

Алгоритм временного сжатия сформированных образов в системах автономного адаптивного управления

Юрий Мазур, Александр Жданов

Институт точной механики и вычислительной техники им. С. А. Лебедева, Москва, Россия

yumazur@ipmce.ru, aazhdanov@ipmce.ru

Abstract

Autonomous adaptive control systems during working constantly observe input information from sensors, which has to be saved by certain method in embedded memory, what is necessary for self-learning (adaptation) of the control system. However, memory capacity for storage of input information is limited in common case, therefore the problem of saving huge amount of information in the limited volume of memory with consideration not uniformity of information importance is arisen. We assume that importance of indicated information in biological systems from one hand depends on the period of remoteness, and from the other hand on the related emotional evaluation [1]. For biologically-inspired systems it is necessary to develop suitable method that is described in the current paper. Proposed method corresponds to known properties of human memory, consisting in that we, the human beings, remember observed events the more distinctly the newer they are or the more they related with powerful emotional impressions and, vice versa, we remember events the worse the older they are or the more they are neutral in emotional impressions. In the current paper corresponding method of input information stream compression is proposed.

Algorithm allows compressing unlimited amounts of information, saving the most important information maximally detailed over the time of system life. It is shown necessity of estimation of system parameters, which provide saving of useful, nonsingular information.

Keywords: *autonomous adaptive control system, data compression, emotions.*

Аннотация

Системы автономного адаптивного управления во время функционирования постоянно наблюдают поступающую с сенсоров входную информацию, которая некоторым способом должна сохраняться во внутренней памяти, что необходимо для самообучения (адаптации) системы управления. Однако, объем памяти для хранения входной информации в любых системах ограничен, поэтому возникает проблема размещения большого объема информации в ограниченном объеме памяти с учетом неоднородности информации по ее важности. Мы предполагаем, что важность указанной информации в биологических системах зависит, с одной стороны, от срока давности информации, а с другой стороны – от связанной с ней эмоциональной оценки [1]. Для биологически-инспирированных систем управления необходимо разработать соответствующий способ, чему и

посвящена данная работа. Предлагаемый метод соответствует известным нам свойствам человеческой памяти, состоящим в том, что мы помним наблюдаемые события тем отчетливее, чем они «свежее» или чем более они были связаны с сильными эмоциональными впечатлениями, и наоборот – помним их тем хуже, чем они «старее» или чем они нейтральнее по эмоциональным впечатлениям. В настоящей работе предлагается соответствующий метод сжатия потока входной информации.

Алгоритм позволяет сжимать неограниченные объемы информации, при сохранении наиболее важной её части максимально подробно на протяжении всей жизни системы. Показана необходимость оценивания параметров системы, обеспечивающих сохранение полезной, невырожденной информации.

Ключевые слова: *автономные адаптивные системы управления, сжатие данных, эмоции.*

1. ВВЕДЕНИЕ

В процессе конструирования сложных автоматизированных систем управления все чаще приходится сталкиваться с объектами и свойствами окружающей среды, которые изменяются со временем, что значительно затрудняет создание детерминированных систем управления. Растет спрос на системы управления, способные самостоятельно принимать решения в непредусмотренных условиях, в частности, это относится к мобильным и автономным роботам.

Одним из возможных подходов к созданию подобных адаптивных систем является метод автономного адаптивного управления (ААУ) [1-3]. Метод базируется на предложенной автором концептуальной модели нервных систем, и призван воспроизводить естественные механизмы адаптации живых существ к изменяющимся условиям существования. В системе ААУ важную роль играет «подсистема эмоций», которая, в частности, обеспечивает «мотивацию» к активному поведению объекта управления, старающегося максимизировать интегральную эмоциональную оценку своих текущего и будущих состояний. На рисунке 1 представлена структура системы управления в соответствии с методом ААУ.

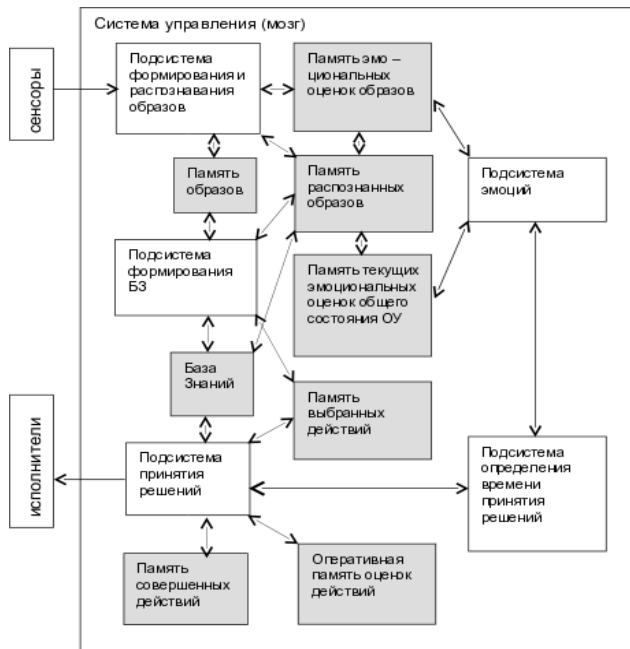


Рисунок 1: Структура функциональных подсистем и памяти управляющей системы ААУ.

Во время работы данной системы управления временная последовательность распознанных подсистемой распознавания и формирования изображений образов добавляется в память распознанных образов, но так как эта память ограничена, то после добавления нового образа приходится удалять самый старый образ. Однако, удаляя его, мы рискуем удалить важные для системы управления данные, так как, как видно из рисунка 1, эта память используется и другими самообучаемыми подсистемами, в частности, подсистемой формирования базы знаний, которая в некоторых реализациях может использовать протяженные временные последовательности из памяти распознанных образов.

Таким образом, возникает задача сохранения максимального количества образов в ограниченном объеме памяти, причем, желательно самые важные для системы управления данные хранить наиболее долго.

2. МОДЕЛЬ ПАМЯТИ РАСПОЗНАННЫХ ОБРАЗОВ

Как известно, для хранения больших объемов информации в ограниченной памяти часто используют сжатие с помощью различных алгоритмов [4,5]. При этом, сжатие информации возможно как с потерями данных (например, многие алгоритмы, используемые в аудио (MP3, AAC и т.д.) и видео (MPEG-1/2/4) кодировании, а также при сжатии цифровых фотографий (JPEG, GIF, PNG)), так и без потерь (например, алгоритмы, используемые при сжатии файлов в архивы Zip, RAR, gz и т.д.). В данной работе предлагается оригинальный алгоритм сжатия с потерями, который описывается ниже. Причина использования сжатия с потерями заключается в следующей аналогии с памятью человека. Как известно, людям свойственно забывать подробности событий, происходивших давно. С другой стороны события, которые произошли недавно, еще свежи в памяти вместе с их мельчайшими подробностями. Подобным

же образом с неравномерностью предполагается сжимать данные по времени и в настоящей работе.

Кроме описанного выше сжатия по времени, у памяти человека можно отметить следующую особенность, а именно эмоциональную направленность памяти. Например, человек иногда очень хорошо помнит произошедшие 20 или 30 лет назад события, которые связаны с сильными переживаниями и эмоциями (как с положительными, так и с отрицательными). Таким образом, помимо сжатия образов (событий) по времени следует еще учитывать связанную с ними эмоциональную оценку. Поясним, что термин “эмоциональная оценка” взят из упомянутого выше метода ААУ, в котором он отражает субъективную интегральную качественную оценку текущего состояния системы, вырабатываемую бионической системой управления в соответствие с заложенными в нее правилами. Данная оценка вырабатывается аппаратом эмоций, существующим в каждой нервной системе.

Итак, перейдем непосредственно к описанию алгоритмов сжатия образов по времени с учетом эмоциональных оценок.

2.1 Сжатие образов по времени

Пусть на вход разрабатываемого блока памяти распознанных образов в каждый момент времени $t_k, k \in \mathbb{N}$ поступает новый распознанный образ $P(t_k)$, характеризующийся в пространстве признаков вектором $\bar{X}(t_k) = \{x_1(t_k), \dots, x_N(t_k)\}$. Причем будем рассматривать такие системы распознавания и формирования образов, что существует функция $\Phi(P(t_k))$ такая, что:

$$\{\bar{X}_{init}(t_k), \bar{X}_{fin}(t_k)\} = \Phi(P(t_k)) \quad (1)$$

где N-мерные вектора $\bar{X}_{init}(t_k)$ и $\bar{X}_{fin}(t_k)$ таковы что $\bar{X}_{init}(t_k) = \{\tilde{x}_1(t_k), \dots, \tilde{x}_N(t_k)\}$ и $\bar{X}_{fin}(t_k) = \{\tilde{\tilde{x}}_1(t_k), \dots, \tilde{\tilde{x}}_N(t_k)\}$ и для любого $j, 1 \leq j \leq N$ выполняется следующее неравенство $\tilde{x}_j(t_k) \leq x_j(t_k) \leq \tilde{\tilde{x}}_j(t_k)$. Смысл векторов $\bar{X}_{init}(t_k)$ и $\bar{X}_{fin}(t_k)$ в ограничении области признакового пространства, в которой находится вектор $\bar{X}(t_k)$, характеризующий образ $P(t_k)$.

Итак, сохраним в памяти распознанных образов ω_1 самых последних образов $\{P(t_{k-\omega_1+1}), \dots, P(t_k)\}$. Назовем эту последовательность образов l-м уровнем сжатия, а ω_1 числом окон на данном уровне и запишем в виде $\{C(1, \omega_1, t_{k-\omega_1+1}, t_{k-\omega_1+1}), \dots, C(1, 1, t_k, t_k)\}$, где первый параметр это номер уровня, второй - номер окна на данном уровне, третий и четвертый параметры это начальный и конечный моменты времени, к которым относится данный образ соответственно (см. Рис. 2). Обозначим как $l \in \mathbb{N}, 1 \leq l \leq L$ текущий рассматриваемый уровень из L возможных. Причем для первого уровня ($l=1$) верно следующее: $C(1, \omega_1, t_{k-\omega_1+1}, t_{k-\omega_1+1}) = P(t_{k-\omega_1+1}), \dots, C(1, 1, t_k, t_k) = P(t_k)$.

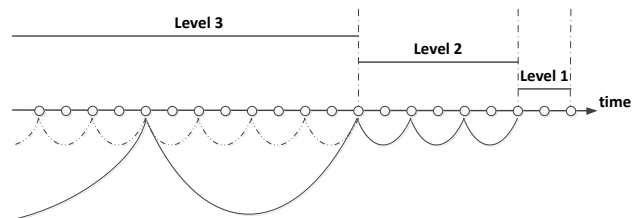


Рисунок 2: Схема трех уровней сжатия по времени.

Теперь рассмотрим второй уровень $l = 2$, на котором находится ω_2 окон второго уровня. Каждое окно данного уровня с индексом $p \in \mathbb{N}$, $1 \leq p \leq \omega_l$ рассчитывается следующим образом:

Шаг 1. Для каждого окна второго уровня запоминается во временной памяти β_2 значений предыдущего уровня (в данном случае первого) $\{C(1, \omega_1, t_{k-\omega_1+1-\beta_2}, t_{k-\omega_1+1-\beta_2}), \dots, C(1, 1, t_{k-\omega_1+1}, t_{k-\omega_1+1})\}$. Сначала заполняется память для окна с номером 1, потом 2 и т.д. до ω_2 , таким образом, значения предыдущего уровня, используемые для окна 1, смещаются во временной памяти и используются для окна 2. После того как эти значения становятся ненужными для окна ω_2 они удаляются из временной памяти.

Шаг 2. Если для данного окна достаточно накопленных данных предыдущего уровня, то вычисляется значение этого окна и записывается в память распознанных образов. Записываемое в память значение вычисляется по следующей формуле (близко к вычислению среднего значения для компонент векторов из признакового пространства сжимаемых образов):

$$C(2, p, t_{\alpha_1}, t_{\alpha_2}) = \frac{1}{\beta_2} \sum_{t_{\alpha_1} \leq t_{init} \leq t_{fin} \leq t_{\alpha_2}} C(1, \omega_1, t_{init}, t_{fin}) \quad (2)$$

где

$$\alpha_1 = k - \omega_1 + 1 - p\beta_2 + 1$$

$$\alpha_2 = k - \omega_1 + 1 - (p - 1) * \beta_2$$

Суммирование в правой части формулы (2) и деление на β_2 следует понимать как операции над соответствующими компонентами векторов из признакового пространства, описывающие образы $C(1, \omega_1, t_{init}, t_{fin})$. В результате такой операции получается новый вектор из признакового пространства, который в свою очередь характеризует образ $C(2, p, t_{\alpha_1}, t_{\alpha_2})$, т.е. при сжатии фактически формируется новый образ – образ усреднённого воспоминания. Данная операция возможна благодаря требованию (1) и существованию функции $\Phi(P(t_k))$. Стоит отметить, что процедуры сжатия образов можно определить, видимо, и иными способами.

Если для данного окна недостаточно данных предыдущего уровня, то вычисление не происходит, и дальнейшая обработка окон данного и более высокого уровней прекращается.

Шаг 3. Если на шаге 2 вычислено новое значение для последнего окна ω_2 , то оно помещается во временную память для дальнейшего расчета уровня 3, если максимальное число уровней $L \geq 3$.

Подобным же образом уровень за уровнем происходит расчет всех окон, и результат записывается в память распознанных образов.

Выпишем теперь общую формулу для произвольного уровня l и окна p , использованную на шаге 2:

$$C(l, p, t_{\alpha_1}, t_{\alpha_2}) = \frac{1}{\beta_l} \sum_{t_{\alpha_1} \leq t_{init} \leq t_{fin} \leq t_{\alpha_2}} C(l-1, \omega_{l-1}, t_{init}, t_{fin}) \quad (3)$$

где

$$\alpha_1 = k - \sum_{i=1}^{l-1} (\omega_i * \beta_i) - p\beta_l + l$$

$$\alpha_2 = k - \sum_{i=1}^{l-1} (\omega_i * \beta_i) - (p - 1) * \beta_l + l - 1$$

Общая формула для величины β_l , выражающая количество необходимых элементов уровня $l-1$ для расчета одного элемента уровня l :

$$\beta_l = \begin{cases} d_l, & l = 1, 2 \\ (d_l - 1)(d_{l-1} - 1) \prod_{i=3}^{l-1} d_i - 1, & l \geq 3 \end{cases}$$

где d_i – число элементов уровня $i-1$, задающих размер одного окна уровня i . Причем $d_1 = 1$, так как на нем находятся элементарные точки.

2.2 Учет эмоциональной оценки при сжатии образов

Используем теперь эмоциональную оценку распознанных образов, обозначив её через $S(P(t_k))$. Сразу стоит отметить, что здесь и далее под значением эмоциональной оценки понимается её модуль (обозначим через $\check{S}(P(t_k)) = |S(P(t_k))|$). Это объясняется тем, что мы можем предположить примерно одинаковую степень забывания событий с близкими по модулю отрицательными и положительными значениями. Как уже упоминалось в методе ААУ, у каждого образа существует его эмоциональная оценка, однако возможно несколько вариантов учета этой оценки при сжатии. Например, следующие:

А) Использование эмоциональной оценки напрямую при сжатии. Т.е. можно на шаге 2 в пункте 2.1 назначить каждому элементу суммы вес в зависимости от его эмоциональной оценки. Тогда, очевидно, более эмоциональные образы будут вносить больший вклад в сумму, и в результате будет потеряно меньше информации об этих образах. Возможным недостатком данного подхода является слишком значительная потеря информации о не очень эмоциональных, но потенциально важных образах.

Б) Использование эмоциональной оценки как еще одного измерения пространства (1-е это время), в котором хранятся образы. Тем самым, образы с высокой эмоциональной оценкой будут сжиматься независимо от образов с низкой оценкой (см. Рис. 3).

Недостатком этого подхода является использование большего объема памяти, хотя конкретные цифры сильно зависят от используемых параметров сжатия и шкалы \check{S} .

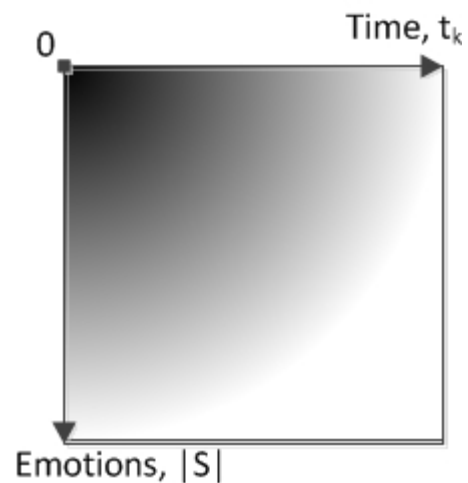


Рисунок 3: Схематическая зависимость степени сжатия образа от времени, прошедшего с момента его распознавания и его эмоциональной оценки. Более темный цвет заливки означает более высокую степень сжатия.

В данной работе используется подход, описанный в пункте Б) в следующем виде:

- Вместо непрерывной оси \check{S} , использовалось разбиение на эмоциональные уровни E_r , т.е. весь диапазон эмоциональных оценок разбивался равномерно на заданное число уровней R . Границы эмоционального уровня E_r обозначаются следующим образом:
 \check{S}_{r-1} – нижняя граница уровня r
 \check{S}_r – верхняя граница уровня r
 Причем $\check{S}_0 = \check{S}_{init}$ и $\check{S}_R = \check{S}_{fin}$ изначально задаются.
- Для каждого эмоционального уровня используется независимое сжатие образов по времени (см. пункт 2.1 выше)
- Существует нулевой уровень (E_0), на который попадают все образы с любыми эмоциональными оценками.

Получается следующая последовательность действий для нового распознанного образа $P(t_k)$ с модулем эмоциональной оценки $\check{S}(P(t_k))$:

Шаг 1. Независимо от $\check{S}(P(t_k))$, образ всегда попадает на уровень E_0 и сжимается по времени.

Шаг 2. Если $\check{S}(P(t_k)) \geq \check{S}_0$, тогда находится номер уровня γ , для которого верно следующее неравенство:

$$\check{S}_{\gamma-1} \leq \check{S}(P(t_k)) < \check{S}_\gamma$$

Шаг 3. На найденный уровень γ добавляется образ $P(t_k)$ и сжимается по времени.

Отметим также, что на всех эмоциональных уровнях E_r , за исключением нулевого E_0 , хранящиеся в памяти сжатые образы часто имеют между собой значительные временные промежутки, что усложняет их анализ в хронологическом порядке. Для решения этой проблемы используется простое техническое решение, а именно указание временных меток (“привязок”) начала и конца интервала, к которому относится сжатый образ. Эти метки начала и конца интервала в формулах (2) и (3) уже были введены и обозначены, как t_{init} и t_{fin} соответственно.

3. ИСПОЛЬЗУЕМЫЕ ИНСТРУМЕНТЫ И РЕЗУЛЬТАТЫ

Программная реализация прототипа памяти и визуализация результатов его работы была осуществлена с помощью языка программирования Java и свободно распространяемой библиотеки JFreeChart [6].

Было смоделировано поступление “сырых” данных в геометрическую систему распознавания, а затем результаты распознавания записывались в разработанную память распознанных образов (см. Рис. 4-6). Эти образы характеризовались в признаковом пространстве лишь одним абстрактным параметром. Входные “сырые” данные эмулировались периодической функцией с несколькими гармониками и случайным шумом (см. Рис. 5). Функция, моделирующая эмоциональную оценку образов, имела следующий вид:

$$S(P(t_k)) = \check{S}(\bar{X}_{init}(t_k), \bar{X}_{fin}(t_k)) = 20 * \exp(-10(\frac{\tilde{x}_1(t_k) + \tilde{\tilde{x}}_1(t_k)}{\tilde{x}_1(t_k) + \tilde{\tilde{x}}_1(t_k) + 2}))$$

где $\tilde{x}_1(t_k)$ и $\tilde{\tilde{x}}_1(t_k)$ определены согласно (1) и поскольку $N=1$.

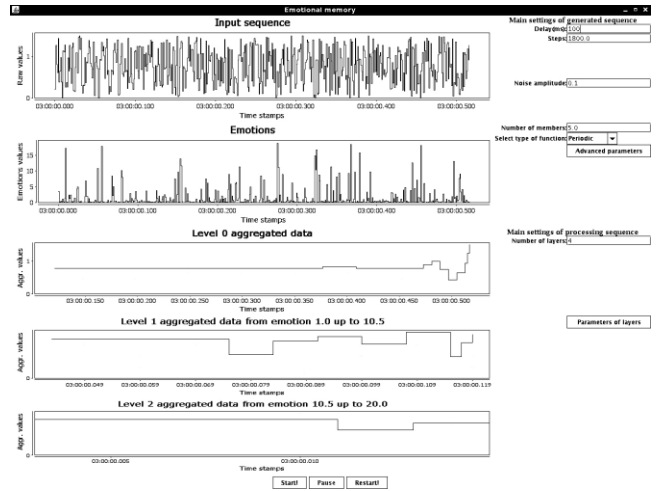


Рисунок 4: Обработка образов, содержащих случайные синусоидальные данные с шумом. Используются только два эмоциональных уровня и один нулевой уровень. Более “свежие” данные – справа.

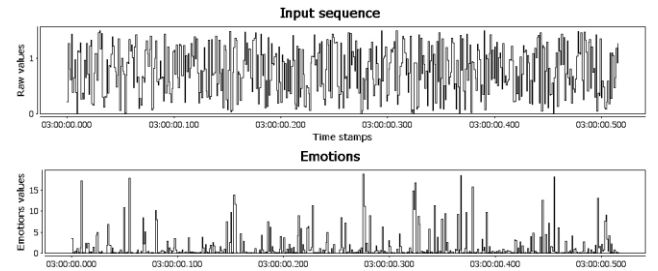


Рисунок 5: Первый график сверху – “сырые” данные, поступающие на вход системе распознавания. Второй график сверху – эмоциональная оценка распознанных образов.

На рисунке 6 показано состояние разработанной памяти распознанных образов в некоторый момент времени. Использован один нулевой уровень и два эмоциональных уровня памяти. Видны окна каждого уровня и границы самих уровней. Видно также, что более новые данные (справа) более подробны, чем более старые данные (слева).

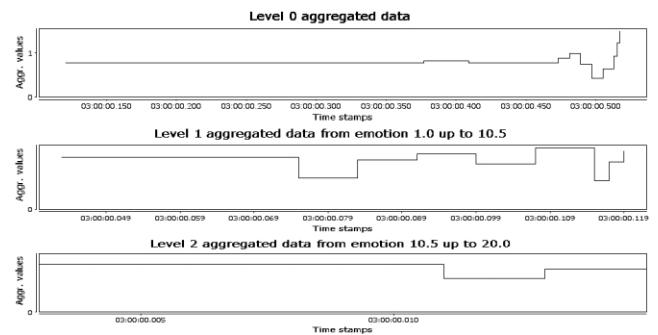


Рисунок 6: Многоуровневая разработанная память распознанных образов в определенный момент времени.

4. ЗАКЛЮЧЕНИЕ

Таким образом, мы видим (в том числе – и по результатам моделирования), что в предложенной памяти, информация неравномерно отражает предысторию: с одной стороны, чем

информация новее, тем она более подробно описывает предысторию, с другой стороны, информация о фактах, связанных с более сильными эмоциональными оценками, подвергается меньшему сжатию со временем.

Хотелось бы отметить ключевое отличие разработанного алгоритма от большинства используемых алгоритмов, состоящее в том, что информация проходит многоступенчатое сжатие, причем на вход данного уровня подаются результаты сжатия с предыдущего уровня. Эта особенность, возможно, позволит лучше понять работу памяти человека, а также улучшить качество управления в системах ААУ.

Первая предложенная нами реализация алгоритма достаточно проста. В дальнейшем алгоритм можно значительно улучшить, например, заменив простейшую функцию на шаге 2 пункта 2.1 на более эффективную функцию [5,4]. Также можно оптимизировать автомасштабирование эмоциональных уровней в пункте 2.2, так как сейчас происходит добавление еще одного уровня, если эмоциональная оценка нового образа больше верхней заданной границы $\check{S}(P(t_k)) > \check{S}_{fin}$.

5. ССЫЛКИ

- [1] Жданов А.А. *“Автономный искусственный интеллект”*. Монография. М.: БИНОМ. Лаборатория знаний, 2008.
- [2] Жданов А. А. *“Метод автономного адаптивного управления”*. Известия Академии Наук. Теория и системы управления, 1999, 5, с. 127-134.
- [3] Жданов А.А. *“О методе автономного адаптивного управления” (лекция)*. Научная сессия МИФИ-2004. VI Всероссийская н.-т. конференция *“Нейроинформатика-2004”*: Лекции по нейроинформатике. Ч. 2. - М.: МИФИ, 2004. с. 15-56.
- [4] D. Salomon. *“Data compression: the complete reference”*, 4-th ed., Springer, 2007.
- [5] A. Moffat, A. Turpin. *“Compression and coding algorithms”*, Kluwer AP, 2002.
- [6] JFreeChart. *“<http://www.jfree.org/jfreechart/>”*

Об авторах

Юрий Мазур – аспирант группы методов автономного адаптивного управления в институте точной механики и вычислительной техники им. С. А. Лебедева. Контактный e-mail: yuamazur@ipmce.ru.

Александр Жданов – профессор, доктор физ.-мат. наук, руководитель группы методов автономного адаптивного управления в институте точной механики и вычислительной техники им. С. А. Лебедева. Контактный e-mail: aazhdanov@ipmce.ru.