A2.11: RS–Decodierung nach „Erasures”

Potenzen von α	Polynome in α	Vektoren $k_2\,k_1\,k_0$
$\alpha^{-\infty}=0$	0	000
$\alpha^0=1$	1	001
α^1	α	010
α^2	α^2	100
α^3	$\alpha+1$	011
α^4	$\alpha^2+\alpha$	110
α^5	$\alpha^2+\alpha+1$	111
α^6	$\alpha^2\;\;+1$	101

Wir betrachten hier ein Codier– und Decodiersystem entsprechend der **Grafik** im Theorieteil zu diesem Kapitel. Anzumerken ist:

- Der Reed–Solomon–Code ist durch die Generatormatrix $\mathbf{G}$ und die Prüfmatrix $\mathbf{H}$ vorgegeben, wobei alle Elemente aus dem Galoisfeld GF(2^3) \ {0} stammen:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^1 & \alpha^3 & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha^1 & \alpha^4 \end{pmatrix},$$

$$\mathbf{H} = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^1 & \alpha^3 & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha^1 & \alpha^4 \end{pmatrix}.$$

- Alle Codesymbole $c_i \in \{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$ werden durch m = 3 Bit dargestellt und über den grün hinterlegten Auslöschungskanal (m–BEC) übertragen. Ein Codesymbol wird bereits dann als Auslöschung (*Erasure*) E markiert, wenn eines der drei zugehörigen Bit unsicher ist.
- Der *Codewortfinder* (CWF) hat die Aufgabe, aus dem teilweise ausgelöschten Empfangswort $\underline{y}$ das regenerierte Codewort $\underline{z}$ zu erzeugen. Dabei muss sicher gestellt sein, dass das Ergebnis $\underline{z}$ tatsächlich ein gültiges Reed–Solomon–Codewort ist.
- Beinhaltet das Empfangswort $\underline{y}$ zu viele Auslöschungen, so gibt der Decoder eine Meldung der Art „Symbol ist nicht decodierbar” aus. Es wird also nicht versucht, das Codewort zu schätzen. Wird $\underline{z}$ ausgegeben, so ist dieses auch richtig: $\underline{z} = \underline{c}$.
- Das gesuchte Informationswort $\underline{v} = \underline{u}$ ergibt sich durch die inverse Coderfunktion $\underline{v}$ = enc^{-1}($\underline{z}$). Mit der Generatormatrix $\mathbf{G}$ lässt sich diese wie folgt realisieren:

$$\underline{c} = \mathrm{enc}(\underline{u}) = \underline{u}\cdot\mathbf{G} \quad\Rightarrow\quad \underline{z} = \mathrm{enc}(\underline{v}) = \underline{v}\cdot\mathbf{G}$$
$$\Rightarrow\quad \underline{v} = \mathrm{enc}^{-1}(\underline{z}) = \underline{z}\cdot\mathbf{G}^{\mathrm{T}}.$$

Hinweis: Die Aufgabe bezieht sich auf das **Kapitel 2.4.** Hinsichtlich des *Codewortfinders* verweisen wir insbesondere auf die Seiten **Vorgehensweise** und **Lösung der Matrixgleichungen.**

Alle Berechnungen sind in GF(2^3) durchzuführen. Die obere Grafik beschreibt deren q = 8 Elemente in Potenz–, Polynom– und Koeffizientenvektordarstellung.

Fragebogen zu "A2.11: RS–Decodierung nach „Erasures”"

a) Geben Sie die Codeparameter des vorliegenden Reed–Solomon–Codes an.

$\boldsymbol{n}$ =

$\boldsymbol{k}$ =

$\boldsymbol{d}_{\text{min}}$ =

b) Kann der Empfangsvektor $\underline{y}$ = (0, 0, 0, 0, 0, 0, E) decodiert werden?

- JA.
- NEIN.

c) Kann der Empfangsvektor $\underline{y}$ = (E, E, 1, 1, 1, 1, 1) decodiert werden?

- JA.
- NEIN.

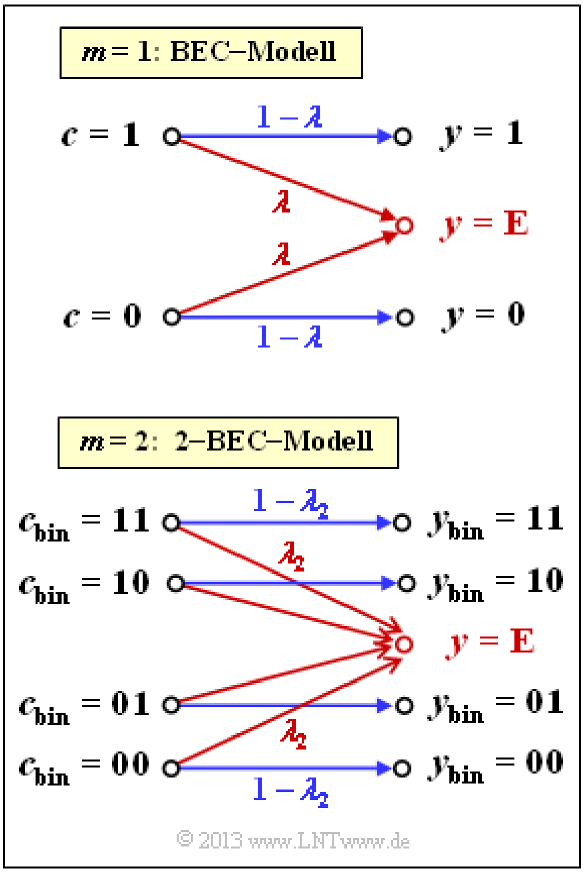
d) Welches Ergebnis liefert die Decodierung von $\underline{y} = (\text{E, E, E}, 0, 1, \alpha, 0)$?

- [ ] $z_0 = \alpha$, $z_1 = \alpha^3$, $z_2 = 0$.
- [ ] $z_0 = \alpha$, $z_1 = \alpha^3$, $z_2 = \alpha^3$.
- [ ] $z_0 = 1$, $z_1 = 0$, $z_2 = \alpha^3$.
- [ ] Die Decodierung führt zu keinem Ergebnis.

e) Welches Ergebnis liefert die Decodierung von $\underline{y} = (\text{E, E, E}, 0, 1, \alpha, \text{E})$?

- [ ] $z_0 = \alpha$, $z_1 = \alpha^3$, $z_2 = 0$, $z_6 = 1$.
- [ ] $z_0 = \alpha$, $z_1 = \alpha^3$, $z_2 = \alpha^3$, $z_6 = 1$.
- [ ] $z_0 = 1$, $z_1 = 0$, $z_2 = \alpha^3$, $z_6 = 1$.
- [ ] Die Decodierung führt zu keinem Ergebnis.

Z2.11: Erasure–Kanal für Symbole



Das Kanalmodell **Binary Erasure Channel** (BEC) beschreibt einen Auslöschungskanal auf Bitebene. Ein Binärsymbol **0** bzw. **1** wird mit der Wahrscheinlichkeit $1 - \lambda$ richtig übertragen und mit der Wahrscheinlichkeit λ als Auslöschung E (*Erasure*) markiert. Im Gegensatz zum **BSC** kann es hier nicht zu Verfälschungen (**0** → **1**, **1** → **0**) kommen.

Ein Reed–Solomon–Code basiert auf einem Galoisfeld GF(2^m) mit ganzzahligem m. Jedes Codesymbol c lässt sich somit durch m Bit darstellen. Will man hier das BEC–Modell anwenden, so muss man dieses zum ***m*–BEC–Modell** modifizieren, wie es in der unteren Grafik für $m = 2$ gezeigt ist:

Alle Codesymbole – in binärer Darstellung **00**, **01**, **10** und **11** – werden mit der Wahrscheinlichkeit $1 - \lambda_2$ richtig übertragen. Damit beträgt die Wahrscheinlichkeit für ein ausgelöschtes Symbol λ_2. Zu beachten ist, dass bereits ein einziges ausgelöschtes Bit zum ausgelöschten Empfangssymbol y = E führt.

Hinweis: Die Aufgabe gehört zu **Kapitel 2.4.** Bei einem auf GF(2^m) basierenden Code ist das skizzierte 2–BEC–Modell zum m–BEC zu erweitern. Die Auslöschungswahrscheinlichkeit dieses Modell wird dann mit λ_m bezeichnet.

Für die Teilaufgaben (a), (b) und (c) gelte für die Auslöschungswahrscheinlichkeit des Grundmodells gemäß der oberen Grafik stets $\lambda = 0.2$.

Fragebogen zu "Z2.11: Erasure–Kanal für Symbole"

a) Es gelte $\lambda = 0.2$. Mit welchen Wahrscheinlichkeiten treten beim BEC–Modell die möglichen Empfangswerte auf?

1–BEC: $\mathbf{Pr}(y = 0)$ =

$\mathbf{Pr}(y = \mathrm{E})$ =

$\mathbf{Pr}(y = 1)$ =

b) Wie groß ist die Auslöschungswahrscheinlichkeit λ_2 auf Symbolebene, wenn der Reed–Solomon–Code auf $\mathrm{GF}(2^2)$ basiert ($\lambda = 0.2$)?

2–BEC: λ_2 =

c) Wie groß ist die Auslöschungswahrscheinlichkeit λ_m, wenn das m–BEC–Modell an den $\mathrm{RSC}\ (255, 223, 33)_{256}$ angepasst wird ($\lambda = 0.2$)?

***m*–BEC:** λ_m =

d) Wie groß darf die Auslöschungswahrscheinlichkeit λ beim Grundmodell (BEC) maximal sein, damit $\lambda_m \le 0.2$ gilt?

$\lambda_m \le$ **0.2: Max[**λ**]** =

e) Mit welcher Wahrscheinlichkeit wird damit das „Nullsymbol” empfangen?

$\lambda_m = 0.2$: $\mathbf{Pr}(y_{\mathrm{bin}} = 00000000)$ =