

КОРРЕКТИРУЮЩИЕ (i, k, s) -АЛГОРИТМЫ

Н. В. Алипов, И. В. Витенько, А. П. Стахов

Харьков

В работе [1] построена модель АЦП, названная моделью «щупов». Функционирование АЦП на модели «щупов» описывается следующим образом. Аналого-цифровое преобразование осуществляется за i ходов с помощью системы из k щупов. Щупы действуют на отрезке $[0, 1]$. Преобразуемый сигнал $x \in [0, 1]$. Алгоритмы аналого-цифрового преобразования, в течение i ходов управляющие k щупами, движение которых подчиняется условию s , названы (i, k, s) -алгоритмами.

Рассмотрим работу АЦП в дискретные моменты времени $t_1, t_2, \dots, t_i$. t_j -му дискретному моменту времени соответствует момент приложения щупов на j -ом ходе ($j = 1, 2, \dots, i$).

В работах [1, 2] найдены оптимальные (i, k, s) -алгоритмы в предположении, что сигнал x остается неизменным в течение цикла преобразования, т. е. $x(t_1) = x(t_2) = \dots = x(t_i) = x_u$ (x_u — истинное значение преобразуемого сигнала). Указанное предположение является достаточно сильной идеализацией реального процесса аналого-цифрового преобразования. В действительности в процессе преобразования сигнал x может смещаться по отрезку $[0, 1]$.

Целесообразно смещение сигнала x по отрезку $[0, 1]$ классифицировать следующим образом:

- 1) А-смещение;
- 2) В-смещение;
- 3) АВ-смещение (комбинированное смещение).

Под А-смещением x понимается такое смещение сигнала x , когда в некоторые дискретные моменты времени он отличается от своего истинного значения x_u , а в остальные моменты времени сигнал $x = x_u$. А-смещению в реальном АЦП соответствует наложение импульсной помехи на преобразуемый сигнал.

Под В-смещением понимается такое смещение сигнала x , когда истинное значение сигнала имеет тенденцию развития во времени, т. е. в общем случае $x_u(t_1) \neq x_u(t_2) \neq \dots \neq x_u(t_i)$. В-смещение x приводит к возникновению в АЦП динамической погрешности второго рода [3].

Под АВ-смещением понимается сложное смещение сигнала x , представляющее комбинацию А- и В-смещения.

Возникает задача нахождения оптимальных (i, k, s) -алгоритмов, в которых сигнал x подвергается одному из видов смещения. (i, k, s) -алгоритмы, рассматриваемые при условии, что сигнал подвергается А-В-или АВ-смещению, соответственно будем называть $(i, k, s, А)$ -, $(i, k, s, В)$ -, $(i, k, s, АВ)$ -алгоритмами.

По аналогии с теорией избыточных или корректирующих кодов [4] рассматриваемые алгоритмы назовем корректирующими алгоритмами.

В настоящей статье формулируются задачи нахождения оптимальных (i, k, s, A) -алгоритмов и проводится решение одной задачи для частного случая.

Проведем дальнейшую аналогию с корректирующими кодами. Подобно корректирующим кодам для исправления ошибок (i, k, s, A) -алгоритмы предназначены для корректирования случайных импульсных помех на входе АЦП. Различают корректирующие коды для исправления симметричных одиночных, двоичных и т.д. ошибок, созданы корректирующие коды для исправления «пачек» ошибок, описаны коды с исправлением несимметричных или асимметричных ошибок, когда, например, наибольшую вероятность имеют ошибки $1 \rightarrow 0$ [5]. Таким образом, вид кода зависит от характера ошибок. Этот общий вывод справедлив, по-видимому, и для (i, k, s, A) -алгоритмов.

Для более детального изучения (i, k, s, A) -алгоритмов необходимо начать с рассмотрения видов A -смещений сигнала x .

Все A -смещения x разобьем на две группы:

I. СИММЕТРИЧНЫЕ A -СМЕЩЕНИЯ ИЛИ A_1 -СМЕЩЕНИЯ

Под A_1 -смещением x понимаем такое смещение x , когда может иметь место отклонение x влево и вправо от x_u .

II. НЕСИММЕТРИЧНЫЕ СМЕЩЕНИЯ ИЛИ A_2 -СМЕЩЕНИЯ

Под A_2 -смещением x понимаем такое смещение x , когда может иметь место только однонаправленное отклонение x от точки x_u (влево от x_u либо вправо от x_u).

Одиночным A -смещением x длительности l будем называть такое смещение x , когда в течение не более l подряд идущих дискретных моментов времени ($l \leq i$) значение x отличается от истинного значения x_u .

$A(m, l)$ -смещением x назовем смещение x , состоящее из p не более m одиночных A -смещений.

$[i, k, s, A_v(m, l)]$ -алгоритмом назовем (i, k, s) -алгоритм, рассматриваемый при условии, что сигнал x подвергается $A_v(m, l)$ -смещению ($v = 1, 2$).

К $[i, k, s, A_v(m, l)]$ -алгоритму применимо понятие (i, k) -точности алгоритма $\Psi(i, k)$ введенное в работе [1].

Задача нахождения оптимального $[i, k, s, A_v(m, l)]$ -алгоритма формулируется следующим образом:

Задача 1. Среди всего множества $[i, k, s, A_v(m, l)]$ -алгоритма найти оптимальный $[i, k, s, A_v(m, l)]$ -алгоритм, (i, k) -точность которого удовлетворяет условию

$$\Psi(i, k) \geq \Psi'(i, k),$$

где $\Psi'(i, k)$ — (i, k) -точность любого другого $[i, k, s, A_v(m, l)]$ -алгоритма.

$[i, k, 0, A_v(m, l)]$ -алгоритмы

Конкретизируем задачу 1 для простейшего случая $[i, k, s, A_v(m, l)]$ -алгоритмов, характеризующегося следующими условиями: алгоритм состоит из i ходов; имеет в своем распоряжении k щупов, на движение которых по отрезку $[0, 1]$ никаких ограничений не накладывается ($s = 0$); сигнал x подвергается одиночному ($m = 1$) A_v -смещению ($v = 1, 2$) длительности $l = 1$. Согласно приведенным выше обозначениям, рассматриваемые алгоритмы являются $[i, k, 0, A_v(1, 1)]$ -алгоритмами.

Вредное влияние A_j -смещений x состоит в том, что A_j -смещение x , возникшее на j -ом ходе $[i, k, 0, A_j(1, 1)]$ -алгоритма может привести к ложной информации относительно истинного значения x_u . Следует отметить, что не всякое смещение x может привести к ложной информации относительно x_u .

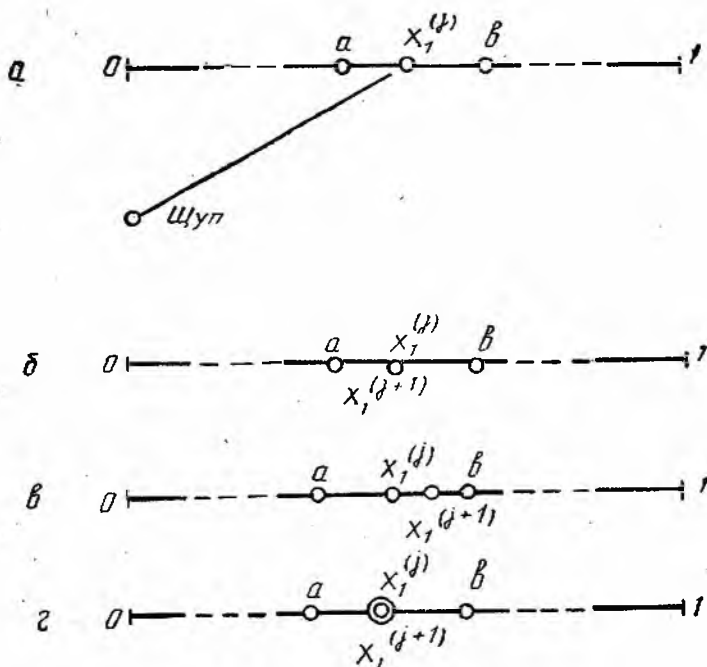


Рис. 1.

Рассмотрим пример. Пусть на j -1-ом ходе корректирующего алгоритма достоверно установлено, что $x_u \in [a, b]$ ($[a, b] \subset [0, 1]$) и на j -ом ходе в распоряжении алгоритма имеется один щуп (рис. 1). На j -ом ходе щуп прикладывается к некоторой точке $x_j^{(j)} \in [a, b]$. Тогда возможно получение от щупа информации двух видов:

- x находится слева от $x_j^{(j)}$;
- x находится справа от $x_j^{(j)}$.

Пусть $x_u \in [a, x_j^{(j)}]$. Тогда информация а) вместе с информацией $x_u \in [a, b]$ дает достоверную информацию $x_u \in [a, x_j^{(j)}]$. Следует отметить, что информация а) может быть получена как в случае отсутствия A -смещения x ($x = x_u$), так и в случае, когда имеют место A -смещения x , которые не выводят сигнал x правее точки $x_j^{(j)}$. В дальнейшем нас будут интересовать только A -смещения x , дающие ложную информацию относительно x_u . Такие A -смещения x будем называть ложными.

Если помеха, действующая в канале связи, настолько искажает передаваемый сигнал, что сигнал y превращается в свое отрицание $\bar{y}$, а затем $\bar{y}$ превращается в свое отрицание $\bar{\bar{y}} = y$, то такое искажение сигнала обнаружить никакими средствами, имеющимися у нас, нельзя. Теория корректирующих кодов берется только с искажениями сигнала, которые приводят к ложной информации. Аналогичная ситуация воз-

никает и в теории корректирующих алгоритмов аналого-цифрового преобразования.

Уточним определение $[i, k, 0, A, (1, 1)]$ -алгоритма: под $[i, k, 0, A, (1, 1)]$ -алгоритмом понимается такой $[i, k, 0]$ -алгоритм, в котором может возникнуть не более одного ложного A -смещения x длительности $l = 1$.

По условию в процессе преобразования может возникнуть не более одного ложного A -смещения x , поэтому стратегия действий до обнаружения ложного A -смещения существенно отличается от стратегии действий после его обнаружения. Стратегия действий до обнаружения ложного A -смещения x строится в предположении, что такое смещение может возникнуть. Задача состоит в том, чтобы получить наибольшее количество информации в предположении, что информация со шупов не всегда является достоверной. Назовем эту стратегию стратегией 1-го рода.

Задача алгоритма после обнаружения A -смещения получить наибольшее количество информации в предположении, что информация со шупов является достоверной. Такую стратегию назовем стратегией 2-го рода.

Задача корректирующего алгоритма — обнаружить и исправить ложное A -смещение x . По условию ложное A -смещение x может возникнуть на любом ходе, неизвестном заранее.

Пусть на j -ом ходе возникло или не возникло ложное A -смещение x и на $j + L$ -м ходе достоверно установлено: было или не было ложное A -смещение x на j -ом ходе. Величину L назовем длиной обнаружения ложного смещения.

Для всех $x_u \in \{0, 1\}$ рассмотрим всевозможные варианты ложных A -смещений, которые могут возникнуть на любом ходе в любую возможную точку $x \in \{0, 1\}$.

Каждому варианту поставим в соответствие длину обнаружения ложного смещения L и выберем среди всех длин максимальную длину L^* . Максимальная длина обнаружения ложного A -смещения L^* (в дальнейшем эту величину будем называть просто длиной обнаружения) является такой же важной характеристикой корректирующего алгоритма, как и (i, k) -точность $\Psi(i, k)$.

Тогда возникает задача оптимизации корректирующего алгоритма по величине L^* .

Задача 2. Среди всех $[i, k, 0, A, (1, 1)]$ -алгоритмов найти $[i, k, 0, A, (1, 1)]$ -алгоритм, удовлетворяющий условию

$$L_{\text{орт}}^* \leq L^*, \quad (1)$$

где $L_{\text{орт}}^*$ — длина обнаружения оптимального $[i, k, 0, A, (1, 1)]$ -алгоритма;

L^* — длина обнаружения любого другого $[i, k, 0, A, (1, 1)]$ -алгоритма.

В общем случае оптимальные по длине обнаружения смещения $[i, k, 0, A, (1, 1)]$ -алгоритмы образуют некоторый подкласс $[i, k, 0, A, (1, 1)]$ -алгоритмов.

Тогда задачу 1 конкретизируем следующим образом и назовем ее задачей 3.

Задача 3. В подклассе оптимальных по длине обнаружения смещения $[i, k, 0, A, (1, 1)]$ -алгоритмов найти оптимальный по точности $[i, k, 0, A, (1, 1)]$ -алгоритм, (i, k) -точность которого удовлетворяет условию

$$\Psi(i, k) \geq \Psi'(i, k),$$

где $\Psi'(i, k)$ — (i, k) -точность любого другого оптимального по длине обнаружения алгоритма:

$[i, k, 0, A_v(1, 1)]$ -алгоритмы, являющиеся решением задачи 3, будем называть дважды оптимальными $[i, k, 0, A_v(1, 1)]$ -алгоритмами.

Не исключено, что (i, k) -точность $[i, k, 0, A_v(1, 1)]$ -алгоритма, являющегося решением задачи 1, может оказаться выше (i, k) -точности дважды оптимального $[i, k, 0, A_v(1, 1)]$ -алгоритма. Тогда представляет интерес зависимость (i, k) -точности оптимального $[i, k, 0, A_v(1, 1)]$ -алгоритма от величины L^* при $L^* > L_{\text{опт}}^*$.

В классе $[i, k, 0, A_v(1, 1)]$ -алгоритмов выделим подкласс $[i, k, 0, A_v(1, 1)]$ -алгоритмов, включающий все $[i, k, 0, A_v(1, 1)]$ -алгоритмы, длина обнаружения которых не превышает L_0^* ($L_0^* > L_{\text{опт}}^*$). Указанный подкласс назовем L_0^* -классом $[i, k, 0, A_v(1, 1)]$ -алгоритмов.

В этом случае возникнет следующая задача.

Задача 4. L_0^* -классе $[i, k, 0, A_v(1, 1)]$ -алгоритмов найти оптимальный по точности $[i, k, 0, A_v(1, 1)]$ -алгоритм, (i, k) -точность которого удовлетворяет условию

$$\Psi(i, k) \geq \psi'(i, k),$$

где $\Psi'(i, k)$ — (i, k) -точность любого алгоритма из L_0^* -класса.

Ясно, что задача 3 является частным случаем задачи 4, когда $L_0^* = L_{\text{опт}}^*$.

Дважды оптимальный $[i, k, 0, A_2(1, 1)]$ -алгоритм

Решим задачу нахождения дважды оптимального алгоритма для $[i, 1, 0, A_2(1, 1)]$ -алгоритмов частного случая $[i, k, 0, A_v(1, 1)]$ -алгоритмов, когда $k = 1$ и сигнал x подвергается A_2 -смещению.

Предположим, что A_2 -смещение x происходит в направлении от x_u к 1. Перед тем, как приступить к решению задачи 3, решим задачу 2.

Пусть существует оптимальный по длине обнаружения $[i, 1, 0, A_2(1, 1)]$ -алгоритм. Сформулируем условия, которым должен удовлетворять оптимальный по длине обнаружения $[i, 1, 0, A_2(1, 1)]$ -алгоритм и найдем для него значение $L_{\text{опт}}^*$.

Пусть перед j -ым ходом ($j = 1, 2, \dots, i-1$) выделен отмеченный отрезок $[a, b]$, содержащий x_u , и известно, что ложное A_2 -смещение x не имело место. Пусть на j -ом ходе щуп прикладывается к точке $x_1^{(j)} \in [a, b]$ (рис. 1).

Тогда возможно получение от щупа информации двух видов:

- x находится слева от $x_1^{(j)}$;
- x находится справа от $x_1^{(j)}$.

Если получена информация а), то, сопоставляя ее с информацией о том, что $x_u \in [a, b]$ и в системе может иметь место только A_2 -смещение x , можно сразу после первого хода сделать вывод, что $x_u \in [a, x_1^{(j)}]$ и на j -ом ходе не было ложного A_2 -смещения x .

Таким образом, в случае а) длина обнаружения смещения $L = 0$. Если получена информация б) то на j -ом ходе могло быть ложное A_2 -смещение x с отрезка $[a, x_1^{(j)}]$ (если $x_u \in [a, x_1^{(j)}]$) в область, лежащую справа от $x_1^{(j)}$, и могло не быть ложного A_2 -смещения x , если $x_u \in [x_1^{(j)}, b]$. В последнем случае на j -ом ходе невозможно установить было ли A_2 -смещение x , поэтому $L > 0$. $j+1$ -й ход $[i, 1, 0, A_2(1, 1)]$ -алгоритма состоит в том, что щуп прикладывается к некоторой точке $x_1^{(j+1)}$. Возможны три случая, указанные на рис. 1, б; 1, в; 1, г:

- 1) $x_1^{(j+1)} < x_1^{(j)}$;
- 2) $x_1^{(j+1)} > x_1^{(j)}$;
- 3) $x_1^{(j+1)} = x_1^{(j)}$.

Пусть в первом случае получена информация: $x > x_1^{(j+1)}$. Сопоставляя информацию с информацией $x > x_1^{(j)}$, можно сделать два предположения: либо $x_u \in [x_1^{(j+1)}, x_1^{(j)}]$ и на j -ом ходе произошло ложное A_2 -смещение x , либо $x_u \in [x_1^{(j)}, b]$ и ложного A_2 -смещения x не было. Из этих предположений нельзя достоверно установить, было ли A_2 -смещение x на j -ом ходе и, следовательно, для случая 1) $L > 1$.

Пусть во втором случае получена информация $x < x_1^{(j+1)}$. Сравнивая эту информацию с информацией $x > x_1^{(j)}$, можно высказать два предположения: либо $x_u \in [a, x_1^{(j)}]$ и на j -ом ходе было ложное A_2 -смещение x , либо $x_u \in [x_1^{(j)}, x_1^{(j+1)}]$ и ложного A_2 -смещения x на j -ом ходе не было. В этом случае также нельзя установить, было ли ложное A_2 -смещение x на j -ом ходе, поэтому $L > 1$.

Рассмотрим третий случай $x_1^{(j+1)} = x_1^{(j)}$. Пусть получена информация $x > x_1^{(j+1)}$. Сопоставляя эту информацию с информацией $x > x_1^{(j)} = x_1^{(j+1)}$, можно сделать достоверный вывод, что на j -ом ходе не было ложного A_2 -смещения x и $x_u \in [x_1^{(j)}, b]$.

Пусть получена информация $x < x_1^{(j+1)}$. Сопоставляя эту информацию с информацией $x > x_1^{(j)} = x_1^{(j+1)}$, можно сделать единственный достоверный вывод, что на j -ом ходе произошло ложное A_2 -смещение x и, следовательно, $x_u \in [a, x_1^{(j)}] = x_1^{(j+1)}$.

Для третьего случая $L = 1$. Это значение L и является $L_{\text{опт}}$ для оптимального по длине обнаружения $[i, 1, 0, A_2(1, 1)]$ -алгоритма.

Действие оптимального по длине обнаружения $[i, 1, 0, A_2(1, 1)]$ -алгоритма на отрезке $[0, 1]$ состоит в следующем: 1-й ход: шуп прикладывается к некоторой точке $x_1^{(1)}$ отрезка $[0, 1]$; 2-й ход: а) если $x < x_1^{(1)}$, то 2-й ход есть 1-й ход $[i-1, 1, 0, A_2(1, 1)]$ -алгоритма, примененного к отрезку $[0, x_1^{(1)}]$; б) если $x > x_1^{(1)}$, то шуп прикладывается к точке $x_2^{(1)} = x_1^{(1)}$ и в результате двух ходов выделяется отмеченный отрезок $[a, b]$, содержащий x_u , и устанавливается, было ли ложное A_2 -смещение x . Если было, то стратегия действий изменяется на стратегию 2-го рода. Если не было, то 3-й и 4-й ходы алгоритма являются 1-м и 2-м ходами $[i-2, 1, 0, A_2(1, 1)]$ -алгоритма, примененного к отрезку $[a, b]$ и т. д.

В классе оптимальных по длине обнаружения $[i, 1, 0, A_2(1, 1)]$ -алгоритмов найдем оптимальный по точности алгоритм. (i, k) -точность дважды оптимального $[i, 1, 0, A_2(1, 1)]$ -алгоритма обозначим через $\Psi(i, 1)$.

Для частных случаев функции $\Psi(i, 1)$ имеют следующие равенства:

$$\Psi(0, 1) = 1; \quad (1)$$

$$\Psi(1, 1) = 1; \quad (2)$$

$$\Psi(2, 1) = 2. \quad (3)$$

Соотношение (1) тривиально. Соотношение (2) вытекает из того факта, что за один ход корректирующего алгоритма в предположении, что на этом ходе может быть, а может и не быть ложное A_2 -смещение x , в наиболее неблагоприятном случае невозможно получить достоверной информации относительно x_u .

Легко доказать, что дважды оптимальный $[2, 1, 0, A_2(1, 1)]$ -алгоритм, действующий на отрезке $[0, 1]$, состоит в следующем:

$$1\text{-й ход: } x_1^{(1)} = \frac{1}{2}; \quad 2\text{-й ход: } x_2^{(2)} = \frac{1}{2},$$

откуда $\Psi(2, 1) = 2$.

Сконструируем дважды оптимальный $[i, 1, 0, A_2(1, 1)]$ -алгоритм для любого i . Предположим, что существует дважды оптимальный $(i, 1, 0, A_2(1, 1))$ -алгоритм, делящий отрезок $[0, 1]$ на равные части. Тогда $\Psi(i, 1)$ представляет собой число частей, на которые делится отрезок $[0, 1]$ дважды оптимальным $[i, 1, 0, A_2(1, 1)]$ -алгоритмом.

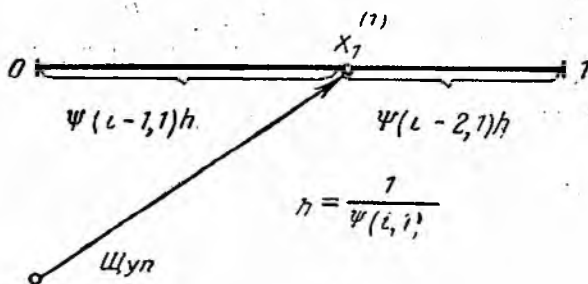


Рис. 2.

Обозначим $h = \frac{1}{\Psi(i, 1)}$. Пусть первый ход дважды оптимального $[i, 1, 0, A_2(1, 1)]$ -алгоритма состоит в прикладывании щупа к точке $x_1^{(1)}$ (рис. 2).

I. Пусть после первого хода оказалось, что $x < x_1^{(1)}$, откуда следует, что

1) $x_u \in [0, x_1^{(1)}]$;

2) ложного A_2 -смещения x не было.

В этом случае имеем отрезок $[0, x_1^{(1)}]$ один щуп и $(i-1)$ -ходов. Применяя к отрезку $[0, x_1^{(1)}]$ дважды оптимальный $[i-1, 1, 0, A_2(1, 1)]$ -алгоритм, можно разделить данный отрезок $[0, x_1^{(1)}]$ на $\Psi(i-1, 1)$ -частей длиной h' . Для оптимального по (i, k) -точности алгоритма должно выполняться условие

$$h' = \frac{x_1^{(2)}}{\Psi(i-1, 1)} \leq h = \frac{1}{\Psi(i, 1)}. \quad (4)$$

II. Пусть после первого хода оказалось, что $x > x_1^{(1)}$. Тогда на втором ходе щуп прикладывается к точке $x_1^{(2)} = x_1^{(1)}$.

а) Если $x > x_1^{(2)}$, то $x_u \in [x_1^{(2)}, 1]$ и ложного A_2 -смещения не было. Применяя к отрезку $[x_1^{(2)}, 1]$ дважды оптимальный $[i-2, 1, 0, A_2(1, 1)]$ -алгоритм, можно разделить отрезок $[x_1^{(2)}, 1]$ на $\Psi(i-2, 1)$ -частей длины h . Отсюда получаем условие для дважды оптимального $[i, 1, 0, A_2(1, 1)]$ -алгоритма.

$$1 - x_1^{(1)} = \Psi(i-2, 1)h. \quad (5)$$

б) Если $x < x_1^{(2)}$, то $x_u \in [0, x_1^{(2)}]$ и имело место ложное A_2 -смещение x на первом ходе. Так как по условию ложного A_2 -смещения x на последующих ходах возникнуть не может, то к отрезку $[0, x_1^{(2)}]$ применяется новая стратегия. Легко показать, что оптимальная стратегия второго рода состоит в том, что к отрезку $[0, x_1^{(2)}]$ применяется $A(i-2, 1, 0)$ -алгоритм [1]. Число равных частей, на которое может быть разделен отрезок $[0, x_1^{(2)}]$ $A(i-2, 1, 0)$ -алгоритмом, обозначим $\phi((i-2, 1)) = -2_{i-2}$.

Ясно, что для дважды оптимального $[i, 1, 0, A_2(1, 1)]$ -алгоритма длина наименьшего отрезка h'' , в который заключается любое $x_u \in [0, x_1^{(2)}]$ $A(i-2, 1, 0)$ -алгоритмом, не должна превышать h , т. е.

$$h'' = \frac{x_1^{(2)}}{\Psi(i-2, 1)} \leq h = \frac{1}{\Psi(i, 1)}. \quad (6)$$

Соотношения (1) — (6) являются условиями, которым должен удовлетворять дважды оптимальный $[i, 1, 0, A_2(1, 1)]$ -алгоритм. Рассмотрим следующую функцию:

$$\begin{aligned} \Psi(i, 1) &= \Psi(i-1, 1) + \Psi(i-2, 1); \\ \Psi(0, 1) &= 1; \quad \Psi(1, 1) = 1. \end{aligned} \quad (7)$$

На основании приведенных выше интуитивных рассуждений и функции (7) определим следующий $[i, 1, 0, A_2(1, 1)]$ -алгоритм:

1-й ход: $x_1^{(1)} = \Psi(i-1, 1)h$;

2-й ход: а) если $x < x_1^{(1)}$, то 2-й ход есть 1-й ход $[i-1, 1, 0, A_2(1, 1)]$ -алгоритма, примененного к отрезку $[0, x_1^{(1)}]$;

б) если $x > x_1^{(1)}$, то щуп прикладывается к точке $x_1^{(2)} = x_1^{(1)}$.

3-й ход: а) если $x > x_1^{(2)}$, то 3-й ход есть 1-й ход $[i-2, 1, 0, A_2(1, 1)]$ -алгоритма, примененного к отрезку $[x_1^{(2)}, 1]$;

б) если $x < x_1^{(2)}$, то 3-й ход есть 1-й ход $A(i-2, 1, 0)$ -алгоритма, примененного к отрезку $[0, x_1^{(2)}]$.

Докажем, что данный алгоритм реализует $(i, 1)$ точность $\Psi(i, 1)$, задаваемую (7).

Для этого достаточно показать, что

$$\Phi(i-1, 1) \geq \Psi(i, 1). \quad (8)$$

Докажем это соотношение индукцией по i . Очевидно, что

$$\Psi(1, 1) \leq \varphi(0, 1) = 1;$$

$$\Psi(2, 1) \leq \varphi(1, 1) = 2.$$

Пусть

$$\Psi(i-1, 1) \leq 2^{i-2};$$

$$\Psi(i-2, 1) \leq 2^{i-3}, \quad (9)$$

тогда

$$\Psi(i, 1) \leq 2^{i-2} + 2^{i-3}. \quad (10)$$

Легко заметить, что неравенство (10) равносильно неравенству $\Psi(i, 1) < 2^{i-2} + 2^{i-3} \leq 2^{i-2} + 2^{i-2} < 2^{i-1} = \varphi(i-1, 1)$. Сформулированный алгоритм назовем $A[i, 1, 0, A_2(1, 1)]$ -алгоритмом.

Индукцией по i легко доказать следующую теорему.

Теорема 1.

$A[i, 1, 0, A_2(1, 1)]$ -алгоритм является дважды оптимальным $[i, 1, 0, A_2(1, 1)]$ -алгоритмом.

В таблице приведены значения для функции $\Psi(i, 1)$ при $i = 0, 1, 2, \dots, 10$.

i	0	1	2	3	4	5	6	7	8	9	10
$\Psi(i, 1)$	1	1	2	3	5	8	13	21	34	55	89

Интересно отметить, что числовая последовательность $\Psi(i, 1)$ представляет собой ни что иное, как ряд Фибоначчи [6], а $A[i, 1, 0, A_2(1, 1)]$ -алгоритм до обнаружения ложного A_2 -смещения совпадает со стратегией фибоначчиевого поиска [7].

Пример 1. Рассмотрим $A[5, 1, 0, A_2(1, 1)]$ -алгоритм, действующий на отрезке $[0, 8]$ (рис. 3, а).

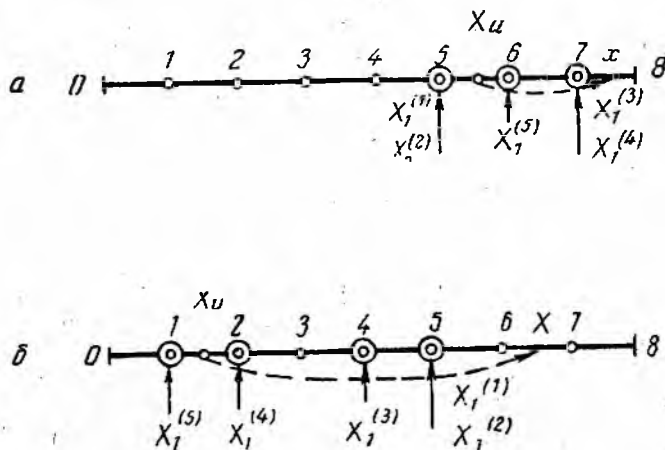


Рис. 3.

а. Пусть $x_u = 5,5$ и на 3-м ходе возникает ложное A_2 -смещение.

1-й ход: $x_1^{(1)} = 5 \rightarrow x > 5$;

2-й ход: $x_1^{(2)} = 5 \rightarrow x > 5$, следовательно, $x_u \in [5, 8]$.

3-й ход: (имеет место ложное A_2 -смещение) $x_1^{(3)} = 7$;

4-й ход: $x_1^{(4)} = 7 \rightarrow x < 7$, следовательно, $x_u \in [5, 7]$;

5-й ход: $x_1^{(5)} = 5 \rightarrow x < 6$, следовательно, $x_u \in [5, 6]$;

б. Пусть $x_u = 1,5$ и на 1-м ходе возникает ложное A_2 -смещение (рис. 3, б).

1-й ход: $x_1^{(1)} = 5$ (имеет место ложное A_2 -смещение).

2-й ход: $x_1^{(2)} = 5 \rightarrow x < 5$, следовательно, $x \in [0, 5]$ и на первом ходе имело место ложное A_2 -смещение.

Стратегия действий на последующих ходах — стратегия 2-го рода. За оставшиеся три хода, действуя на отрезке $[0, 5]$ алгоритмом поразрядного кодирования, можно было бы разбить отрезок $[0, 5]$ на 2^3 равных частей, т. е. заключить x_u в отрезок, длиной $\frac{5}{8} < h = 1$. Так как в таком точном определении x нет необходимости, то на последующих ходах можно действовать другим алгоритмом, отличным от алгоритма поразрядного кодирования. Один из возможных алгоритмов состоит в следующем:

3-й ход: $x_1^{(3)} = 4 \rightarrow x < 4$;

4-й ход: $x_1^{(4)} = 2 \rightarrow x < 2$;

5-й ход: $x_1^{(5)} = 1 \rightarrow x > 1$

и следовательно, $x_u \in [1, 2]$.

Сравним $A[5, 1, 0, A_2(1, 1)]$ -алгоритм с $A[5, 1, 0]$ -алгоритмом (алгоритмом поразрядного кодирования) [1]. Если x_u находится в окрестности точки 0 и на первом ходе ($x_1^{(1)} = \frac{1}{2}$) имело место ложное A_2 -смещение x ,

то на всех последующих ходах поиск будет осуществляться на отрезке $\left[\frac{1}{2}, 1\right]$. При этом будет допущена большая ошибка в определении x_n . Можно показать, что (i, k) -точность такого алгоритма при условии действия A_2 -смещения x не превышает двух, что существенно меньше (i, k) -точности корректирующего алгоритма в примере 1.

ЛИТЕРАТУРА

1. И. В. Витенько, А. А. Волков, А. П. Стахов. Об оптимальных алгоритмах функционирования АЦП. Сб. «Вопросы радиоэлектроники», серия VII, «Электронная вычислительная техника», вып. 2, 1966.
2. И. В. Витенько, А. А. Волков, А. П. Стахов. Новые классы (i, k, s) -алгоритмов функционирования АЦП. Сб. «Вопросы радиоэлектроники», серия VII, «Электронная вычислительная техника», вып. 2, 1967.
3. В. Н. Хлистунов. Основы цифровой электроизмерительной техники. Изд-во «Энергия», 1966.
4. У. Петерсон. Коды, исправляющие ошибки. Изд-во «Мир», 1964.
5. У. Коц. Коды с исправлением ошибок и их реализация в цифровых системах. В кн. «Методы введения избыточности для вычислительных систем», пер. с англ. Изд-во «Советское радио», 1966.
6. Н. Н. Воробьев. Число Фибоначчи. Изд-во «Наука», 1964.
7. Д. Уайлд. Методы поиска экстремума. Изд-во «Наука», 1967.