

СТАТИСТИЧЕСКИЕ МОДЕЛИ ЗАДАЧИ СИНХРОНИЗАЦИИ

А.Н.Манохин, С.П.Сидоров, Н.Т.Тукубаев

В работе рассматривается задача синхронизации последовательности $\{x_i\}$, $i = \overline{1, n}$; $x_i = 0$ либо 1. Последовательность $\{x_i\}$ получается на выходе детектора, который по входному сигналу $f(t)$ на интервале времени $[t_0, t_0 + \Delta t]$ вырабатывает одно из двух решений: 0 либо 1. Задача синхронизации сводится к определению в последовательности $\{x_i\}$ момента k , с которого начинается полезный сигнал, т.е. сигнал, несущий передаваемую информацию. Алгоритм определения момента k основывается на том, что последовательность $\{x_i\}$ имеет, начиная с момента k , специальную структуру и, следовательно, можно сформулировать априорные сведения о распределении случайного вектора $\{x_i\}$.

Задача синхронизации в цифровых системах связи рассматривалась в [1, 2]. Новизна и место настоящей работы определяются следующими ее особенностями:

- в ней решается задача синхронизации сигнала, дискретного по времени и значению (обычно рассматривается непрерывный сигнал на фоне гауссовского шума);

- строится игровая статистическая модель, которая, по нашему мнению, наиболее адекватно отражает существо содержательной задачи;

- рассматриваются вопросы алгоритмической реализации предложенных методов синхронизации на ЭВМ.

1. Модель сигнала, задающего квант информации. Рассмотрим вероятностную модель бинарного сигнала $\bar{x} = \{x_i\}$, $i = \overline{1, n}$. Случайный вектор \bar{x} полностью определяется совместным распределением, которое задается следующим образом. На множестве кодовых комбинаций $\Phi = \{c\}$, где $c = (c_1, \dots, c_n)$, $c_j = 0$ либо 1, задано распре-

деление p_1 . Конкретная комбинация C является выборкой из p_1 . На множестве целых чисел $\{k_0, k_0+1, \dots, k_1\}$, $k_0 < k_1$, задано распределение p_2 . Момент k является выборкой из p_2 . Без ограничения общности можно считать $k_0 = 1$. Рассмотрим множества:

$$J(1, k) = \{k, k+1, \dots, k+l(1)-1\};$$

$$J(2, k) = \{k+l(1), \dots, k+l(1)+l(2)-1\};$$

.....

$$J(m, k) = \{k + \sum_{i=1}^{m-1} l(i), \dots, k + \sum_{i=1}^m l(i)-1\}.$$

Величину $l(j)$ назовем длиной j -й посылки ($1 \leq j \leq m$), а s_j - ее значением.

По заданным C и k строим \bar{s} - вектор из нулей и единиц длины $n = k_1 + \sum_{j=1}^m l(j) - 1$, такой, что $p(s_i = 0) = p(s_i = 1) = 1/2$

для $1 \leq i \leq k-1$ и $k + \sum_{j=1}^m l(j) \leq i \leq n$; $s_i = s_j$ для $i \in J(j, k)$.

Далее на вектор \bar{s} накладывается вектор ошибок $\bar{e} = \{e_i\}$, $i = \overline{1, n}$, $e_i \in \{0, 1\}$. Если $e_i = 0$, то $x_i = s_i$, если $e_i = 1$, то $x_i = \bar{s}_i$,

где \bar{s}_i - инвертированное значение s_i . Вектор \bar{e} - выборка из некоторого распределения p^e , заданного на множестве всех возможных \bar{e} . Рассмотрим примеры возможных распределений.

1) Модель независимых, одинаково распределенных ошибок, задается условиями: $p(e_i = 0) = p > 1/2$, $p(e_i = 1) = 1-p$, e_i и e_j независимы для любых i и j .

2) Марковская модель задается условиями:

$$p(e_i = 1/e_{i-1} = 1) = p(1, 1),$$

$$p(e_i = 1/e_{i-1} = 0) = p(0, 1),$$

$$p(e_i = 0/e_{i-1} = 1) = p(1, 0),$$

$$p(e_i = 0/e_{i-1} = 0) = p(0, 0),$$

$$0 \leq p(i, j) \leq 1, p(i, 1) + p(i, 0) = 1 \text{ для } i = 1, 0; p(e_i = 0) = p > 1/2, \\ p(e_{i+1} = 0) = p(e_i = 0) \cdot p(0, 0) + p(e_i = 1) \cdot p(1, 0), p(e_{i+1} = 1) = \\ = p(e_i = 0) \cdot p(0, 1) + p(e_i = 1) \cdot p(1, 1) \text{ для } 1 \leq i \leq n-1.$$

3) Марковская ν -связная модель строится аналогично примеру 2, но учитывается зависимость не от одной, а от ν предыдущих величин.

Если момент k и кодовая комбинация C определены, то распределение p^e определяет условное распределение $P(\bar{x}/C, k)$. Совместное распределение C, k и \bar{x} имеет вид:

$$p_1(C) \cdot p_2(k) \cdot P(\bar{x}/C, k) . \quad (1)$$

Случайный вектор \bar{x} будем называть квантом информации.

Содержательная задача заключается в оценивании k и C по вектору \bar{x} , который наблюдается в эксперименте. При этом нам известны следующие данные: m - число посылок, $l(j)$, $j = \overline{1, m}$, - длины посылок, k_1 - размер окна неопределенности, а также, возможно, значения отдельных посылок. В реальных устройствах эту задачу, как правило, решают в два этапа. На первом строят оценку \hat{k} для k , на втором - по \hat{k} и известным значениям $l(j)$ достаточно просто оценивают C . Алгоритм вычисления момента \hat{k} будем называть алгоритмом синхронизации. Наша цель состоит в том, чтобы рассмотреть процедуры построения оптимальной оценки для k в различных статистических постановках.

Для иллюстрации приведем пример задачи синхронизации с типичными значениями параметров. Пусть число посылок $m = 24$, $l(2) = l(23) = 24$, $l(j) = 16$, для остальных $1 \leq j \leq m$. Значения 1-й, 2-й, 23-й и 24-й посылки известны: $c_1 = 0$, $c_2 = 1$, $c_{23} = 1$, $c_{24} = 0$. Значения c_j для $3 \leq j \leq 22$ не известны, и все возможные кодовые комбинации равновероятны. Размер окна $k_1 = 32$, и момент k с одинаковой вероятностью может принять любое значение из этого диапазона. Ошибки распределены одинаково и независимо. Наблюдается вектор \bar{x} длины $n = k_1 + \sum_{j=1}^{24} l(j) - 1$. Необходимо оценить неизвестную кодовую комбинацию C и момент k .

2. Игровая модель задачи синхронизации. Попытаемся обосновать рациональность двухэтапной процедуры оценивания C по \bar{x} и получить оптимальный и приближенный к нему алгоритмы синхронизации в игровой статистической модели [3, гл. III] и в статистической модели, основанной на принципе максимального правдоподобия. Перенесем наши понятия в определения общей игровой модели. Пусть

$X = \{ \bar{x} \}$ - множество возможных \bar{x} (пространство исходов испытаний);

$\Phi = \{ C \}$ - множество состояний природы;

$r(\bar{x})$ - стратегия игрока, которая ставит в соответствие каждому \bar{x} решение $r(\bar{x}) = C'$;

$g(c, c')$ - функция потерь: $g(c, c') = 0$, если $c = c'$, и $g(c, c') = 1$, если $c \neq c'$;

$Mg(c, r(\bar{x}))$ - риск (матожидание $g(c, r(\bar{x}))$). Напомним, что c и \bar{x} строятся в соответствии с распределением (I). Оптимальный алгоритм r_0 определяется выражением:

$$r_0 = \arg \min_r Mg(c, r(\bar{x})) . \quad (2)$$

Условие (2) эквивалентно следующему условию: $r_0(\bar{x}) = c'$, где $P(c'/\bar{x}) \geq P(c/\bar{x})$ для любого $c \in \Phi$. Действительно, выражение для риска можно представить в виде:

$$\begin{aligned} Mg(c, r(\bar{x})) &= \sum_{c \in \Phi} \sum_{\bar{x} \in X} p_1(c) \cdot P(\bar{x}/c) \cdot g(c, r(\bar{x})) = \\ &= \sum_{\bar{x} \in X} \sum_{c \in \Phi} p_1(c) P(\bar{x}/c) g(c, r(\bar{x})) . \end{aligned}$$

Если для \bar{x} найдено c' такое, что $p_1(c') P(\bar{x}/c') \geq p_1(c) \cdot P(\bar{x}/c)$ для любого $c \in \Phi$ и $r(\bar{x}) = c'$, то каждое слагаемое в сумме по $\bar{x} \in X$ минимально и, следовательно, вся сумма минимальна. Но неравенство $p_1(c') P(\bar{x}/c') \geq p_1(c) P(\bar{x}/c)$ эквивалентно неравенству $P(c'/\bar{x}) \geq P(c/\bar{x})$. Через моменты k его можно записать следующим образом:

$$p_1(c') \sum_{1 \leq k \leq k_1} p_2(k) P(\bar{x}/c', k) \geq p_1(c) \sum_{1 \leq k \leq k_1} p_2(k) P(\bar{x}/c, k) .$$

При $p_2(c) = 1/\lambda$ и $p_2(k) = 1/k_1$, где λ - количество различных кодовых комбинаций в Φ , т.е. при равновероятных c и k , неравенство, задающее оптимальное правило, имеет вид:

$$\sum_{1 \leq k \leq k_1} P(\bar{x}/c', k) \geq \sum_{1 \leq k \leq k_1} P(\bar{x}/c, k) \quad (3)$$

для каждого $c \in \Phi$.

3. Метод максимального правдоподобия в задаче синхронизации.

С нашей точки зрения, изложенная выше игровая статистическая модель наиболее адекватно отражает содержательный смысл задачи синхронизации. Но полезно рассмотреть ее и как задачу оценки параметра по принципу максимального правдоподобия, так как этот подход в статистических алгоритмах является широко распространенным.

Вероятностная модель задается следующим образом. Момент k и кодовая комбинация C считаются фиксированными (не случайными), но неизвестными, а \bar{x} является выборкой из распределения $P(\bar{x}/C, k)$. В качестве оценки максимального правдоподобия для C и k выбираются

$$(\hat{C}, \hat{k}) = \arg \max_{1 \leq k \leq k_1} \max_{C \in \Phi} P(\bar{x}/C, k). \quad (4)$$

Выражения (3) и (4) определяют различные алгоритмы оценивания C по \bar{x} . В вычислительном плане алгоритм (3) слишком трудоемок, и нам не удалось найти варианта его реализации в масштабе реального времени. Поэтому далее будем рассматривать вопросы реализации алгоритма, основанного на выражении (4). По нашему мнению, оба алгоритма должны давать близкие результаты. Это предположение требует дальнейших исследований.

4. Алгоритм вычисления \hat{k} . Рассмотрим кратко отдельные аспекты реализации алгоритма оценивания момента синхронизации на основе выражения (4).

Пусть Φ состоит из векторов $C = (c_1, \dots, c_m)$ следующего вида. Множество $\{1, \dots, m\}$ разбито на два непересекающихся подмножества M_1 и M_2 мощностью m_1 и m_2 соответственно ($m_1 + m_2 = m$). Если $j \in M_1$, значения c_j одинаковы и известны априори, в противном случае - $j \in M_2$ - c_j принимают произвольные значения. Таким

образом, мощность Φ равна 2^{m_2} . Для модели независимых и одинаково распределенных ошибок получаем из (4) следующую оценку момента синхронизации:

$$\hat{k} = \arg \min_k \gamma(\bar{x}, k), \quad (5)$$

где

$$\begin{aligned} \gamma(\bar{x}, k) &= \sum_{j=1}^m \gamma_j(k), \\ \gamma_j(k) &= \min \left(\sum_{i \in J(j, k)} x_i, 1(j) - \sum_{i \in J(j, k)} x_i \right) \quad \text{для } j \in M_2, \\ \gamma_j(k) &= \sum_{i \in J(j, k)} |x_i - c_j| \quad \text{для } j \in M_1. \end{aligned}$$

Таким образом, функционал γ представим в виде суммы: $\gamma(\bar{x}, k) = \gamma'(\bar{x}, k) + \gamma''(\bar{x}, k)$, где γ' получается суммированием по M_1 , а γ'' - суммированием по M_2 . Рассмотрим рекуррентный алгоритм вы-

числения γ' и γ'' для случая, когда $M_2 = \{j: m' \leq j \leq m''\}$, где $m' \geq 1$, $m'' \leq m$ и $l(j) = 1$ для $m' \leq j \leq m''$. Для $1+1 \leq k \leq k_1$ справедливо рекуррентное соотношение:

$$\gamma''(\bar{x}, k) = \gamma''(\bar{x}, k-1) - \min(A_1(k), 1-A_1(k)) + \min(A_2(k), 1-A_2(k)),$$

$$\text{где } A_1(k) = A_1(k-1) - x_{1_1(k)-1} + x_{1_1(k)},$$

$$A_2(k) = A_2(k-1) - x_{1_2(k)-1} + x_{1_2(k)},$$

$$1_1(k) = k + \sum_{j=1}^{m'-1} l(j) - 1,$$

$$1_2(k) = k + \sum_{j=1}^{m'-1} l(j) + (m'' - m' + 1)l - 1.$$

Начальные значения $\gamma''(\bar{x}, k)$ для $1 \leq k \leq 1$ вычисляются по формуле:

$$\gamma''(\bar{x}, k) = \sum_{j=m'}^{m''} \gamma_j(k),$$

$$\text{где } \gamma_j(k) = \min\left(\sum_{i \in I(j, k)} x_i, l(j) - \sum_{i \in I(j, k)} x_i\right),$$

начальные значения для A_1 и A_2 задаются в виде:

$$A_1(1+1) = \sum_{i=n'}^{n''} x_i, \quad A_2(1+1) = \sum_{i=n'+d}^{n''+d} x_i,$$

$$\text{где } n' = \sum_{j=1}^{m'-1} l(j) + 1, \quad n'' = n' + 1 - 1, \quad d = (m'' - m' + 1)l.$$

Трудоёмкость прямого вычисления $\gamma''(\bar{x}, k)$ через $\gamma_j(k)$ в фиксированный момент времени k определяется выражением $K \cdot N_2$, где K - некоторая константа, а $N_2 = l(m'' - m' + 1)$ - количество отсчетов в посылках, входящих в M_2 . Для предложенного рекуррентного алгоритма трудоёмкость определяется лишь константой K и не зависит от количества отсчетов.

Аналогичный рекуррентный алгоритм можно построить для вычисления $\gamma'(\bar{x}, k)$, но в этом случае удастся снизить трудоёмкость с $K \cdot N_1$, где K - константа, $N_1 = \sum_{j \in M_1} l(j)$ - общее число отсчетов, входящих в посылки из M_1 , до $K \cdot (m - m'' + m' - 1)$, где $m - m'' + m' - 1$ - мощность множества M_1 .

Для марковской модели ошибок вычисление \hat{C} по (4) требует больших затрат времени, так как при отыскании $\max_{C \in \Phi} P(\bar{X}/C, k)$ решения по соседним посылкам, влияя друг на друга, не дают независимого вклада, как это было при независимых ошибках. Есть основания предполагать, что отказ от точной процедуры приведет к незначительным потерям. Поэтому, как и для независимых ошибок, будем считать, что по \bar{x} и k строится $C^* = (c_1^*, \dots, c_m^*)$, где $c_j^* = c_j$ для $j \in M_1$, и

$$c_j^* = \begin{cases} 1, & \text{если } \sum_{i \in T(j, k)} x_i \geq 1(j)/2 \\ 0 & \text{в противном случае,} \end{cases} \quad \text{для } j \in M_2.$$

По C^* и вектору \bar{x} строим вектор ошибок, начиная с точки k , т.е. по C^* вычисляем сигнал $\{s_i\}$, как было описано в разделе I, и, сравнивая его с \bar{x} , получаем вектор ошибок $\{e_i\}$. Подсчитываем по вектору $\{e_i\}$ величины n_{uv} ($u, v \in \{0, 1\}$) - частоты встречаемости в векторе \bar{e} комбинаций 00, 01, 10, 11 и вычисляем функционал

$$\gamma_1(\bar{x}, k) = n_{11} \ln p_{11} + n_{01} \ln p_{01} + n_{10} \ln p_{10} + n_{00} \ln p_{00}.$$

Задача оценки потерь от замены вычисления $\arg \max_{C \in \Phi} P(\bar{X}/C, k)$ процедурой минимизации $\gamma_1(\bar{x}, k)$ требует дальнейшего исследования. Для γ_1 можно построить рекуррентный алгоритм, аналогичный тому, что был построен для γ .

5. Модель сигнала, задающего телеграмму. Выше рассматривалась вероятностная модель задачи синхронизации на одном кванте информации. Рассмотрим последовательность квантов $\{\bar{x}(1), \dots, \bar{x}(L)\}$, где $\bar{x}(i)$ - вектор из нулей и единиц длины n , $1 \leq i \leq L$. Момент k , $1 \leq k \leq k_1$, выбирается из распределения p_2 и не изменяется от кванта к кванту, $\tilde{C} = \{c(1), \dots, c(L)\}$ - независимая выборка из распределения p_1 , $\bar{x}(i)$ - независимая выборка из генеральной совокупности с распределением $P(\bar{X}/C(i), k)$.

Будем опять искать оценки для \tilde{C} и k на основе принципа максимального правдоподобия

$$(\tilde{C}^*, \hat{k}) = \arg \max_{1 \leq k \leq k_1} \max_{C \in \Phi} \prod_{i=1}^L P(\bar{x}(i)/C(i), k). \quad (6)$$

Оценка для модели независимых ошибок имеет вид:

$$\hat{k} = \arg \min_{1 \leq k \leq k_1} \tilde{\gamma}(\tilde{x}, k), \quad \text{где} \quad \tilde{\gamma}(\tilde{x}, k) = \sum_{i=1}^L \gamma(\bar{x}(i), k),$$

а для марковской модели -

$$\hat{k} = \arg \min_{1 \leq k \leq k_1} \tilde{\gamma}_1(\tilde{x}, k), \quad \text{где} \quad \tilde{\gamma}_1(\tilde{x}, k) = \sum_{i=1}^L \gamma_1(\bar{x}(i), k).$$

Нахождение оценки \hat{k} по формуле (6) требует накопления и хранения в памяти всей телеграммы $\tilde{x} = \{\bar{x}(1), \dots, \bar{x}(L)\}$. Но в реальной ситуации, как правило, существует ограничение на объем памяти, заключающееся в том, что мы можем хранить и обрабатывать лишь один вектор $\bar{x}(i)$. В этом случае для обеих моделей строится текущая оценка максимального правдоподобия по первым j квантам. Она имеет вид:

$$\hat{k}_j = \arg \min_{1 \leq k \leq k_1} \sum_{i=1}^j \gamma(\bar{x}(i), k) \quad \text{для модели независимых ошибок,}$$

$$\hat{k}_j = \arg \min_{1 \leq k \leq k_1} \sum_{i=1}^j \gamma_1(\bar{x}(i), k) \quad \text{для марковской модели ошибок.}$$

Легко видеть, что для вычисления \hat{k}_j не нужно хранить в памяти кванты $\bar{x}(1), \dots, \bar{x}(L)$, а достаточно запоминать на каждом j -м шаге вектора $\tilde{\gamma}(\tilde{x}, k, j) = \sum_{i=1}^j \gamma(\bar{x}(i), k)$, $\tilde{\gamma}_1(\tilde{x}, k, j) = \sum_{i=1}^j \gamma_1(\bar{x}(i), k)$, $1 \leq k \leq k_1$, $\bar{x}(j)$. На основе быстрых алгоритмов вычисления γ и γ_1 , рассмотренных в п.4, можно довольно просто построить быстрые рекуррентные алгоритмы вычисления $\tilde{\gamma}$ и $\tilde{\gamma}_1$.

6. Модель изменения момента синхронизации. Выше описывалась вероятностная модель, в которой момент синхронизации задавался одним и тем же для различных квантов, т.е. полезная информация в каждом кванте $\bar{x}(1), \dots, \bar{x}(L)$ начиналась с одного и того же места. Для некоторых задач момент синхронизации изменяется от кванта к кванту. Этот факт описывается следующей моделью. Как и раньше, $C(i)$ выбирается в соответствии с распределением p_1 , $C(i)$ и $C(j)$ - независимы при $i \neq j$. Таким образом, получаем $\mathcal{C} = \{C(1), \dots, C(L)\}$. В соответствии с распределением p_2 выбирается k , $1 \leq k \leq k_1$. Распределение p_3 задает выбор $\tau(i)$ в интервале $k - \tau_0 \leq \tau(i) \leq k + \tau_0$, где τ_0 - заданное целое число. Для каждого i выбор производится

независимо. По $C(i)$ и $\tau(i)$ строим для i -го кванта полезный сигнал \bar{x} и искажаем его в соответствии с вектором ошибок $\bar{\epsilon}$, как это было описано в п. I.

Опуская выкладки, для оптимального τ_0 в этом случае имеем:

$$(\hat{C}, \hat{k}, \hat{\tau}(i)) = \arg \max_{1 \leq k \leq k_1} \max_{k - \tau_0 \leq \tau(i) \leq k + \tau_0} \max_{C(i)} \prod_{i=1}^L P(\bar{x}(i)/C(i), \tau(i)).$$

Этот функционал определяет алгоритм оценивания k и $\tau(i)$.

Рассмотрим модель независимых и одинаково распределенных ошибок в предположении, что память ограничена и можно хранить лишь один квант. В этом случае после поступления кванта $\bar{x}(j)$ оценки \hat{k}_j и $\hat{\tau}_j$ определяются в процессе минимизации по k и $\tau(i)$ функционала

$$F(k, \tau(1), \dots, \tau(j)) = \sum_{i=1}^j \gamma(\bar{x}(i), \tau(i)),$$

где $1 \leq k \leq k_1$, $k - \tau_0 \leq \tau(i) \leq k + \tau_0$, $i = \overline{1, j}$. Для марковской модели ошибок функционал имеет тот же вид, но вместо $\gamma(\bar{x}(i), \tau(i))$ используется $\gamma_1(\bar{x}(i), \tau(i))$.

Л и т е р а т у р а

1. СТИФФЛЕР Дж. Теория синхронной связи. - М.: Связь, 1975. - 487 с.
2. КОЛТУНОВ М.Н., КИРЯКОВ Г.В., ЛАНГУРОВ Э.И. Синхронизация по циклам в цифровых системах связи. - М.: Связь, 1980. - 153 с.
4. БЛЕКУЭЛЛ Д., ГИРШИК М.А. Теория игр и статистических решений. - М.: ИЛ, 1958. - 374 с.

Поступила в ред.-изд. отд.
27 октября 1986 года