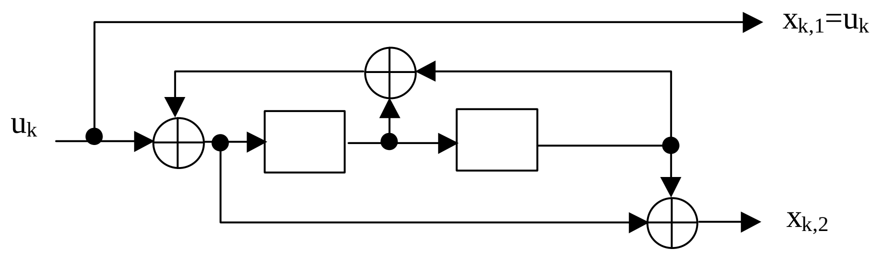
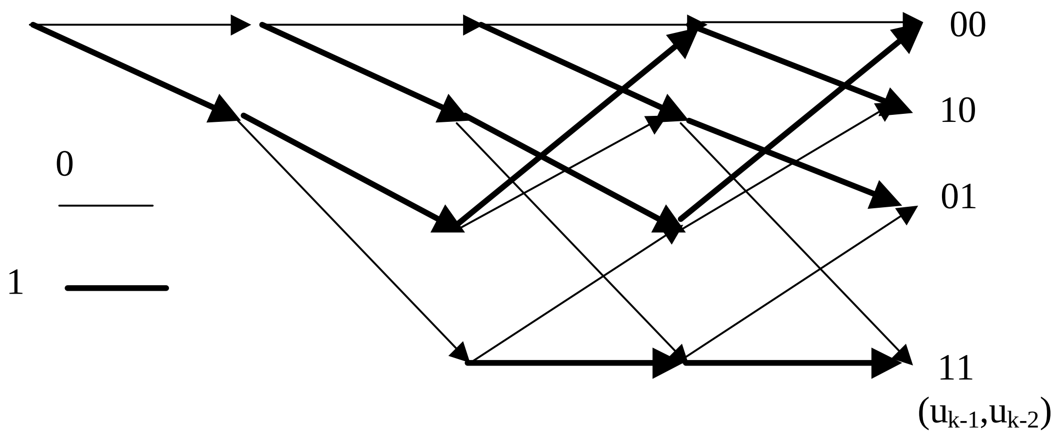
B. Analysis of Log Likelihood Ratio (LLR) of Systematic Convolutional Codes
(in AWGN, Rate ($1/n_0$) **Feed Back or Recursive Systematic - RSC** type)



Trellis Diagram:



The definition for states would be not very much useful here, as both information bits (+1, -1) would reach the same state. Therefore by specifying state only the information bit cannot be uniquely identified. We can therefore use the summation of branch transitions (corresponding to i_t =+1 and i_t=-1 respectively) to get the APP of information bits. This is equivalent to using the summation of σ's.

Thus if we define (as before):

Let $B_t^{(j)}$ be the set of transitions $S_{t-1}=m'$ to $S_t = m$ such that the jth input digit $i_t^{(j)}$ on that transition is 1. $B_t^{(j)}$ is independent of t for time invariant codes. Then,

$$\Pr\{ i_t^{(j)} = 1 \,;\, Y_1^{\tau} \} = \sum_{(m',m)\in B_t^{(j)}} s_t(m',m)$$

Which can be normalized to give $\Pr\{ i_t^{(j)} = 1 \,|\, Y_1^{\tau}\}$.

$$\sigma_t(m',m) = \alpha_{t-1}(m') \,.\, \gamma_t(m',m) \,.\, \beta_t(m)$$

Therefore,

$$\text{LLR} = Log \frac{\Pr\{i_t = 1 | Y_1^t\}}{\Pr\{i_t = -1 | Y_1^t\}} = Log \frac{\sum_{(m'',m) i_t = 1} s_t(m',m)}{\sum_{(\hat{m},m') t_t = -1} s_t(\hat{m},m')}$$

$$= \text{Log} \frac{\sum_{(m'',m)\, i_t = 1} a_{t-1}(m'').\, g_t(m'',m).b_t(m)}{\sum_{(\hat{m},m')\, i_t = -1} a_{t-1}(\hat{m}).\, g_t(\hat{m},m').b_t(m')}$$

The above breaks down as before with the additional term of the *a priori* probabilities provided we don't assume them to be equally likely,

$$\text{LLR} = \text{Log} \frac{p(y_t^{(1)} \,|\, x_t^{(1)} = 1)}{p(y_t^{(1)} \,|\, x_t^{(1)} = -1)} \; + \; \text{Log} \frac{\Pr\{i_t = 1)}{\Pr\{i_t = -1)} \; +$$

$$\text{Log}\ \frac{\sum_{(m'',m)\ i_t=1} a_{t-1}(m'').\prod_{j=2}^{n_0} p(y_t^{(j)} \mid x_t^{(j)}).b_t(m)}{\sum_{(\hat{m},m')\ i_t=-1} a_{t-1}(\hat{m}).\prod_{k=2}^{n_0} p(y_t^{(k)} \mid x_t^{(k)}).b_t(m')}$$

where $$a_t(m) = \sum_{m'=0}^{M-1} a_{t-1}(m').\,g_t(m',m)$$

$$b_t(m) = \sum_{m'=0}^{M-1} b_{t+1}(m').\,g_{t+1}(m,m')$$

- ***Note***

In our earlier equations we have used $p_t(m' \mid m) = ½$ due to the equally likely input probabilities. Now we denote this by $p_I(i_t)$, (in the sense $p(i_t = -1)$ or $p(i_t = 1)$ associated) with $\gamma_t(m',m)$.

Define $L_I(i) = Log\,\frac{p_I(i_t = +1)}{p_I(i_t = -1)}$

Thus we'll get our equations for α,β modified by $p_I(i_t = -1)$, $p_I(i_t = 1)$ and LLR by $L_I(i)$.

This $L_I(i)$ is usually referred to as *a priori* information.

Using log-likelihoods, the *a priori* probability $p_I(i_t)$ can be expressed as ,i.e., using

$L_I(I_t)$,

$$p_I(i_t=\pm 1) = \frac{e^{\pm L_I(i_t)}}{1+e^{\pm L_I(i_t)}} = \left(\frac{e^{-L_I(i_t)/2}}{1+e^{-L_I(i_t)}}\right) e^{L_I(i_t).i_t/2} = A_t.e^{L_I(i_t).i_t/2}$$

Thus

$\text{Log } p_I(i_t=\pm 1) = \text{Log } A_t + L_I(i_t).i_t/2$

The terms A_t are equal for all transition from (m′,m) at time t and hence will cancel in the LLR ratio. We can therefore neglect the A_t term.

Assuming a channel with

$$y_t = a.x_t + n \quad \text{where n is AWGN and a is a constant.}$$

$$\gamma_t(m',m) = p(\mathbf{y}_t|\, i_t).\, p(i_t)$$

The conditional probability for a convolutional code can be written as,

$$p(\mathbf{y}_t|\, i_t) = B_t.\exp\left(\frac{1}{2}L_c y_{(t)}^1 x_{(t)}^1 + \frac{1}{2}\sum_{j=2}^{n_0} L_c y_{(t)}^j x_{(t)}^j\right)$$

L_c is a channel dependent constant. The summation in the above equation is only over non-punctured coded bits.

A_t and B_t are equal for all transitions from t to t+1.

Hence effectively the branch transition $\gamma_t(m',m)$ reduces to the expression,

$$\exp\left(\frac{1}{2}i_t.L_I(i_t) + \frac{1}{2}x_{(t)}^1.L_c y_{(t)}^1\right)\gamma_t^{(e)}(m',m)$$

$x_{(t)}^1 = i_t$ for systematic codes and

$$\gamma_t^{(e)}(m',m) = \exp\left(\frac{1}{2}\sum_{j=2}^{n_0} L_c y_{(t)}^j x_{(t)}^j\right) \text{ with respect to the systematic bit.}$$

OR $\gamma_t \; (m', m)$ effectively,

$$\mathrm{Exp}\left(\frac{1}{2} i_{(t)} . L_I(i_t) + \frac{1}{2} x_{(t)}^1 . L_c y_{(t)}^1 + \frac{1}{2} \sum_{j=2}^{n_0} L_c y_{(t)}^j x_{(t)}^j \right)$$

We define

$$\mathrm{L}(x_{(t)}^j) = \mathrm{Log}\ \frac{p(y_t^j \mid x_t^j = 1)}{p(y_t^j \mid x_t^j = -1)} = L_c y_t^j$$

The above equation for $\gamma_t(m',m)$ can be written as

$$\mathrm{Exp}\left(\frac{1}{2} i_{(t)} . L_I(i_t) + \frac{1}{2} x_{(t)}^1 . \mathrm{L}(x_{(\mathrm{t})}^1) + \frac{1}{2} \sum_{j=2}^{n_0} x_{(t)}^j . L(x_{(t)}^j) \right)$$

$$\mathrm{LLR} = \mathrm{L_c}\, y_{(t)}^1 + \mathrm{L_I(i_t)} + \mathrm{Log}\ \frac{\sum_{(m',m) i_t = 1} \gamma_t^{(e)}(m', m) \alpha_{t-1}(m') . \beta_t(m)}{\sum_{(m',m) i_t = -1} \gamma_t^{(e)}(m', m) \alpha_{t-1}(m') . \beta_t(m)}$$

The extrinsic information can be denoted in a similar manner by $L_e(i)$ and the systematic information by $L_{sys}(i)$.

Therefore,

LLR or $L_{app}(i_t) = L_{sys}(i_t) + L_I(i_t) + L_e(i_t)$

C. Analysis of Log Likelihood Ratio (LLR) of *Coded bits* in Convolutional Codes

The objective is to determine the APP of the encoder output digits,

i.e., Pr{ $x_t^{(j)}=1 \mid Y_1^{\tau}$}.

As defined earlier, let $B_t^{(j)}$ be the set of transitions $S_{t-1}=m'$ to $S_t = m$ such that the j[th] output digit $x_t^{(j)}$ on that transition is 1. $B_t^{(j)}$ is independent of t for time invariant codes. Then,

$$\Pr\{ x_t^{(j)}=1\,;\, Y_1^{\tau} \} = \sum_{(m',m)\in B_t^{(j)}} s_t(m',m)$$

Which can be normalized to give Pr{ $x_t^{(j)}=1 \mid Y_1^{\tau}$}.

What we did was to use the summation of branch transitions corresponding to $x_t^{(j)}=1$,(or $x_t^{(j)}=-1$ respectively) to get the APP of *coded bits*.

$$\sigma_t(m',m) = \alpha_{t-1}(m') \,.\, \gamma_t(m',m) \,.\, \beta_t(m)$$

Therefore,

$$\text{LLR}= Log\,\frac{\Pr\{x_t^{(j)}=1 \mid Y_1^t\}}{\Pr\{x_t^{(j)}=-1 \mid Y_1^t\}} = Log\,\frac{\sum_{(m'',m)x_t^{(j)}=1} s_t(m',m)}{\sum_{(\hat{m},m')x_j^{(j)}=-1} s_t(\hat{m},m')}$$

$$= \text{Log}\ \frac{\displaystyle\sum_{(m'',m)\ x_t^{(j)}=1} a_{t-1}(m'').\, g_t(m'',m).b_t(m)}{\displaystyle\sum_{(\hat{m},m')\ x_t^{(j)}=-1} a_{t-1}(\hat{m}).\, g_t(\hat{m},m').b_t(m')}$$

$$= \text{Log}\,\frac{p(y_t^{(j)} \mid x_t^{(j)}=1)}{p(y_t^{(j)} \mid x_t^{(j)}=-1)} \quad +$$

$$\text{Log}\ \frac{\displaystyle\sum_{(m'',m)\ x_t^{(j)}=1} a_{t-1}(m'').\prod_{k=1,k\neq j}^{n_0} p(y_t^{(k)} \mid x_t^{(k)})\, p(i_t(m'',m)).b_t(m)}{\displaystyle\sum_{(\hat{m},m')\ x_t^{(j)}=-1} a_{t-1}(\hat{m}).\prod_{l=1,l\neq j}^{n_0} p(y_t^{(l)} \mid x_t^{(l)})\, p(i_t(\hat{m},m').b_t(m')}$$

$$= \text{L}(x_{(t)}^{j}) + \text{Log}\ \frac{\displaystyle\sum_{(m',m)\ x_t^{(j)}=1} g_t^{(e)}(m',m)a_t(m').b_t(m)}{\displaystyle\sum_{(m',m)\ x_t^{(j)}=-1} g_t^{(e)}(m',m)a_t(m').b_t(m)}$$

$$= L(x^{j}_{(t)}) + \text{Log}\frac{\displaystyle\sum_{(m',m)\; x_t^{(j)}=1} a_{t-1}(m').\exp\left(\frac{1}{2} i_t L_I(i_t) + \frac{1}{2}\sum_{k=1,k\neq j}^{n_0} x^{j}_{(t)} L(x^{j}_{(t)})\right).b_t(m)}{\displaystyle\sum_{(m',m)\; x_t^{(j)}=-1} a_{t-1}(m').\exp\left(\frac{1}{2} i_t L_I(i_t) + \frac{1}{2}\sum_{k=1,k\neq j}^{n_0} x^{j}_{(t)} L(x^{j}_{(t)})\right).b_t(m)}$$

- **<u>Simplification in computations</u>**

We can avoid calculating actual probabilities by using the logarithm of probabilities and the approximation,

Log($e^{L1} + e^{L2}$) ≈ max (L1,L2). Then this algorithm works with log α, log β and log γ and the summations in the earlier equations can be replaced by the corresponding maximizations.

This is referred to as the "**Max-Log-MAP**" algorithm.