

УДК 519.67; 004.896

В.А. Сычев

РЕАЛИЗАЦИЯ ДИСКРЕТНЫХ НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ С ХАОТИЧЕСКИМИ РЕЖИМАМИ НА АРИФМЕТИКЕ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

Рассматривается проблема практической реализации нелинейных систем с хаотической динамикой для целенаправленной генерации хаотических последовательностей в цифровых устройствах. Исследуется возможность применения в данной задаче арифметики с фиксированной запятой для обеспечения идентичности результатов, полученных на различных программно-аппаратных платформах. Описывается реализация логистического отображения, проводится анализ полученных результатов. Предлагается применение полученных результатов для решения различных задач из области мобильной робототехники.

Введение

В последние три десятилетия возрос интерес исследователей к системам, демонстрирующим хаотические режимы работы. Такие известные ученые, как И. Пригожин, Э. Лоренц, а также Г. Хакен, В. Эбелинг и др., показали, что новый порядок (самоорганизация) в сложной системе возникает через динамический хаос – хаотический режим функционирования сложной системы [1, 2].

Новые знания о хаотической динамике нашли свое применение для решения ряда практических задач, которые можно подразделить на две основные группы. Первая – это задача стабилизации нелинейных динамических систем (НДС), включая и сложные НДС, т. е. предотвращения возникновения нежелательных хаотических колебаний в них [3]. Вторая задача обратна первой и состоит в целенаправленной генерации хаотических колебаний. Вслед за теоретическими исследованиями появляются практические примеры использования знаний о хаотических режимах НДС для решения инженерных задач [4]. Методы стабилизации НДС нашли применение для управления электромеханическими приводами, оптическими системами, нелинейными электронными цепями. Хаотические процессы используются в задачах прогнозирования, обработки и защиты информации, управления и планирования [2–7].

Задача целенаправленной генерации хаотических колебаний возникает и в аналоговой, и в цифровой технике. В то время как в аналоговых электронных приборах хаотические сигналы генерируются специальными устройствами, в числе которых наиболее известна схема Чуа [8], в вычислительной технике хаотические последовательности генерируются путем решения одного из известных уравнений или системы уравнений с параметрами, обеспечивающими хаотическую динамику [4, 5].

Важным условием при генерации хаотических последовательностей, особенно в задачах коммуникации, является точность вычислений. Недостаточно высокая точность приводит к постепенному переходу НДС из хаотического в периодический режим либо вовсе препятствует возникновению хаотического режима. Минимально необходимый уровень точности определяется для каждой НДС в отдельности.

Существует множество способов представления в памяти ЭВМ вещественных чисел и алгоритмов выполнения вычислений. В результате вычисления, произведенные на различных платформах, могут давать неодинаковый результат. Следовательно, важна переносимость решений на различные программно-аппаратные платформы, также называемая кроссплатформенностью.

Одним из способов достижения идентичности результатов на разных платформах является стандартизация алгоритмов представления и обработки данных. Однако в процессе разработки кроссплатформенных алгоритмов нередко возникает необходимость в использовании программно-аппаратных платформ, не соответствующих существующим стандартам. Напри-

мер, большинство микропроцессоров, выпускаемых в мире, не имеют аппаратной поддержки вычислений с плавающей запятой. Следовательно, алгоритм работы с числами с плавающей запятой реализуется программным путем. Чаще всего программная реализация ориентирована на достижение высокой производительности в ущерб точности вычислений и переносимости программ.

Общепринятым подходом к достижению требуемой точности и переносимости разработок является применение арифметики с фиксированной запятой [9] (в англоязычной литературе используется термин *fixed-point arithmetic*). Алгоритм генерации хаотических последовательностей, реализованный на целочисленной арифметике, позволит получить результат, воспроизводимый на различных программно-аппаратных платформах с требуемой точностью. Помимо прочего, такой алгоритм обеспечит большую скорость генерации хаотических последовательностей.

Целью данной работы является исследование моделей дискретных НДС с хаотическими режимами, реализованных на арифметике с фиксированной запятой, для их использования в системах управления робототехническими аппаратами.

1. Свойства дискретных нелинейных систем с хаотической динамикой

В отличие от случайных процессов динамику системы принято называть хаотической, если на нее не оказывают влияние случайные или непредсказуемые величины и параметры. Помимо этого, хаотическая динамика отличается рядом особых свойств, в числе которых чувствительность к начальным условиям, потеря информации о начальных условиях, образование фрактальных структур, способность к самосинхронизации [5]. Именно эти свойства хаотической динамики находят применение в инженерных задачах [10].

Хаотические режимы работы обнаруживаются в различных НДС с непрерывным и дискретным временем. К числу широко известных и всесторонне исследованных НДС с непрерывным временем относятся аттракторы Лоренца, Ресслера и др., представляющие собой систему дифференциальных уравнений, решение которой с определенными параметрами позволяет наблюдать хаотическую динамику [5]. Данные аттракторы могут реализовываться на аналоговой элементной базе и использоваться как источник хаотических сигналов. Однако с точки зрения практической реализации НДС в качестве источника хаотических последовательностей наибольший интерес представляют дискретные системы как не требующие дополнительных вычислительных затрат и дополнительных устройств для получения хаотических последовательностей.

Одним из примеров систем с хаотической динамикой является логистическое отображение [10]

$$x_i = a - x_{i-1}^2. \quad (1)$$

Другой вариант дискретной НДС с хаотической динамикой представлен аттрактором Энона [5]:

$$\begin{aligned} x_{i+1} &= 1 + y_i - ax_i^2; \\ y_{i+1} &= bx_i. \end{aligned} \quad (2)$$

Хаотические режимы наблюдаются и в такой дискретной НДС, как кубическое отображение (отображение Холмса), что также описано в [5]:

$$\begin{aligned} x_{i+1} &= y_i; \\ y_{i+1} &= -bx_i + dy_i - y_i^3. \end{aligned} \quad (3)$$

Параметры данных НДС, при которых наблюдаются хаотические режимы работы, и измеренные показатели Ляпунова (старшие экспоненты Ляпунова) для каждой из них приведены в табл. 1.

Таблица 1

Дискретные отображения и их характеристики

| Отображение | Параметры хаотизации | Показатель Ляпунова |
|-----------------|----------------------|---------------------|
| Логистическое | $a = 1,99$ | 0,6607 |
| Аттрактор Энона | $a = 1,4; b = 0,3$ | 0,4192 |
| Кубическое | $b = 0,2; d = 2,77$ | 0,5323 |

Для вычисления показателя Ляпунова был использован алгоритм М.Т. Розенштейна [11], реализованный в системе компьютерной математики (СКМ) Matlab [12].

Исследуя отображение (1), можно определить, что ряд значений x_i , вычисленный несколько раз на компьютере при равных начальных условиях, но с использованием различных микропроцессоров, компиляторов, языков программирования, может различаться. Так как отображение (1) в соответствии с общим свойством хаотических систем обладает высокой чувствительностью к начальным условиям, малейшее различие в результатах вычислений в итоге приводит к значительному расхождению результатов.

Иллюстрацией может служить ряд значений, полученных итерационным решением формулы (1) с начальными условиями $x_1 = 0,5$ и $a = 1,99$. В первом случае значения были получены с помощью программы на языке Си, откомпилированной компилятором *icc*, во втором случае – компилятором *gcc* с параметрами, установленными по умолчанию при инсталляции. В остальном условия исполнения программ оставались идентичными. Вычисления проводились с использованием чисел с плавающей запятой в формате *double*. На графике (рис. 1, а) показана разность между полученными в обоих случаях значениями.

На 59-м шаге разница достигает уровня, при котором значения полностью расходятся (рис. 1, б).

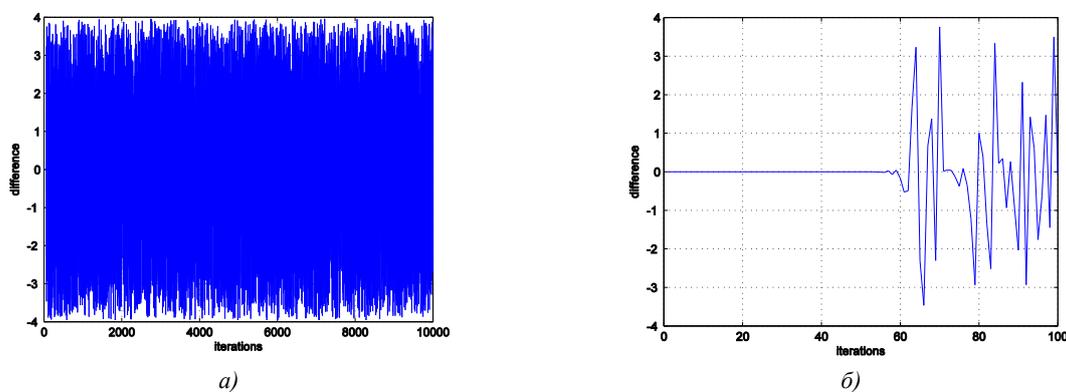


Рис. 1. Разность между значениями, полученными с использованием компиляторов *icc* и *gcc*: а) на всем диапазоне значений; б) начало расхождения в значениях, сгенерированных с использованием данных компиляторов

Таким образом, при использовании того или иного отображения в качестве источника хаотических последовательностей на различных программно-аппаратных платформах необходимо принять дополнительные меры, обеспечивающие идентичность получаемых результатов.

2. Особенности применения арифметики с фиксированной запятой

Современные микропроцессоры, применяемые в персональных компьютерах, позволяют на аппаратном уровне выполнять вычисления с плавающей запятой. Способы представления и обработки таких чисел описываются стандартом IEEE754 [9, 13], однако не все компиляторы точно соответствуют требованиям данного стандарта, что объясняет расхождение результатов вычислений, показанное на рис. 1.

В то же время в системах управления мобильными роботами применяются энергоэкономичные микропроцессоры с ограниченными вычислительными ресурсами. Вычисления с пла-

вающей запятой такими процессорами на аппаратном уровне могут не поддерживаться или поддерживаться ограниченно. Компиляторы позволяют использовать данные типа *float* и *double*, однако соответствие стандарту IEEE754 ими не гарантируется.

Для решения всех описанных выше задач, в которых используются хаотические процессы, необходимо применение генератора хаотических последовательностей, обеспечивающего повторяемость результата на различных программно-аппаратных платформах. Эта цель может быть достигнута реализацией генератора хаотических последовательностей на основе вычислений с фиксированной запятой [9].

Однако вследствие ограниченной разрядности регистров ЭВМ при выполнении операций в формате с фиксированной запятой возникает ряд нежелательных эффектов, в числе которых шум усечения или образование предельных циклов [9, 14, 15]. Шум усечения может быть как коррелирован, так и не коррелирован с генерируемой хаотической последовательностью чисел. Образование же предельных циклов происходит как вследствие влияния шума усечения, так и по причине переполнения регистров [9].

В настоящей работе для обозначения чисел с фиксированной запятой будет использоваться так называемая *Q*-нотация вида *Qm.n*, где *m* – число бит, использующихся для хранения целой части числа; *n* – число бит, хранящих дробную часть. Числа в формате с фиксированной запятой в соответствии с [16] будут обозначаться индексом *fi*.

Для перевода вещественного десятичного числа в двоичный формат с фиксированной запятой целую часть вещественного числа переводят в двоичную систему счисления методом последовательного деления, а дробную часть – методом последовательного умножения [18]. Полученные результаты объединяют.

Число $a = 1,99$, преобразованное в двоичный формат с одним битом для хранения целой части и 17 битами для хранения дробной части, будет иметь вид $a_2 = 1,11111101011100001$. В данном преобразовании отброшены биты младше 17-го, т. е. произведено усечение числа. Обратное преобразование уже не даст изначальное число $a = 1,99$ по причине потери точности вследствие усечения исходного числа. Разница между числом до усечения и после является шумом усечения [15]:

$$\Delta_y = a - \hat{a}, \quad (4)$$

где Δ_y – шум усечения; a – число до усечения; \hat{a} – число после усечения.

Изменяя величину q , удается достичь приемлемого уровня шума. При этом существуют числа, которые не могут быть точно представлены в двоичном формате.

Помимо усечения, может быть использовано округление чисел. Усечение и округление применяются к числам как с фиксированной, так и с плавающей запятой.

Математическое ожидание шума округления равно нулю [15]. Дисперсия шума квантования при округлении вычисляется по формуле

$$D(\Delta_o) = \frac{2^{-2l}}{12}, \quad (5)$$

где l – разрядность регистра, хранящего число.

При усечении дисперсия шума квантования увеличивается:

$$D(\Delta_y) = \frac{2^{-2l}}{3}. \quad (6)$$

Операции над числами с фиксированной запятой производятся так же, как и над любыми другими числами, представленными в экспоненциальной форме.

Таким образом, при реализации кроссплатформенных генераторов хаотических последовательностей на базе цифровых ЭВМ может быть использована арифметика с фиксированной запятой. Влияние же ограниченной разрядности регистров ЭВМ и значительной дисперсии шума (8) оценивается экспериментально.

3. Экспериментальное исследование дискретных отображений в арифметике с фиксированной запятой

Исследование возможности применения целочисленной арифметики в задачах генерации хаотических последовательностей было проведено на примере отображений, представленных в табл. 1. Исследование включало два этапа. На первом этапе каждое отображение было реализовано в виде компьютерной программы на языке Си и языке СКМ Matlab. При этом были использованы только стандартные целочисленные типы данных длиной от 8 до 64 бит. Специализированные функции для выполнения операций над числами, представленными в формате с фиксированной запятой, не использовались. Однако при данных ограничениях логистическое отображение переходило из хаотического режима в периодический после 334-го шага, аттрактор Эннона – после шестого шага, кубическое отображение стабилизировалось после 39-го шага. В то же время расширенные возможности, предоставляемые функциями из набора *Fixed-point toolbox* СКМ Matlab или аналогичных наборов функций иных языков программирования, позволяют реализовать хаотические режимы отображений, представленных в табл. 1, на значительном числе отсчетов. Таким образом, на втором этапе исследования были применены функции из набора *Fixed-point toolbox* СКМ Matlab, позволившие определить такие длины целой и дробной частей переменных, при которых хаотический режим наблюдался на достаточно большом – до одного миллиона – числе отсчетов для каждого из отображений.

Отображения были реализованы в виде компьютерной программы на арифметике с фиксированной запятой на языке СКМ Matlab.

При выполнении программы была получена хаотическая последовательность данных. Показатель Ляпунова, измеренный для этой последовательности, как описано в разд. 2, и состоящей из 100 значений, составил $\lambda_1 = 0,6712$.

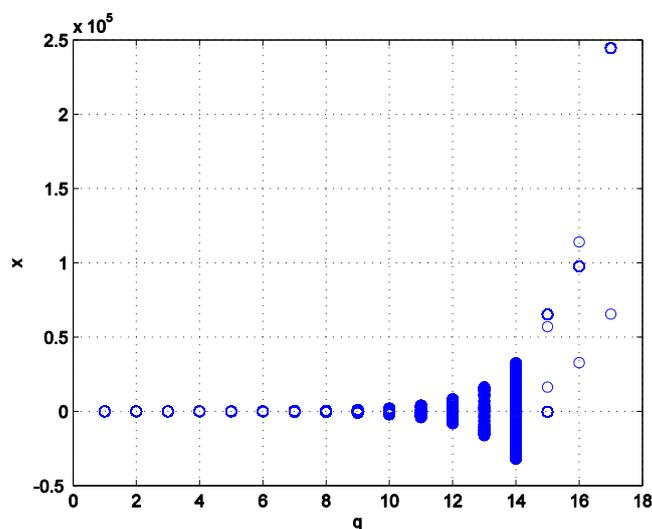


Рис. 2. Влияние величины q на результат вычислений значений логистического отображения

При указанных выше параметрах вычислений хаотический режим наблюдается при величине дробной части числа $q=13$ и $q=14$. Недостаточно высокая точность вычислений ниже указанных значений позволяет данной НДС сохранять стабильность. При значениях $q=15$ и выше и при использовании стандартных типов данных происходит переполнение регистров ЭВМ, вызывающее возникновение предельных циклов (рис. 2). Сравнение результатов вычислений, проведенных в форматах с фиксированной и плавающей запятой, показано на рис. 3.

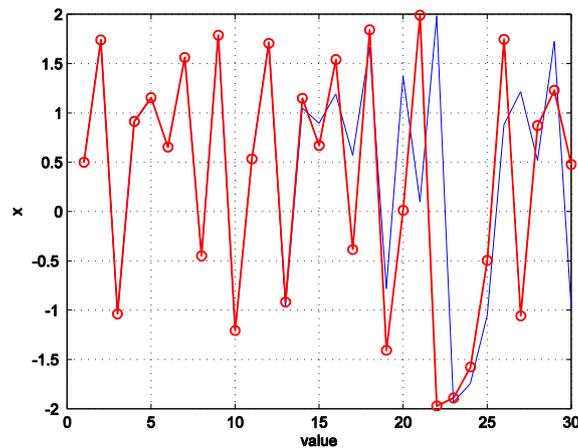


Рис. 3. Результат вычисления логистического отображения с плавающей запятой (красная линия с круглыми маркерами) и фиксированной запятой (сплошная синяя линия)

Как видно из рис. 3, до 13-го шага результаты полностью совпадают, после чего начинается расхождение. В формате $Q1.14$ после 334-го шага происходит переход от хаотического к периодическому режиму вследствие недостаточно высокой точности.

Бифуркационная диаграмма отображения (1), полученная при выполнении вычислений с фиксированной запятой (рис 4, а), напоминает классическую бифуркационную диаграмму логистического отображения [10, 17]. Виден переход к хаотическому режиму через удвоение периода. При каждой смене значения a новое значение x устанавливается не сразу (рис. 4, б).

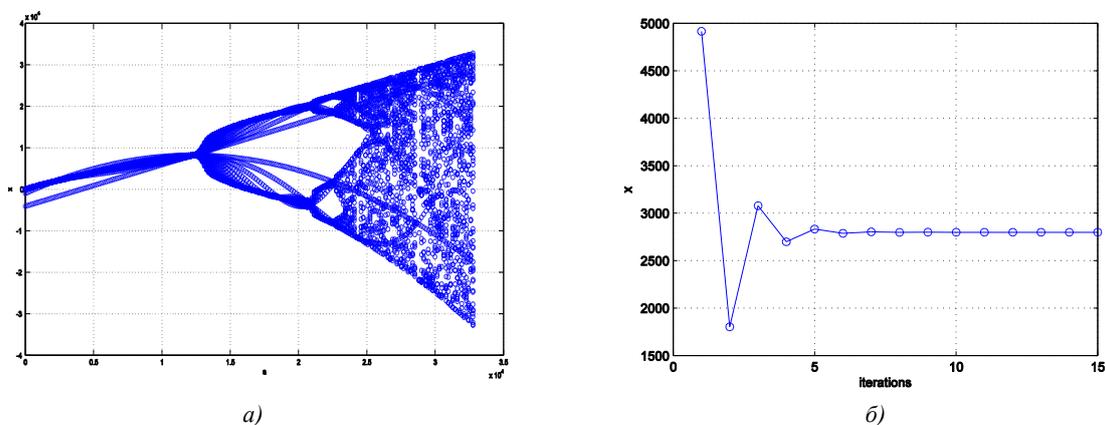


Рис. 4. Влияние управляющего параметра a на значение x : а) бифуркационная диаграмма логистического отображения; б) установление значения x при изменении величины a

Программа на языке Си, реализующая вычисления логистического отображения в формате с фиксированной запятой, позволяет получить повторяемый результат с использованием различных компиляторов не только на персональном компьютере, но и на микроконтроллерных и встраиваемых системах. В процессе испытаний она была откомпилирована с использованием компиляторов *icc* и *gcc*, выполнялась на персональном компьютере 64-битным процессором и восьмибитном микроконтроллере. Во всех случаях результат точно соответствовал тому, что был получен в СКМ Matlab.

Недостатком данной программы, как и описанной выше программы на языке СКМ Matlab, является переход от хаотического к периодическому режиму после 334-й итерации по причине недостаточно высокой точности. Преодолеть данный недостаток можно путем использования в программе нестандартных типов данных длиной более 64 бит. Пример програм-

мы на языке СКМ Matlab, использующей 128 бит для хранения числа и построенной на основе *Fixed-point toolbox*, приведен в табл. 2.

Таблица 2
Текст программы на языке СКМ Matlab с использованием *Fixed-point toolbox* для вычисления логистического отображения

| Номер строки | Текст программы |
|--------------|--|
| 1 | <code>clear all; close all;</code> |
| 2 | <code>wordLength = 128;</code> |
| 3 | <code>fractionLength = 126;</code> |
| 4 | <code>a = fi(1.99,1,wordLength,fractionLength);</code> |
| 5 | <code>X = fi(1:50000,1,wordLength,fractionLength);</code> |
| 6 | <code>startX = fi(0.5,1,wordLength,fractionLength);</code> |
| 7 | <code>X(1) = startX;</code> |
| 8 | <code>for i = 2:50000</code> |
| 9 | <code> X(i) = a - X(i-1)^2;</code> |
| 10 | <code>end</code> |

Графики по переменной x , иллюстрирующие выход из хаотического режима аттрактора Энона и кубического отображения при вычислении в формате $Q1.14$, показаны на рис. 5.

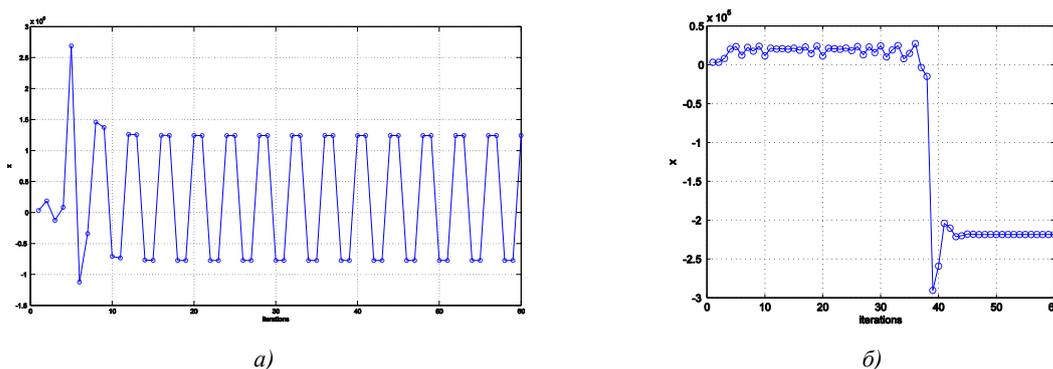


Рис. 5. Графики последовательностей x : а) для аттрактора Энона; б) кубического отображения

При $Q1.14$ аттрактор Энона демонстрирует возникновение предельного цикла по причине переполнения регистров на шестом шаге при вычислении нового значения x . При вычислении кубического отображения переполнение происходит на 39-м шаге.

Для достижения хаотического режима аттрактором Энона и кубическим отображением при реализации на арифметике с фиксированной запятой требуется повышение точности вычислений. Достижение хаотического режима возможно также путем применения метода динамического изменения формата данных, при котором количество бит, хранящих целую и дробную части, изменяется в процессе выполнения программы для исключения переполнений.

По результатам моделирования логистического отображения с использованием *Fixed-point toolbox* хаотический режим на одном миллионе отсчетов наблюдался при $Q2.41$.

Заключение

В работе описана реализация логистического отображения с использованием арифметики с фиксированной запятой, проведено исследование полученных результатов. Отличительной особенностью данного отображения является высокая чувствительность к начальным условиям (старшая экспонента Ляпунова равна 0,6607). Другая особенность состоит в том, что диапазон значений x не превышает пределов от -2 до 2 , благодаря чему удалось использовать только два бита для кодирования целой части получаемых результатов.

Целочисленная реализация других типов дискретных НДС, описанных выше, с теми же параметрами точности и типами данных показала, что для решения таких отображений требуется более высокая точность вычислений, чем при решении логистического отображения.

Полученный результат может быть использован в ряде практических задач. Так, в частности, в работе [19] описывается общая схема поискового адаптивного поведения с инерционным переключением между поисковыми тактиками. Как было указано в данной работе, эффективное поисковое движение может быть реализовано сочетанием перемещения на значительные расстояния, частыми случайными переменами в направлении движения и инерционностью переключения между каждым из видов движения. Данный способ поискового поведения может быть эффективно использован в области мобильной робототехники для управления мобильным роботом, выполняющим задачу поиска на местности некоторых предметов, явлений или условий. При этом описываемый способ поискового поведения удобно реализовать в бортовой системе управления мобильного робота именно на основе целочисленного генератора хаотических последовательностей.

Интерес представляет также возможность криптозащиты информационного обмена по радиоканалу между роботами и системой управления вышестоящего уровня с помощью хаотических последовательностей. В этом случае важную роль играет кроссплатформенность целочисленного подхода, что позволит организовать взаимодействие между вычислительными системами различных типов.

Работа выполнена при поддержке гранта БРФФИ–ГФФИУ № Ф13К-144 «Разработка методов оперативной обработки и передачи информации для эффективного управления мобильными роботами и подвижными системами».

Список литературы

1. Пригожин, И. Порядок из хаоса / И. Пригожин, И. Стенгерс. – М. : Прогресс, 1986. – 431 с.
2. Малинецкий, Г.Г. Хаос. Структуры. Вычислительный эксперимент: введение в нелинейную динамику / Г.Г. Малинецкий. – М. : Эдиториал УРСС, 2000. – 256 с.
3. Handbook of Chaos Control / E. Schöll [et al.]. – 2nd ed. – Wiley-VCH Verlag GmbH&Co.KGaA, 2008. – 819 p.
4. Chaos in Automatic Control / W. Perruquetti [et al.] ; ed. W. Perruquetti. – CRC Press, 2005. – 564 p.
5. Moon, F. Chaotic Vibrations: An Introduction for Applied Scientists and Engineers / F. Moon. – John Wiley&Son, 2004. – 309 p.
6. Basic principles of direct chaotic communications / A.S. Dmitriev [et al.] // Nonlinear Phenomena in Complex Systems. – 2003. – Vol. 6, no. 1. – P. 488–501.
7. Хаотические процессоры / Ю.В. Андреев [и др.] // Успехи современной радиоэлектроники. – 1997. – № 10. – С. 50–79.
8. Nonlinear Dynamics in Circuits / T. Carroll [et al.] ; ed. T Carroll. – World Scientific, 1995. – 344 p.
9. Рабинер, Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Б. Гоулд. – М. : Мир, 1978. – 848 с.
10. Gilmore, R. The Topology of Chaos / R. Gilmore, M. Lefranc. – John Wiley & Sons, 2008. – 518 p.
11. A practical method for calculating largest Lyapunov exponents from small data sets / M.T. Rosenstein [et al.] // Physica D 65 1993. – P. 117–134.
12. Largest Lyapunov Exponent with Rosenstein's Algorithm // Matlab Central [Electronic resource]. – Mode of access : <http://www.mathworks.com/matlabcentral/fileexchange/38424-largest-lyapunov-exponent-with-rosensteins-algorithm/content/lyarosenstein.m>. – Date of access : 30.03.2014.
13. Standard for Binary Floating-Point Arithmetic : IEEE 754–2008. – Institute of Electrical and Electronics Engineers, 2008. – 23 p.
14. Оппенгейм, А. Цифровая обработка сигналов / А. Оппенгейм, Р. Шафер. – М. : Техносфера, 2006. – 856 с.

15. Передача дискретных сообщений : учебник для высших учебных заведений / В.П. Шувалов [и др.] ; под ред. В.П. Шувалова. – М. : Радио и связь, 1990. – 464 с.
16. Wilkinson, J.H. Rounding Errors in Algebraic Processes / J.H. Wilkinson. – Dover Publications, 1994. – 161 p.
17. Korsch, H.J. Chaos. A program collection for PC / H.J. Korsch, H-J. Jodl, T. Hartmann. – 3rd ed. – Springer, 2008. – 341 p.
18. Путков, В.Н. Электронные вычислительные устройства : учеб. пособие для радиотехн. спец. вузов / В.Н. Путков, И.И. Обросов, С.В. Бекетов. – Минск : Вышэйшая школа, 1981. – 333 с.
19. Непомнящих, В.А. Бионическая модель адаптивного поискового поведения / В.А. Непомнящих, Е.Е. Попов, В.Г. Редько // Известия РАН. Теория и системы управления. – 2008. – № 1. – С. 85–93.

Поступила 03.05.2014

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: vsychyov@robotics.by*

U.A. Sychou

REALIZATION OF DISCRETE NONLINEAR DYNAMICAL SYSTEMS WITH CHAOTIC REGIMES BASED ON FIXED-POINT ARITHMETIC

In this article, the problem of the practical realization of nonlinear systems with chaotic dynamics for targeted generation of chaotic sequences in digital devices is considered. The possible application in this task with using fixed-point arithmetic to ensure the identity of the obtained results on different hardware and software platforms is studied. The implementation of logistic mapping is described; carry out the analysis of the results. This article proposes using the obtained results for the various tasks of the field of mobile robotics.