9.4.3 BCJR-Algorithmus am Beispiel von Faltungscodes

Der BCJR-Algorithmus wurde erstmals im Jahr 1974 von Bahl, Cocke, Jelinek und Raviv vorgestellt. Er stellt eine allgemeine Vorschrift zur *Symbol-by-Symbol*-MAP-Decodierung dar, die nicht nur zur Decodierung, sondern auch zur Entzerrung gedächtnisbehafteter Kanäle eingesetzt werden kann [BCJR74, LVS95, DJB95, Jor00]. Der Vorteil des BCJR-Algorithmus gegenüber der direkten Realisierung von (9.4.15) besteht in der effizienten Ausnutzung der Markov-Eigenschaft des Kanals bzw. des Codierers. Er setzt analog zum Viterbi-Algorithmus die Darstellung des Systems (in unserem Fall des Kanalcodes) durch ein Trellisdiagramm voraus. Wie schon in Kapitel 8 beschränken wir uns auf binäre, d.h. $1/n$-ratige Faltungscodes.

Ausgangspunkt zur *Symbol-by-Symbol*-MAP-Decodierung ist wiederum (9.4.11). Dabei ist hinsichtlich der Nomenklatur zu beachten, daß bei Faltungscodes mit einem Informationsbit $u(\ell)$ ein Codewort $\mathbf{c}(\ell) = (c_0(\ell) \cdots c_{n-1}(\ell))$ korrespondiert, so daß wir im Gegensatz zum letzten Abschnitt zu jedem Zeitpunkt ℓ Empfangsvektoren $\mathbf{y}(\ell) = (y_0(\ell) \cdots y_{n-1}(\ell))$ betrachten.

Vom Zeitpunkt $\ell - 1$ zum Zeitpunkt ℓ findet durch das Informationsbit $u(\ell)$ ein Übergang von Zustand S_ν zu Zustand S_μ statt. Alle möglichen Zustandsübergänge $S_\nu \rightarrow S_\mu$ lassen sich in zwei Klassen aufteilen, in solche, die mit $u(\ell) = 0$ korrespondieren und die übrigen, die mit $u(\ell) = 1$ verknüpft sind. Bezeichnen wir ein Zustandspaar mit (S_ν, S_μ) und zerlegen wir ferner den Empfangsvektor $\mathbf{y}(\ell)$ in drei Anteile, einen Anteil $\mathbf{y}(k < \ell)$, der alle empfangenen Symbole vor dem betrachteten Zeitpunkt ℓ enthält, $\mathbf{y}(\ell)$ selbst und einen Anteil $\mathbf{y}(k > \ell)$, der alle nach dem Zeitpunkt ℓ empfangenen Symbole beinhaltet, so können wir (9.4.15) in

$$L(\hat{u}(\ell)) = \ln \frac{\sum\limits_{\substack{(S_\nu, S_\mu) \\ u(\ell)=0}} p\big(S_\nu, S_\mu, \mathbf{y}\big)}{\sum\limits_{\substack{(S_\nu, S_\mu) \\ u(\ell)=1}} p\big(S_\nu, S_\mu, \mathbf{y}\big)} \tag{9.4.19}$$

$$= \ln \frac{\sum\limits_{\substack{(S_\nu, S_\mu) \\ u(\ell)=0}} p\big(S_\nu, S_\mu, \mathbf{y}(\mathbf{k} < \ell), \mathbf{y}(\ell), \mathbf{y}(k > \ell)\big)}{\sum\limits_{\substack{(S_\nu, S_\mu) \\ u(\ell)=1}} p\big(S_\nu, S_\mu, \mathbf{y}(k < \ell), \mathbf{y}(\ell), \mathbf{y}(k > \ell)\big)} \tag{9.4.20}$$

umformen. Sollen LLR's für die Codebit $c_i(\ell)$ berechnet werden, ist in den Summen von Zähler und Nenner $u(\ell)$ durch $c_i(\ell)$ zu substituieren. Die Wahrscheinlichkeitsdichtefunktion in Zähler und Nenner läßt sich mit der Kettenregel für bedingte Wahrscheinlichkeiten folgendermaßen zerlegen:

$$\begin{aligned} p\big(S_\nu, S_\mu, \mathbf{y}(k < \ell), \mathbf{y}(\ell), \mathbf{y}(k > \ell)\big) &= p\big(\mathbf{y}(k > \ell) \,\big|\, S_\nu, S_\mu, \mathbf{y}(k < \ell), \mathbf{y}(\ell)\big) \\ &\cdot p\big(S_\mu, \mathbf{y}(\ell) \,\big|\, S_\nu, \mathbf{y}(k < \ell)\big) \cdot p\big(S_\nu, \mathbf{y}(k < \ell)\big) \; . \end{aligned} \tag{9.4.21}$$