Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Kommunikationstechnik
Prof. Dr. H. Bölcskei

P&S "Bits on Air"
4. Teil

Markus Gärtner & Felix Kneubühler

Revidierte Version vom 29. April 2022

1 Einleitung

Ein nur aus einem Modulator und einem Demodulator bestehendes drahtloses Kommunikationssystem funktioniert nur unter sehr idealen Bedingungen. Bei Verwendung der Detektionsregel

$$\hat{B}[n] = \arg\max_i z_i[n],$$

wobei

$$z_i[n] = \int r(t) \cdot x_i(t) \cdot p(t - n\tau_S) dt,$$

und

$$p(t) = \begin{cases} 1, & t \in [0, \tau_S) \\ 0, & t \notin [0, \tau_S) \end{cases}$$

(siehe Teil 3 des PPS), wurde angenommen, dass das ankommende Signal $r(t)$ mit den Signalen $x_i(t)$ und dem Puls $p(t-nT_\mathrm{s})$ synchronisiert ist, d.h. dass der Empfänger den exakten Zeitpunkt kennt, an dem a) die gesendete Sequenz eintrifft bzw. beginnt und b) ein neues Symbol im Datenstrom beginnt. Insbesondere in drahtlosen Systemen ist dieser Zeitpunkt jedoch unbekannt, da der Abstand zwischen Sender und Empfänger stark variieren kann und die resultierenden Laufzeitunterschiede entsprechend gross sein können. Dementsprechend benötigt jeder Empfänger eines drahtlosen Kommuniationssystems ein Modul zur Synchronisation.

Die Synchronisation erfolgt in zwei Schritten. Die Synchronisation auf (Symbol-)Taktphase von $x_i(t)$ bezeichnet man als Symbolsynchronisation. Nach der Symbolsynchronisation wird die Rahmensynchronisation durchgeführt. Die Rahmensynchronisation dient dazu, den Anfang der gesendeten Bitsequenz zu finden. Dies wird mit Hilfe einer bekannten Sequenz durchgeführt, die dem Empfänger an bekannter Position - z.B. am Anfang und oder am Ende - im zu übertragenden Bitstrom übergeben wird.

2 Synchronisation

Alle Verfahren zur Synchronisation haben die Aufgabe, eine Schätzung vorzunehmen. Im Falle der Symbolsynchronisation wird die (Symbol-)Taktphase geschätzt, und das Ergebnis ist eine Phasendifferenz bzw. ein Laufzeitunterschied $\Delta_{\mathrm{S}} \in [0, T_{\mathrm{s}}]$. Im Falle der Rahmensynchronisation wird der Beginn einer Sequenz geschätzt, der Schätzwert ist eine Verzögerungszeit $\Delta_{\mathrm{R}} \in [0, \infty)$. Prinzipiell sind viele Möglichkeiten denkbar, auf einen Datenpuls oder den Beginn einer Datensequenz zu synchronisieren. Unterschiedliche Verfahren sind unterschiedlich gut, wobei die Güte dieser Verfahren üblicherweise anhand der Varianz der Schätzwerte festgestellt wird. Verfahren mit grosser Varianz sind i.A. schlechter als solche mit kleiner Varianz.

2.1 Korrelationsverfahren

Mit Hilfe der Estimationstheorie kann man optimale Verfahren angeben, d.h. die Verfahren, die das Resultat mit der geringsten Varianz schätzen können. Ein solches Verfahren ist das Korrelationsverfahren, das in diesem Abschnitt beschrieben wird. Für das Korrelationsverfahren werden die Auto- bzw. Kreuzkorrelationsfunktion benötigt, für die eine kurze intuitive Herleitung im folgenden gegeben wird.

Ziel der Symbolsynchronisation ist es, die Verzögerung Δ_{S} eines empfangenen Signals $r(t)$ gegenüber der eines verschobenen Referenzsignals im Empfänger, $\tilde{r}(t + \Delta_{\mathrm{S}})$, zu schätzen. Gehen wir hierfür aus von der Cauchy-Schwarzschen Ungleichung

$$|\langle a, b \rangle| \leq \|a\| \|b\|, \tag{1}$$

für zwei Vektoren a und b, den Betrag $|.|$, und ein geeignetes Skalarprodukt $\langle a, b \rangle$ mit entsprechender Norm $\|a\| = \sqrt{\langle a, a \rangle}$.

Im Falle des Synchronisationsproblems werden die gesendete, in diesem Fall rauschfreie Sequenz und das zur Demodulation verwendete Signal auf der Empfängerseite als unendlich lang betrachtet; Randeffekte zu Beginn einer Sequenz brauchen damit nicht berücksichtigt werden. Damit ergibt sich, dass das gesendete, modulierte Signal $r(t)$ und das dazu verschobene Signal des Empfängers, $\tilde{r}(t + \Delta_{\mathrm{S}})$, zwei Vektoren aus dem Raum der periodischen Signale sind. Für ein Skalarprodukt der Form

$$\langle a(t), b(t) \rangle = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} a(t) b(t) dt$$

ergibt die Norm $\|a\|$ den *Effektivwert* des Signales $a(t)$. Damit ist für die Signale $r(t)$ und $\tilde{r}(t + \Delta_{\mathrm{S}})$ die rechte Seite von (1) unabhängig von Δ_{S} konstant.

Die Cauchy-Schwarzsche Ungleichung wird genau dann maximal, wenn a und b linear abhängig sind. Für $r(t)$ und $\tilde{r}(t + \Delta_{\mathrm{S}})$ ist dies sicher der Fall, wenn $\Delta_{\mathrm{S}} = 0$. Damit reduziert sich das Synchronisationsproblem auf die Suche des Maximums von

$$|\langle r(t), \tilde{r}(t + \Delta_{\mathrm{S}}) \rangle| = \left| \frac{1}{T_{\mathrm{s}}} \int_{-\frac{T_{\mathrm{s}}}{2}}^{\frac{T_{\mathrm{s}}}{2}} r(t) \tilde{r}(t + \Delta_{\mathrm{S}}) dt \right|.$$

Die geschätze Verzögerungszeit $\hat{\Delta}_{\mathrm{S}}$ ergibt sich als

$$\hat{\Delta}_{\mathrm{S}} = \arg \max_{\Delta_{\mathrm{S}} \in [0, T_{\mathrm{s}})} |\langle r(t), \tilde{r}(t + \Delta_{\mathrm{S}}) \rangle|.$$

Es sei erwähnt, dass die Verzögerung $\Delta_{\mathrm{S}} = 0$ nicht notwendigerweise das einzige globale Maximum der Korrelationsfunktion ist.

Die hier behandelten Signale sind deterministisch. Aber auch in der Stochastik werden Integrale der Form $f(\Delta) = \int \mathrm{a}(t + \Delta)\mathrm{b}(t) dt$ im Zusammenhang mit Korrelationsfunktionen definiert. Da durch die Elektronik des Empfängers empfangene Sequenzen grundsätzlich verrauscht sind, arbeitet man oft auch mit stochastischen Signalen. Aus diesem Grund soll im folgenden ein kurzer Überblick über Auto- und Kreuzkorrelationsfunktionen gegeben werden.

2.1.1 Die Autokorrelationsfunktion

Die *Autokorrelationsfunktion (AKF)* $\varphi(\tau)$ eines gegebenen Signals $f(t)$ wird für *stationäre und ergodische stochastische Prozesse* wie folgt berechnet:

$$\varphi(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\, f(t + \tau)\, \mathrm{d}t. \tag{2}$$

An der Stelle $\tau = 0$ ergibt sich wieder die Leistung des Signals $f(t)$: $P = \varphi(0)$. Für $\tau > 0$ wird der Wert der AKF kleiner.

Aus (2) kann sofort gezeigt werden, dass die AKF eine gerade Funktion ist:

$$\varphi(\tau) \; = \; \varphi(-\tau).$$

In der Praxis muss man sich auf begrenzte Zeitintervalle beschränken. Die derart gebildeten Funktionen heissen *Kurzzeit-Autokorrelationsfunktionen* $\hat{\varphi}(\tau)$:

$$\hat{\varphi}(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\, f(t + \tau)\, \mathrm{d}t.$$

Ist das Signal $f(t)$ stochastisch (z.B. weil es einen Rauschterm enthält), so wird es für jede Realisierung $f(t)$ einen etwas anderen Verlauf nehmen, d.h. die Grösse $\hat{\varphi}(\tau)$ ist selbst eine Zufallsvariable, deren Werte um die exakten Werte von $\varphi(\tau)$ gemäss (2) schwanken. Die Funktion $\hat{\varphi}$ ist also ein Schätzer für die AKF, der für grosse T gegen φ konvergiert.

Die AKF eines *periodischen* Signals mit der Periode T_0 ist wie folgt definiert:

$$\varphi(\tau) \; = \; \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(t)\, f(t + \tau)\, \mathrm{d}t.$$

Man beachte, dass $\varphi(\tau + T_0) = \varphi(\tau)$ wegen $f(t + T_0) = f(t)$.

Bei *energiebegrenzten* Signalen hat $\varphi(\tau)$ nicht die Dimension einer Leistung, sondern diejenige einer Energie:

$$\varphi(\tau) \; = \; \int\limits_{-\infty}^{\infty} f(t)\, f(t+\tau)\, \mathrm{d}t.$$

Für *abgetastete* Signale ist die Autokorrelationsfunktion diskret,

$$\varphi(n) = \sum_{\mathbb{Z}} f(k) f(k+n).$$

Beachten Sie, dass bei abgetasteten Signalen $f(k)$ mit endlichem Träger $k \in \{1, ..., N\}$ der Träger von φ grösser ist als der von f.

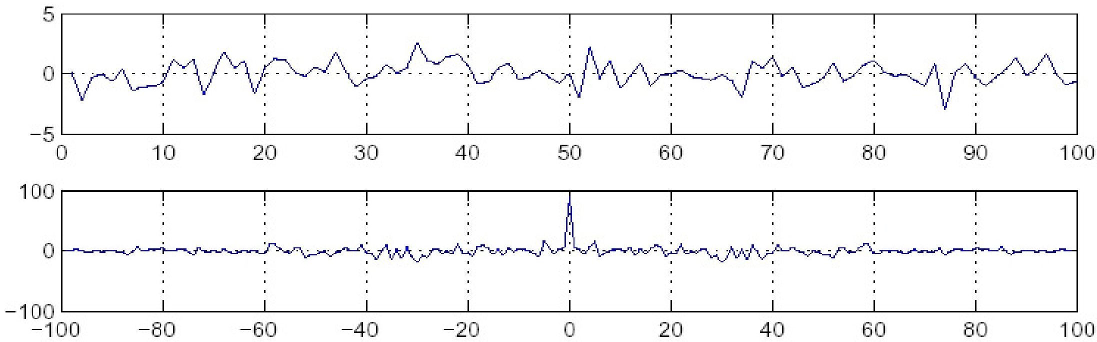

Abbildung 1: Graph einer Autokorrelation

2.1.2 Die Kreuzkorrelationsfunktion

Nachdem in Abschnitt 2.1.1 die Korrelationsfunktion eines Signals $f(t)$ mit sich selbst gebildet wurde, kann diese auch für zwei verschiedene Signale $f_1(t)$ und $f_2(t)$ berechnet werden. In diesem Fall spricht man von der *Kreuzkorrelationsfunktion (KKF)* $\varphi_{12}(\tau)$ der beiden Signale. Die Kreuzkorrelationsfunktion ist wie folgt definiert:

$$\varphi_{12}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int\limits_{-\infty}^{\infty} f_1(t)\, f_2(t+\tau)\, \mathrm{d}t. \tag{3}$$

Es ist natürlich möglich, anstelle von (3) die Funktion $\varphi_{21}(\tau)$ zu bilden:

$$\varphi_{21}(\tau) \; = \; \lim_{T\to\infty} \frac{1}{T} \int\limits_{-\infty}^{\infty} f_2(t)\, f_1(t+\tau)\, \mathrm{d}t.$$

Man kann leicht zeigen, dass
$$\varphi_{12}(\tau) = \varphi_{21}(-\tau)$$
gilt. Man beachte jedoch, dass die KKF im Gegensatz zur AKF im allgemeinen weder eine gerade noch eine ungerade Funktion von τ ist und das Maximum auch nicht notwendigerweise bei $\tau = 0$ liegt. Analog zur AKF existieren auch für die KKF die modifizierten Definitionen für periodische Signale, Signale mit endlicher Energie, die Kurzzeit-Kreuzkorrelationsfunktionen und diskrete KKF.

In der Kommunikationstechnik werden AKFs und KKFs gleichermassen benötigt. Ein verrauschtes Empfangssignal r wird üblicherweise zerlegt in den Anteil s ohne Rauschen und den Rauschanteil, n,
$$r(t) = s(t) + n(t), \tag{4}$$
wobei die Rauschvariable einer bestimmten Statistik gehorcht. Üblicherweise wird sie als gaussverteilt angenommen, d.h. ihr Histogramm kann für viele Realisierungen durch eine Gaussverteilung angenähert werden. Die Korrelation des gesendeten Signals $r(t)$ mit dem Signal $s(t + \Delta)$ des Demodulators wird dann durch eine KKF beschrieben. Wegen (4) kann diese KKF aber aufgespalten werden in die AKF von s und die Kreuzkorrelation zwischen s und dem Rauschterm.

2.2 Kreuzkorrelation in `Matlab`

Die obigen Abschnitte haben einen tieferen Einblick in die Auto- und Kreuzkorrelation gegeben, so dass diese im Prinzip von Hand programmiert werden könnten. Dieses PPS soll jedoch das grundlegende Verständnis dieser Methoden vermitteln und nicht ein reiner `Matlab`-Kurs darstellen. Aus diesem Grund soll in diesem Abschnitt die Kreuzkorrelations-Funktion `xcorr()` von `Matlab` betrachtet werden.

Aus den obigen Formeln kann man sich die Kreuzkorrelation graphisch vorstellen, wobei in diesem Abschnitt ausschliesslich die diskrete Korrelation betrachtet werden soll:
$$\varphi(n) = \sum_{k \in \mathbb{Z}} f_1(k) f_2(k+n)$$

Wird $n = 0$ gesetzt, erhält man für $\varphi(0)$ die Summe der Produkte der Funktionswerte an jedem Abtastpunkt, wobei $f_2(k)$ nicht verschoben wurde. Variiert man $n \in [-\infty, \infty]$, so wird $\varphi(n)$ ein Vektor mit den Werten der Korrelationsfunktion für jede Verschiebung von $f_2(k)$. Wie aus der Formel ersichtlich ist, stimmen beim Maximalwert im erhaltenen Vektor die beiden Signale $f_1(k)$ und $f_2(k)$ am besten überein.

Anhand der folgenden Aufgaben wird Schritt für Schritt die Funktionsweise der Kreuzkorrelation sowie die von `xcorr()` vermittelt.

Aufgaben:

- Um mit `xcorr()` arbeiten zu können, lesen Sie zuerst die `Matlab`-Hilfe zu dieser Funktion genau durch! Beachten Sie ausserdem die Anzahl Rückgabeargumente von `xcorr( )`.

- Erstellen Sie mit `rand()` einen zufälligen kurzen Vektor u mit den Einträgen $u_i \in \{1, -1\}$. Benützen Sie hierfür die Vorgehensweise aus Teil 3. Führen Sie eine Kreuzkorrelation mit dem erstellten Vektor und dem Vektor $[1, -1]$ durch. Berechnen Sie das Resultat zuerst auf dem Papier und betrachten Sie dann den Output von `xcorr()`.

- Schreiben Sie ein Programm `offset = signalsync(a,b)` welches die zwei allgemeinen Vektoren `a` und `b` kreuzkorreliert (wobei wir davon ausgehen, dass `length(a)»length(b)`). Es soll der Offset von `b` zu `a` herausgegeben werden, also soll `a(offset:offset+length(b)-1)` $\simeq$ `b`. Stellen Sie das Ergebnis graphisch dar.
  Beispiel: `signalsync([1,-1,-1,1,1,-1],[-1,1])=3` also: der Vektor `[-1,1]` kommt an `3`ter Stelle im Vektor `[1,-1,-1,1,1,-1]` vor.
  Wieso werden die Werte -1,1 verwendet und nicht 0,1? Was passiert wenn die Sequenz b mehrmals in a vorkommt? Ist die Vereinfachung `length(a)»length(b)` überhaupt nötig?

2.3 Symbolsynchronisation

Für das zu implementierende Modem wird die Detektion der Bits eines Datenstroms offline geschehen, dass heisst, die Daten werden aufgenommen, abgespeichert, und anschliessend vom Empfänger verarbeitet. Da der Empfänger bereits eingeschaltet sein muss, wenn der Sender zu senden beginnt, wird die empfangene Datensequenz zunächst Rauschen enthalten. Erst nach einer unbekannten Verzögerung Δ_M beginnt der Datenstrom. Die zuerst durchgeführte Symbolsynchronisation kann diese Verzögerung maximal um eine Symbollänge korrigieren und ermöglicht dadurch die Detektion von Bits.

Die Synchronisation mit Hilfe des Maximums einer Korrelationsfunktion wurde anhand der binären Frequenzumtastung in einem Symbolintervall erläutert. Für einen Bitstrom aus N Symbolen bzw. Bits ergibt sich das Maximum als das Δ_S, an dem die Summe der Absolutwerte der Korrelationsfunktionen eines Symbols maximal ist,

$$\begin{aligned}\hat{\Delta}_\mathrm{S} &= \arg \max_{\Delta_\mathrm{S} \in [0, T_\mathrm{s})} |\langle r(t+\Delta_\mathrm{S}), \tilde{r}(t)\rangle| \\ &= \arg \max_{\Delta_\mathrm{S} \in [0, T_\mathrm{s})} \sum_{n=-1}^{N-1} \left| \int_{nT_S}^{(n+1)T_S} r(t+\Delta_\mathrm{S}) x(t) p(t-nT_\mathrm{s}) \mathrm{d}t \right|,\end{aligned}$$

wobei n alle Symbole der Dauer T_S durchläuft.

Anders ausgedrückt: Beim Empfangen des modulierten Signals tritt das Problem auf, dass das Mikrophon vor dem Beginn des eigentlichen Signals ein rauschen zufälliger Länge aufnimmt. Bei der Demodulation wird jeweils über die Symboldauer τ_s die Multiplikation vom empfangenen Signal und jedem Träger aufintegriert. Um diese Integration korrekt ausführen zu können müssen wir wissen, wo genau ein Symbol anfängt, um über dieses Integrieren zu können. Durch die Kreuzkorrelation mit der Erkennungssequenz für den Beginn der Datenübertragung (siehe 2.4) finden wir eine Stelle, an der ein Symbol anfängt. Für genauere Erklärung beim Assistenten nachfragen.

Aufgabe:

- Schreiben Sie ein Programm, das die Symbolsynchronisation durchführt.
  Bemerkung: Die Symbolsynchronisation kann entweder von Hand programmiert oder mit Hilfe der in `Matlab` eingebauten Funktion `xcorr()` realisiert werden. Sie können auch das Programm `signalsync` von vorher wieder verwenden.
- Testen Sie das Programm, indem Sie vor und nach Ihrem Empfangssignal eine zufällige Anzahl Nullen setzen und dieses synchronisieren.
- Das Ziel ist, dass der Rückgabevektor von `Symbolsync` direkt für die Demodulation berücksichtigt werden kann (beachten Sie die Länge des Signals).

2.4 Rahmensynchronisation

Ein empfangsbereites Modem ist stets dem Rauschen des Kanals ausgesetzt und braucht daher einen Mechanismus, der den Anfang einer tatsächlich gesendeten Sequenz identifiziert. Diesen Mechanismus bezeichnet man als *Rahmensynchronisation*.

Für das zu implementierende Modem wird die Detektion der Bits eines Datenstroms offline geschehen, dass heisst, die Daten werden aufgenommen, abgespeichert, und anschliessend vom Empfänger verarbeitet. Da der Empfänger bereits eingeschaltet sein muss, wenn der Sender zu senden beginnt, wird die empfangene Datensequenz zunächst Rauschen enthalten. Erst nach einer unbekannten Verzögerung Δ_{M} beginnt der Datenstrom. Die zuerst durchgeführte Symbolsynchronisation kann diese Verzögerung maximal um eine Symbollänge korrigieren. Diese Korrektur ermöglicht die Detektion von Bits, gibt aber noch keine Auskunft darüber, welche Bits aus dem Rauschen zu Beginn des Empfangs, und welche Bits tatsächlich aus der informationstragenden Sequenz stammen. Die Rahmensynchronisation muss also sicherstellen, dass nur diejenigen Bits als Daten interpretiert werden, die nach der Verzögerung Δ_{M} am Empfänger eintreffen.

Eine sehr einfache Art der Rahmensynchronisation ist die Implementierung eines Leistungsdetektors mit einen Schwellenwert. Übersteigt ein empfangenes Signal diese Schwelle (d.h. Lautstärker überschreitet Schwellwert), so wird dieses Signal vom Modem als Beginn einer Datensequenz interpretiert und die Detektion von Bits beginnt. Diese Methode kann jedoch bei einem drahtlosen, akustischen Modem leicht zu Fehlern führen, z.B. wenn jemand in die Hände klatscht und dieses Störgeräusch die Schwelle des Empfängers überschreitet.

Ein besseres Rahmensynchronisation kann ein Korrelationsverfahren benutzt werden. Dieses Verfahren dient dazu, in den bereits detektierten Daten eine sowohl dem Sender wie auch dem Empfänger bekannte *Pilotsequenz* $p(n)$ (auch *Präambel*) zu detektieren, die den Daten vorangestellt wird. Damit können alle anderen zur Sequenz gehörenden Daten erkannt werden. Die Korrelationsfunktion, auf dem das Verfahren basiert, ist jetzt diskret, da bereits mit detektierten Bits gearbeitet wird.

Aufgabe:

- Für das akustische Modem soll eine einfache (suboptimale, [2]) Version einer Rahmensynchronisation implementiert werden. Programmieren Sie dazu in Ihren Sender eine Präambel, die zu Beginn jeder Datenübertragung gesendet wird. Die Prämbel sollte genügend lang sein, damit sie nicht zufällig ebenfalls in der zu sendenden Bitsequenz vorkommt, und sie sollte genug Bitwechsel enthalten, um die Symbolsynchronisation nicht zu erschweren.

  Die diskrete KKF lautet

  $$\varphi_{z,p}(n) = \sum_m z(m)\, p(m+n),$$

  wobei p die Pilotsequenz und z die bereits detektierten Daten sind. Damit die KKF an der richtigen Stelle maximal wird, muss die Bitsequenz allerdings die Werte -1 und 1 annehmen, nicht 0 und 1.

  Implementieren Sie ebenfalls eine Methode, um das Ende der Sequenz zu erkennen. Implementieren Sie die Rahmensynchronisation auch hier entweder von Hand oder mittels `xcorr()` in Ihrem Empfänger. Beachten Sie den Definitionsraum der Eingaben, sollten die Werte {0,1} oder {-1,1} verwendet werden?

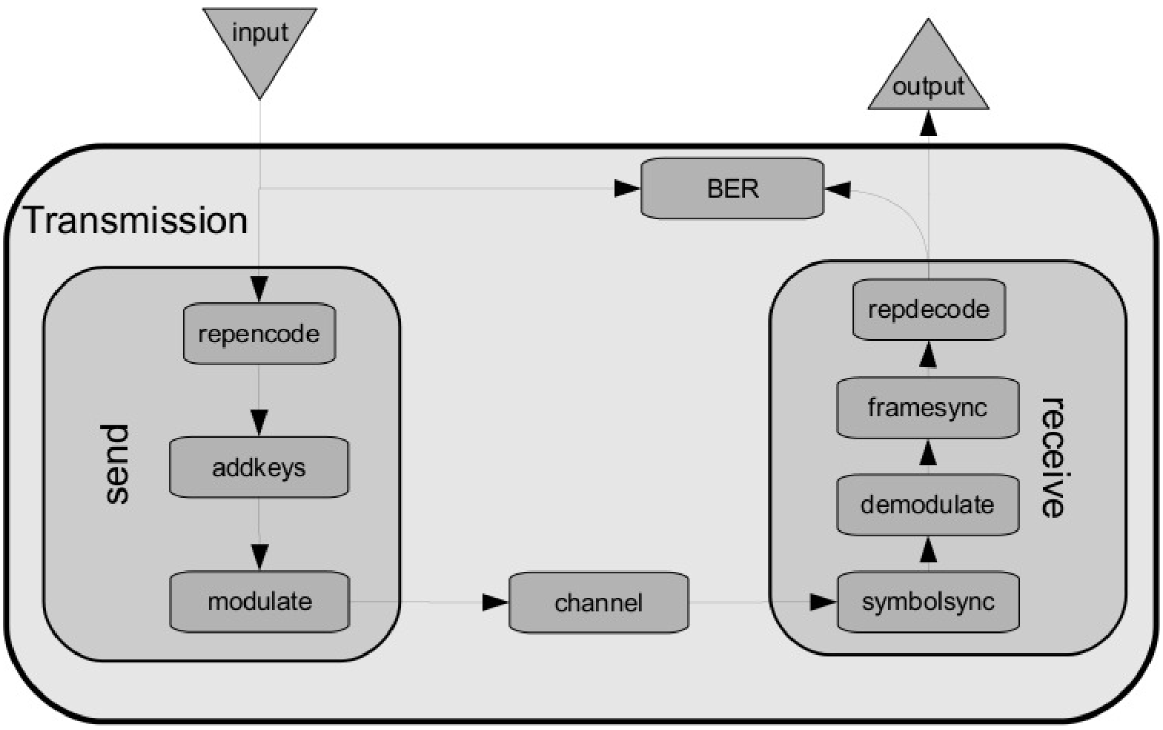

Abbildung 2: So soll der generelle Flow des Programmes ablaufen.

- **Wichtig:** Überprüfen Sie abschliessend Ihren Code auf Funktionalität. Verwenden Sie dazu den Ablauf wie in Abbildung 2 und bauen Sie manuell Störungen ein (zufällige (grosse!) Anzahl 0en vor und nach der eigentlichen Bitsequenz).

Literatur

[1] J.G. Proakis, ”Digital Communications”, 5th edition, McGraw-Hill, New York, 2007.

[2] J. L. Massey, “Optimum Frame Synchronization”, IEEE Trans. Comm., Vol. COM-20, No. 2, April 1972.