状态度量(forawrd state metric):
117. 计算 i 时刻系统处于每一个状态 m 的概率，这个概率由前一个时刻 (i-1) 系统所处状态 u' 和路径转移概率 gamma 决定，
118. 
119. latex 代码：
120. \alpha\_i(u) =P(S\_i=u,Y\_1^i) \hspace{80 mm} \\
121. =P(S\_i=u,Y\_1^{i-1},Y\_i) \hspace{71 mm} \\
122. =\sum\_{u^{'}=0}^{U-1}P(S\_{i-1}=u^{'},S\_i=u, Y\_1^{i-1}, Y\_i) \hspace{46 mm} \\
123. =\sum\_{u^{'}=0}^{U-1}P(S\_{i-1}=u^{'}, Y\_1^{i-1})P(\{S\_i=u, Y\_i\}/\{S\_{i-1}=u^{'}, Y\_1^{i-1} \}) \hspace{5 mm} \\
124. =\sum\_{u^{'}=0}^{U-1}P(S\_{i-1}=u^{'}, Y\_1^{i-1})P(\{S\_i=u, Y\_i\}/\{S\_{i-1}=u^{'} \}) \hspace{16 mm} \\
125. =\sum\_{u^{'}=0}^{U-1} \alpha\_{i-1}(u^{'})\gamma\_i(u^{'}, u) \hspace{72 mm}
126. LLR:
127. 
128. latex 代码：
129. log(\alpha\_i(u)) =log(\sum\_{u^{'}=0}^{U-1} \alpha\_{i-1}(u^{'})\gamma\_i(u^{'}, u)) \hspace{8 mm} \\
130. =log(\sum\_{u^{'}=0}^{U-1} {e^{log(\alpha\_{i-1}(u^{'})) e^{log(\gamma\_i(u^{'}, u))}}})
131. 比如状态0有两条输入分支：0-->;0(00)，2-->;0(11)，那么在k时刻，系统处于0状态的概率为：
132. http://www.feiyilin.com/images/BCJR_eq_alpha_2.png
133. latex 代码：
134. \alpha\_i(0)=\alpha\_{i-1}(0)\gamma\_i(0,0) + \alpha\_{i-1}(2)\gamma\_i(1,1)
135. 即:
136. http://www.feiyilin.com/images/BCJR_eq_alpha_3.png
137. latex 代码：
138. log(\alpha\_i(0))=log(\alpha\_{i-1}(0)\gamma\_i(0,0) + \alpha\_{i-1}(2)\gamma\_i(1,1)) \hspace{20 mm} \\
139. =log(e^{log(\alpha\_{i-1}(0))}e^{log(\gamma\_i(0,0))} + e^{log(\alpha\_{i-1}(2))}e^{log(\gamma\_i(1,1))})
140. alpha = zeros(1,(FRAME\_LENGTH+2)\*4);
141. alpha(1:4) = -100\*ones(1,4);
142. alpha(1) = 0;
143. for k=1:FRAME\_LENGTH
144. m\_t = alpha((k-1)\*4+0+1) - m11(k-1+1);
145. m\_b = alpha((k-1)\*4+2+1) + m11(k-1+1);
146. alpha(k\*4+0+1) = log(exp(m\_t)+exp(m\_b));
147. m\_t = alpha((k-1)\*4+0+1) + m11(k-1+1);
148. m\_b = alpha((k-1)\*4+2+1) - m11(k-1+1);
149. alpha(k\*4+1+1) = log(exp(m\_t)+exp(m\_b));
150. m\_t = alpha((k-1)\*4+1+1) - m10(k-1+1);
151. m\_b = alpha((k-1)\*4+3+1) + m10(k-1+1);
152. alpha(k\*4+2+1) = log(exp(m\_t)+exp(m\_b));
153. m\_t = alpha((k-1)\*4+1+1) + m10(k-1+1);
154. m\_b = alpha((k-1)\*4+3+1) - m10(k-1+1);
155. alpha(k\*4+3+1) = log(exp(m\_t)+exp(m\_b));
156. end
157. **4 belta**
158. 计算后向状态度量（backward state metris）
159. 计算i时刻系统处于状态u的条件下，接收到Y(i+1,n)的概率，这个概率由i+1时刻系统所处状态beta和路径转移概率gamma(u',u)决定，
160. 
161. latex 代码:
162. \beta\_i(u)=P(Y\_{i+1}^n/S\_i=u) \hspace{80 mm} \\
163. =\sum\_{u^{'}=0}^{U-1}P(\{S\_{i+1}=u^{'},Y\_{i+1}^n\}/S\_i=u) \hspace{50 mm} \\
164. =\sum\_{u^{'}=0}^{U-1}P(\{S\_{i+1}=u^{'},Y\_{i+1} \}/S\_i=u)P(Y\_{i+2}^{n}/\{S\_{i+1}=u^{'},S\_i=u\})\\
165. =\sum\_{u^{'}=0}^{U-1}P(\{S\_{i+1}=u^{'},Y\_{i+1} \}/S\_i=u)P(Y\_{i+2}^{n}/\{S\_{i+1}=u^{'} \}) \hspace{14 mm} \\
166. =\sum\_{u^{'}=0}^{U-1} \beta\_{i+1}(u^{'})\gamma\_{i+1}(u,u^{'}) \hspace{69 mm}
167. LLR:
168. 
169. latex 代码:
170. log(\beta\_i(u))=log(\sum\_{u^{'}=0}^{U-1} \beta\_{i+1}(u^{'})\gamma\_{i+1}(u, u^{'})) \hspace{9 mm} \\
171. =log(\sum\_{u^{'}=0}^{U-1} e^{log(\beta\_{i+1}(u^{'}))} e^{log(\gamma\_{i+1}(u, u^{'}))})\\
172. 比如在i时刻状态0有两条从i+1时刻输入分支：0-->;0(00)，1-->;0(11)，那么在i时刻，beta值为：
173. http://www.feiyilin.com/images/BCJR_eq_beta_2.png
174. latex 代码：
175. \beta\_i(0)=\beta\_{i+1}(0) \gamma\_{i+1}(0,0) + \beta\_{i+1}(1) \gamma\_{i+1}(1,1)
176. 即:
177. http://www.feiyilin.com/images/BCJR_eq_beta_3.png
178. latex 代码：
179. log(\beta\_i(0))=log(\beta\_{i+1}(0) \gamma\_{i+1}(0,0) + \beta\_{i+1}(1)\gamma\_{i+1}(1,1)) \hspace{19 mm} \\
180. =log(e^{log(\beta\_{i+1}(0))}e^{log( \gamma\_{i+1}(0,0))} + e^{log(\beta\_{i+1}(1))} e^{log( \gamma\_{i+1}(1,1))})
181. beta = zeros(1,(FRAME\_LENGTH+2)\*4);
182. beta((FRAME\_LENGTH)\*4+0+1:(FRAME\_LENGTH)\*4+0+4) = -100\*ones(1,4);
183. beta((FRAME\_LENGTH)\*4+end\_s+1) = 0;
184. for k=FRAME\_LENGTH-1:-1:0
185. m\_t = beta((k+1)\*4+0+1) - m11(k+1);
186. m\_b = beta((k+1)\*4+1+1) + m11(k+1);
187. beta(k\*4+0+1) = log(exp(m\_t)+exp(m\_b));
188. m\_t = beta((k+1)\*4+2+1) - m10(k+1);
189. m\_b = beta((k+1)\*4+3+1) + m10(k+1);
190. beta(k\*4+1+1) = log(exp(m\_t)+exp(m\_b));
191. m\_t = beta((k+1)\*4+0+1) + m11(k+1);
192. m\_b = beta((k+1)\*4+1+1) - m11(k+1);
193. beta(k\*4+2+1) = log(exp(m\_t)+exp(m\_b));
194. m\_t = beta((k+1)\*4+2+1) + m10(k+1);
195. m\_b = beta((k+1)\*4+3+1) - m10(k+1);
196. beta(k\*4+3+1) = log(exp(m\_t)+exp(m\_b));
197. end
198. **5 sigma**
199. 计算在输入Y(1:n)的情况下，状态i-1为u'，状态i为u的联合概率
200. 
201. latex 代码：
202. \sigma\_i(u^{'},u)=P(S\_{i-1}=u^{'}, S\_i=u,Y\_1^n) \hspace{78 mm} \\
203. =P(Y\_{i+1}^n/\{S\_{i-1}=u^{'}, S\_i=u,Y\_1^{i-1},Y\_i\}) P(S\_{i-1}=u^{'}, S\_i=u, Y\_1^{i-1},Y\_i) \hspace{4 mm} \\
204. =P(Y\_{i+1}^n/S\_i=u) P(S\_{i-1}=u^{'}, S\_i=u, Y\_1^{i-1},Y\_i) \hspace{42 mm} \\
205. =P(Y\_{i+1}^n/S\_i=u) P(S\_{i-1}=u^{'}, Y\_1^{i-1}) P(\{S\_i=u,Y\_i\}/\{S\_{i-1}=u^{'}, Y\_1^{i-1} \})\\
206. =P(Y\_{i+1}^n/S\_i=u) P(S\_{i-1}=u^{'}, Y\_1^{i-1}) P(\{S\_i=u,Y\_i\}/S\_{i-1}=u^{'}) \hspace{14 mm} \\
207. =\alpha\_{i-1}(u^{'}) \gamma\_i(u^{'}, u) \beta\_i(u) \hspace{83mm}
208. LLR:
209. http://www.feiyilin.com/images/BCJR_eq_sigma_2.png
210. latex 代码：
211. log(\sigma\_i(u^{'},u))=log(\alpha\_{i-1}(u^{'})\gamma\_i(u^{'}, u)\beta\_i(u)) \hspace{0 mm} \\
212. =log(e^{log(\alpha\_{i-1}(u^{'}))}e^{log(\gamma\_i(u^{'}, u))}e^{log(\beta\_i(u))}) \hspace{0 mm} \\
213. **6 lambda**
214. 计算在输入Y(1:n)的情况下，状态i为u的联合概率
215. 
216. latex 代码：
217. \lambda\_i(u)=P(S\_i=u, Y\_1^n) \hspace{38 mm} \\
218. =P(Y\_{i+1}^n/\{S\_i=u, Y\_1^{i} \}) P(S\_i=u, Y\_1^{i})\\
219. =\alpha\_{i}(u) \beta\_i(u) \hspace{44 mm}
220. LLR:
221. http://www.feiyilin.com/images/BCJR_eq_lambda_2.png
222. latex 代码：
223. log(\lambda\_i(u))=log(\alpha\_{i}(u)\beta\_i(u)) \hspace{12 mm} \\
224. =log(e^{log(\alpha\_{i}(u))}e^{log(\beta\_i(u))})
225. **7 计算MAP概率**
226. 上面所有的概率计算都是为了得到MAP概率计算的公式
227. 
228. latex 代码：
229. P(S\_{i-1}=u^{'},S\_i=u/Y\_1^n) \hspace{0 mm} \hspace{10 mm} \\
230. =\frac{P(S\_{i-1}=u^{'}, S\_i=u, Y\_1^n) \hspace{0 mm}} {P(Y\_1^n)} \\
231. =\frac{\sigma\_i(u^{'}, u)} {P(Y\_1^n)} \hspace{29 mm}
232. 
233. latex 代码：
234. P(S\_i=u/Y\_1^n) \hspace{10 mm} \\
235. =\frac{P(S\_i=u, Y\_1^n)} {P(Y\_1^n)} \\
236. =\frac{\lambda\_i(u)} {P(Y\_1^n)} \hspace{14 mm}
237. 上面两个式子中，对于某组Y(1:n)，分母部分都是一个常数，所以可以去掉而不会影响结果，所以可以用sigma和lambda来表示而不会影响最后的结果。
238. 计算在时刻i，输入数据b(i)=+1/-1的概率
239. http://www.feiyilin.com/images/BCJR_eq_deci_0.png
240. latex 代码：
241. P(b\_i=A/Y\_1^n)=\sum\_{\{u^{'}, u\} \rightarrow b\_i=A}P(S\_{i-
242. 1}=u^{'},S\_i=u,Y\_1^n)
243. LLR:
244. 
245. latex 代码：
246. L(b\_i/Y\_1^n)=log( \frac{\sum\_{\{u^{'}, u\} \rightarrow b\_i=1} P(S\_{i-1}=u^{'}, S\_i=u, Y\_1^n)} {\sum\_{\{u^{'}, u\} \rightarrow b\_i=-1} P(S\_{i-1}=u^{'}, S\_i=u,Y\_1^n)})\\
247. =log( \frac{\sum\_{\{u^{'}, u\} \rightarrow b\_i=1} \alpha\_{i-1}(u^{'})\gamma\_i(u^{'}, u), \beta\_i(u))} { \sum\_{ \{u^{'}, u\} \rightarrow b\_i=-1} \alpha\_{i-1}(u^{'}) \gamma\_i(u^{'}, u), \beta\_i(u))}) \hspace{4 mm}
248. 代入上面的gamma alpha beta 可以得到：
249. 
250. latex 代码：
251. L(b\_i/Y\_1^n)=log( \frac{\sum\_{\{u^{'}, u\} \rightarrow b\_i=1} \alpha\_{i-1}(u^{'})\gamma\_i(u^{'}, u), \beta\_i(u))} {\sum\_{\{u^{'}, u\} \rightarrow b\_i=-1} \alpha\_{i-1}(u^{'})\gamma\_i(u^{'}, u), \beta\_i(u))}) \hspace{40 mm} \\
252. =log( \frac{\sum\_{\{u^{'}, u\} \rightarrow b\_i=1} \alpha\_{i-1}(u^{'}) e^{+L(b\_i)/2}e^{+L(y\_{i1})/2} \gamma\_{i\\_extr}(u^{'}, u), \beta\_i(u))} {\sum\_{\{u^{'}, u\} \rightarrow b\_i=-1} \alpha\_{i-1}(u^{'}) e^{-L(b\_i)/2}e^{-L(y\_{i1})/2} \gamma\_{i\\_extr}(u^{'}, u), \beta\_i(u))}) \hspace{4 mm} \\
253. =L(b\_i) + L(y\_{i1}) + L\_e(b\_i) \hspace{76 mm}
254. 其中第一项在输入{1，-1}概率均等的情况下为0，第二项是因为我们假设系统传输的是系统码，也就是每一个状态输出的第一个bit为原信息，最后一项为外信息LLR，是译码器之间用来交互的软信息。
255. ext = zeros(1,length(FRAME\_LENGTH));
256. for k = 1:FRAME\_LENGTH
257. enumerator = 0;
258. denominator = 0;
259. t\_d = alpha((k-1)\*4+0+1) + beta(k\*4+0+1) - m11(k-1+1);
260. denominator = denominator + exp(t\_d);
261. t\_e = alpha((k-1)\*4+0+1) + beta(k\*4+1+1) + m11(k-1+1);
262. enumerator = enumerator + exp(t\_e);
263. t\_d = alpha((k-1)\*4+1+1) + beta(k\*4+2+1) - m10(k-1+1);
264. denominator = denominator + exp(t\_d);
265. t\_e = alpha((k-1)\*4+1+1) + beta(k\*4+3+1) + m10(k-1+1);
266. enumerator = enumerator + exp(t\_e);
267. t\_d = alpha((k-1)\*4+2+1) + beta(k\*4+1+1) - m11(k-1+1);
268. denominator = denominator + exp(t\_d);
269. t\_e = alpha((k-1)\*4+2+1) + beta(k\*4+0+1) + m11(k-1+1);
270. enumerator = enumerator + exp(t\_e);
271. t\_d = alpha((k-1)\*4+3+1) + beta(k\*4+3+1) - m10(k-1+1);
272. denominator = denominator + exp(t\_d);
273. t\_e = alpha((k-1)\*4+3+1) + beta(k\*4+2+1) + m10(k-1+1);
274. enumerator = enumerator + exp(t\_e);
275. ext(k) = log(enumerator/denominator);
276. end
277. Lu\_new = ext;
278. end
279. **8 一些实现上的考虑**
280. 上面的LLR公式中涉及到很多log，exp的操作，在实现中这些操作可以进一步简化
281. http://www.feiyilin.com/images/BCJR_eq_other_0.png
282. latex 代码：
283. log(e^{x}+e^{y})=max(x,y)+log(1+e^{-|x-y|})
284. 在x，y相差较大的情况下，后面一项可以忽略不计；在相差较小的情况下可以用一个look-up-table来查询得到，从而避免复杂的对数指数计算。