УДК 656.09

ПОЛУЧЕНИЕ ПСЕВДОСЛУЧАЙНЫХ ВЕЛИЧИН МЕТОДАМИ ЛЕМЕРА

Волхонский А.Н.[1], Бизюкова Е.Е.[2]
[1]*ФГБОУ ВО «Самарский государственный технический университет», Самара, e-mail: avolhonskij34@gmail.com*
[2]*ФГБОУ ВО «Самарский государственный технический университет», Самара, e-mail: lizaveta5.6@mail.ru*

Аннотация. Направления автоматизации технологических процессов является одним из наиболее активно развивающихся в современной промышленности. Управление процессами осуществляется на базе ПЛК и помощью некоторой рассчитанной и запрограммированной логики. Расчет данной логики управления является одним из наиболее важных этапов разработки автоматизированных систем, наряду с выбором средств автоматизации и осуществлением установки и настройки оборудования. Логика управления осуществляется при помощи регуляторов, позволяющих изменять параметры происходящих процессов для достижения наиболее оптимальных результатов управления процессом. Настройка параметров регуляторов осуществляется при помощи математической модели, построенной на основе изучения свойств объекта управления. Для произведения наиболее эффективной настройки при расчете параметров регулирования необходимо учитывать наличие действующих на объект внешних воздействий. Внешние воздействия представляют собой белый шум, который может случайным образом воздействовать на объект управления. Воздействие белого шума моделируется при помощи генератора псевдослучайных чисел, которые используя некоторые алгоритмы обработки ключевых значений, получают величины с низким коэффициентом корреляции. В данной статье будут рассмотрены методы Лемера для получения псевдослучайных чисел, составлены блок-схемы реализации алгоритмов, написан код программы на языке java script.

Ключевые слова: автоматизация, белый шум, псевдослучайные числа, блок-схема, java script.

OBTAINING PSEUDORANDOM VALUES BY LEHMER METHODS

Volkhinskij A.N.[1], Bizykova E.E.[2]
[1]*Samara State Technical University, Samara, e-mail: avolhonskij34@gmail.com*
[2]*Samara State Technical University, Samara, e-mail: lizaveta5.6@mail.ru*

Annotation. The direction of automation of technological processes is one of the most actively developing in modern industry. The processes are controlled on the basis of a PLC and using some calculated and programmed logic. The calculation of this control logic is one of the most important stages of the development of automated systems, along with the choice of automation tools and the installation and configuration of equipment. The control logic is carried out with the help of regulators that allow changing the parameters of the ongoing processes to achieve the most optimal results of process control. Adjustment of the parameters of the regulators is carried out using a mathematical model based on the study of the properties of the control object. To make the most effective adjustment, when calculating the control parameters, it is necessary to take into account the presence of external influences acting on the object. External influences are white noise, which can randomly affect the control object. The effect of white noise is modeled using a pseudo-random number generator, which, using some algorithms for processing key values, obtain values with a low correlation coefficient. In this article, we will consider Lehmer's methods for obtaining pseudo-random numbers, draw up flowcharts for implementing algorithms, and write the program code in java script.

Keywords: automation, white noise, pseudorandom numbers, flowchart, java script.

При исследовании различных динамических систем часто используются имитационные модели. Согласно одному из определений, имитационная модель это определенный алгоритм, описывающий функционирование системы во времени, реализация которого осуществляется с помощью соответствующей машинной программы.

При невозможности описать весь спектр воздействующих на систему факторов в модель вводится элемент случайности. Таким образом, для работы моделирующей программы необходима генерация случайных чисел.

Компьютерная программа, работающая по определенному алгоритму, может генерировать только псевдослучайные числа. Результаты работы различных алгоритмов проверяются статистическими тестами. Для генерации случайных чисел в программу подставляются случайным образом начальные условия. Для моделирования могут быть необходимы случайные величины с самыми разными характеристиками, но обычно, сначала генерируется базовая последовательность случайных чисел. Совокупность независимых, равномерно распределенных на отрезке [0,1] случайных величин Ri , (i=0,1…) называется последовательностью базовых случайных чисел. Гистограмма, показывающее равномерное распределение величин представлена на рисунке 1.

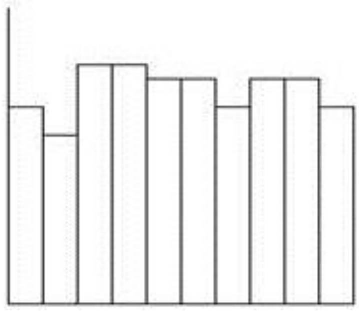

Рисунок 1 – Гистограмма равномерного распределения величин

Алгоритм Лемера

Американский исследователь Д. Лемер предложил метод генерации базовой последовательности, представимый в виде формулы или реккурентного соотношения.

Алгоритмы Лемера получения псевдослучайных чисел Ri+1 имеют достаточно простой вид: $R_{i+1} = F(R_i)$, i=0,1, …

Начальное число R_0 задано, а все последующие числа R_1, R_2, … вычисляются по одной и той же формуле. Заметим, что метод серединных квадратов, рассмотренный выше, также имеет аналогичный вид, но вместо аналитического задания функции y=F(x) была указана совокупность операций, которые надо проделать над аргументом x, чтобы получить значение y.

Рассмотрим, какие требования необходимо предъявлять к функции F(x), чтобы последовательность удовлетворяла требованиям, предъявляемым к базовой последовательности, т.е. числам, равномерно распределенным на отрезке [0, 1]

Рассмотрим пример, позволяющий понять, в чем состоит одна из основных трудностей при выборе F(x). Во-первых, аргумент и значение функции не должны выходить за пределы отрезка [0, 1]. Но это не все. Произвольная функция, например, график которой показан на рис.2 а) не подходит для генерации случайных чисел, поскольку все точки будут сконцентрированы на кривой, а настоящие случайные точки должны равномерно заполнять весь единичный квадрат.

Таким свойством y = {g·x}, где g – очень большое число. Символ·обладает, например, функция фигурных скобок {*} означает дробную часть числа *.

На рис. 2,б-в) построен график функции y = {g·x}, при не очень больших числах они равны в случае б) g=5, и в) g=20.

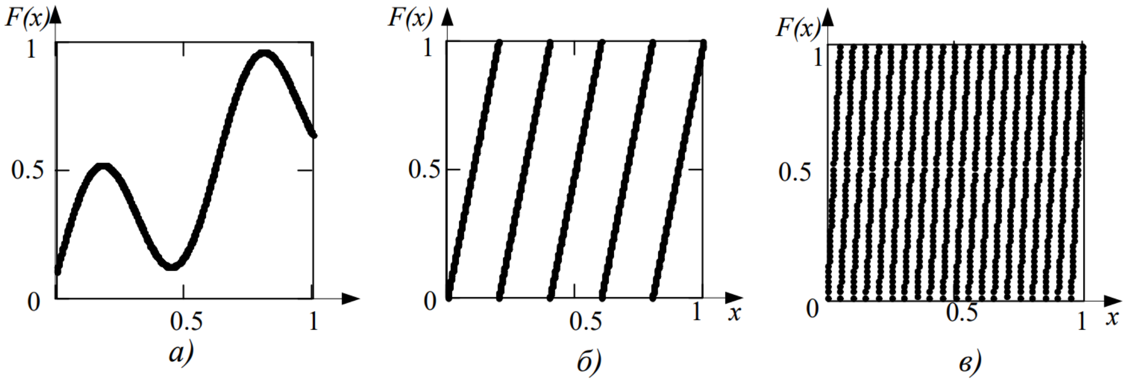

Рисунок 2 – функции F(x), использующиеся в алгоритме Лемера

Блок-схема данного алгоритма представлена на рисунке 3.

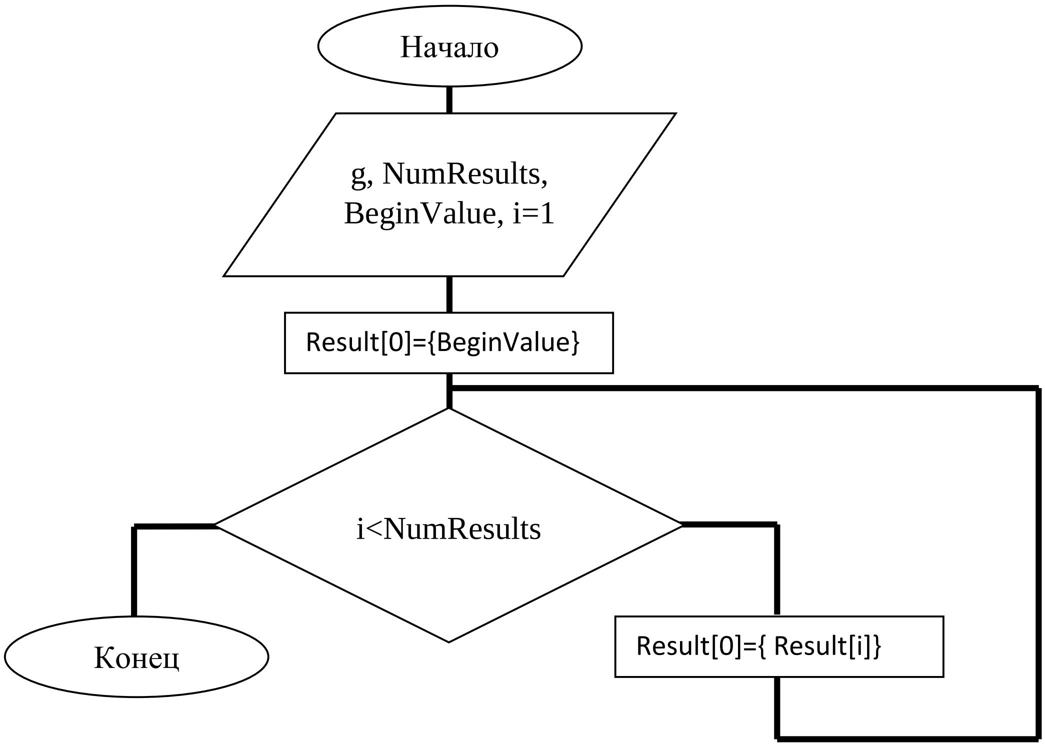

Рисунок 3 – Блок-схема реализации алгоритма Лемера

Программная реализация данного метода на языке программирования java script представлена на рисунке 4.

```
<html>
    <head>
        <script type="text/javascript">
            let g = 927;
            let BeginValue = 0.341;
            var ResultTable = [];
            var NumResults = 100;
            ResultTable[0] = (g * BeginValue - (g * BeginValue).toFixed(0)).toFixed(3);
            for(var i = 1 ; i<NumResults; i++){
                ResultTable[i] = g * ResultTable[i-1] - (g * ResultTable[i-1]).toFixed(0);
                if (ResultTable[i] < 0){
                    ResultTable[i] = ResultTable[i] + 1;
                }
                ResultTable[i] = ResultTable[i].toFixed(3);
            }
            console.log(ResultTable);
        </script>
    </head>
</html>
```

Рисунок 4 – Реализация метода на языке программирования java script

Конгруэнтный мультипликативный метод Лемера

В настоящее время, почти все стандартные библиотечные программы вычисления случайных базовых чисел основаны на конгруэтном методе, или методе сравнений.

Основная формула мультипликативного конгруэнтного метода имеет вид:

$$X_{i+1} = aX_i(\text{mod } m),$$

где a и m — неотрицательные целые числа.

Согласно этому выражению, мы должны взять последнее случайное число R_i , умножить его на постоянный коэффициент *a* и взять модуль полученного числа по m (разделить aXi на m и остаток считать как Xi+1).

Выбираются *a*, X_0 и *m* так, чтобы обеспечить максимальную длину (или, как говорят, период) неповторяющейся последовательности Xi и минимальную корреляцию между генерируемыми числами. В результате получается последовательность псевдослучайных чисел равномерно распределенных на интервале от 0 до m. Для того чтобы получить базовую последовательность, нужно разделить все числа последовательности X_i на m. $R_i = X_i/m$.

Блок-схема данного алгоритма представлена на рисунке 5.

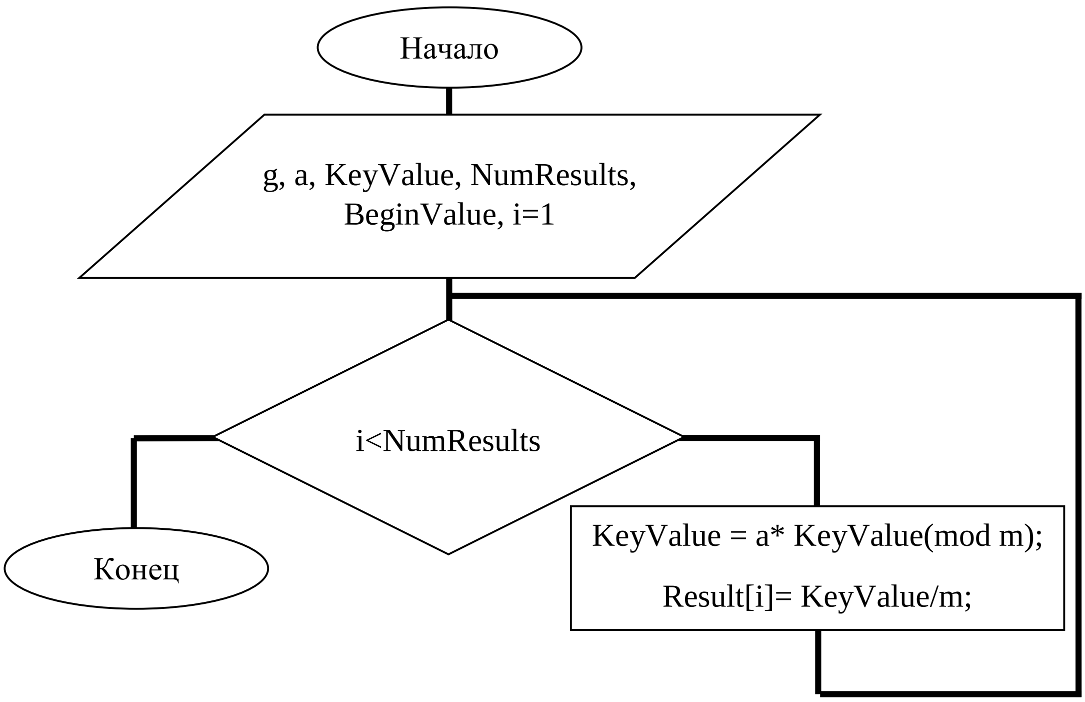

Рисунок 5 – Блок-схема конгруэнтного мультипликативного метода Лемера

Программная реализация данного метода на языке программирования java script представлена на рисунке 6.

```
<html>
    <head>
        <script type="text/javascript">
            var g = 237;
            var m = 131;
            var KeyValue = 472;
            var ResultTable = [];
            var NumResults = 100;
            for(var i = 0 ; i < NumResults; i++){
                KeyValue = (g*KeyValue - Math.floor(g*KeyValue/m) * m);
                ResultTable[i] = (KeyValue / m).toFixed(3);
            }
            console.log(ResultTable);
        </script>
    </head>
</html>
```

Рисунок 6 – Реализация конгруэнтного мультипликативного метода Лемера на языке программирования java script

Список литературы:

1. Линейный конгруэнтный генератор псевдослучайных чисел[Электронный ресурс]// Интуит национальный открытый университет// URL: https://intuit.ru/studies/courses/691/547/lecture/12383?page=2
2. В.А. РУМЯНЦЕВА ПРАКТИКУМ ПО МОДЕЛИРОВАНИЮ СИСТЕМ [Электронный ресурс]// Страничка Валентины Румянцевой// URL: http://valaru.narod.ru/work/metodichka.pdf
3. Генерация случайных чисел [Электронный ресурс]//old.shatalov.su// URL: http://oldshatalov.ghost17.ru/ru/articles/algorithms/random_0.html