

БИЛЬФЕЛЬД Н. В., ВОЛОДИНА Ю. И.
ПОДАЧА ПРОИЗВОЛЬНОГО СИГНАЛА
НА ВХОД SIMULINK-МОДЕЛИ
УДК 681.5+004.02, ВАК 05.13.00, ГРНТИ 20.00.00

Подача произвольного сигнала на
вход simulink-модели

Non-standard signal input to simulink
model

Н. В. Бильфельд, Ю. И. Володина

N. V. Bilfeld, Yu. I. Volodina

Березниковский филиал Пермского
национального исследовательского
политехнического университета,
г. Березники

Berezniki Branch of Perm National Re-
search Polytechnic University,
Berezniki

В статье рассмотрены способы подачи произвольного (не стандартного) сигнала на вход модели при исследовании поведения систем. Приведено использование блоков библиотеки simulink, а также программная реализация нестандартных сигналов. Рассмотрен пример по использованию пакета по многокритериальному анализу систем управления. Даны рекомендации по сохранению массивов используемых сигналов.

This article is about of methods of non-standard signal inputs to simulink model for investigation of models. Using of simulink blocks is described. Software sources of non-standard signals are given. An example of using of multi-criteria control systems analysis software is presented. Some recommendations for signal arrays saving are given.

Ключевые слова: моделирование, управление, исследование, simulink.

Keywords: modeling, control, re-
search, simulink.

Введение

При моделировании систем управления для исследования динамических свойств объекта на вход системы, как правило, подается один из стандартных возмущающих воздействий [1]. К таким воздействиям принято относить следующие сигналы:

- единичный скачок (функция Хевисайда);
- единичный импульс (бета-импульс);
- гармонический сигнал (как правило, синусоидальный сигнал).

Но при исследовании поведения реальной системы, моделируемой в единицах технологического параметра [2] часто возникает необходимость подать на вход модели не стандартный сигнал. Особенно такая необходимость возникает

при проверке точности идентификации при пассивном эксперименте [3]. При таком эксперименте на входе объекта присутствует случайно изменяющийся во времени сигнал, получаемый с реальных трендов технологических контроллеров. Рассмотрим основные способы моделирования таких систем.

Использование блока *Signal Builder*.

Блок *Signal Builder* находится в стандартной библиотеке *simulink* в разделе *Sources* [4]. Рассмотрим варианты его использования.

Создадим новую модель и поместим в нее указанный блок, как показано на рисунке 1.



Рисунок 1. Блок *Signal Builder*

В командном окне MATLAB введем два вектора, например,

$$X = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]$$

$$Y = [2 \ 5 \ 3 \ 5 \ 8 \ 4 \ 3 \ 5 \ 7]$$

Кликнем по блоку. В результате откроется окно, приведенное на рисунке 2.

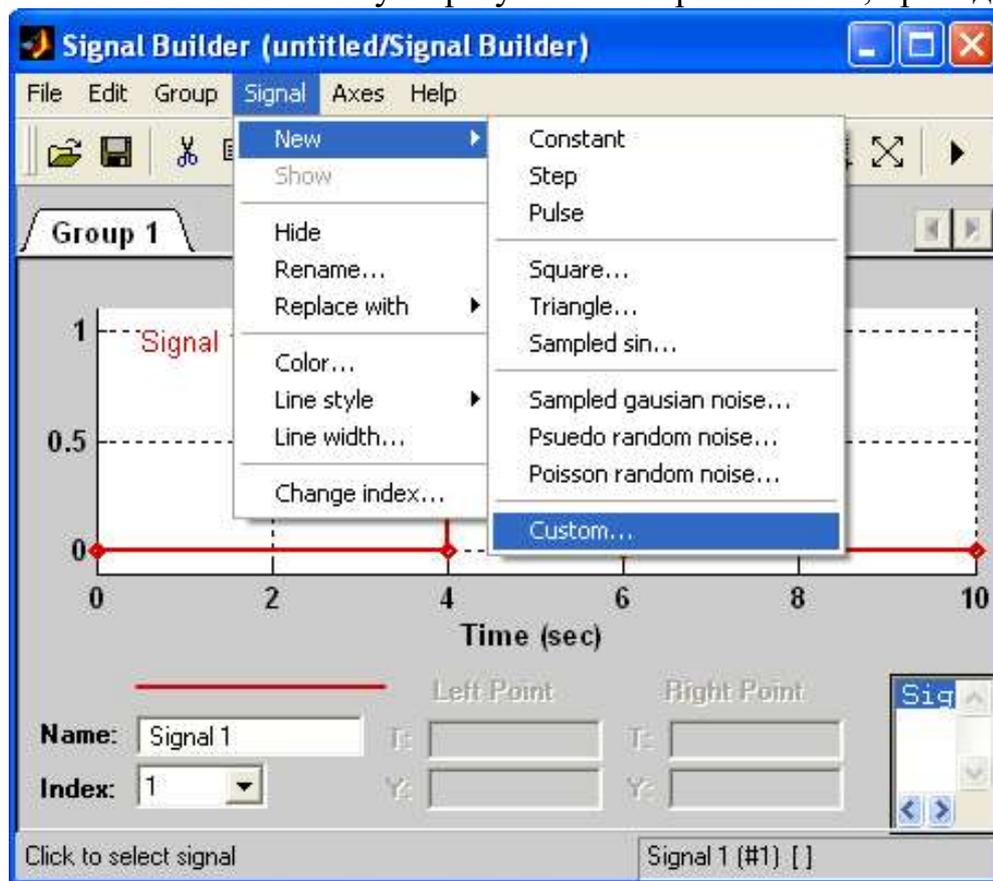


Рисунок 2. Создание сигнала пользователя

Выполним меню «*Signal > New > Custom*». В открывшемся окне введем наши вектора, как показано на рисунке 3.

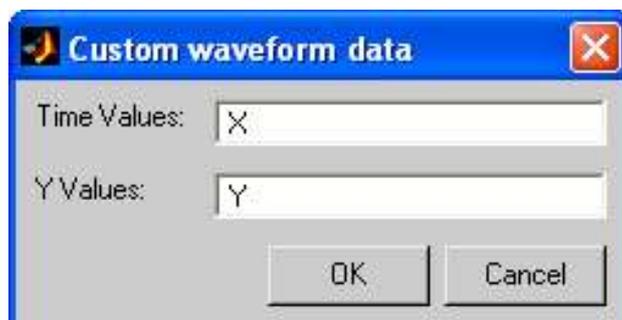


Рисунок 3. Задание векторов

В результате у блока появится второй выход, на который подключим блок Scope, как показано на рисунке 4.

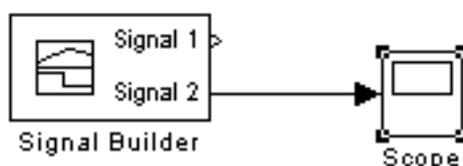


Рисунок 4. Вид блока после задания сигнала

После запуска модели сигнал на осциллографе будет выглядеть, как показано на рисунке 5.

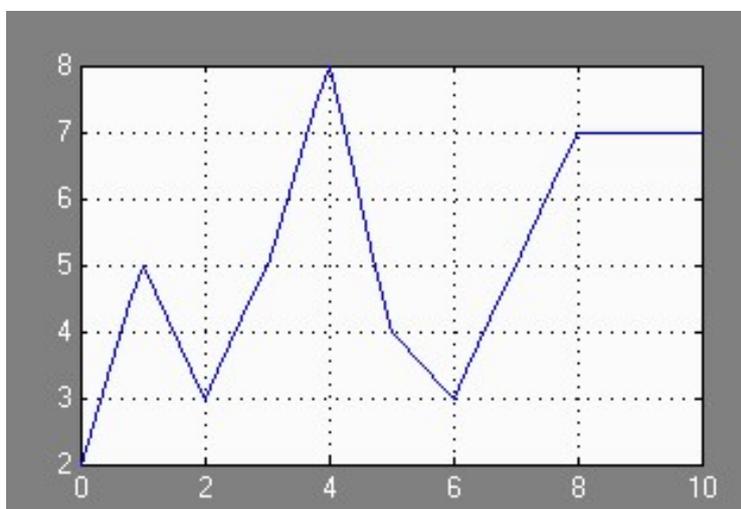


Рисунок 5. Сигнал, заданный векторами. Время моделирования равно 10

С помощью блока *Signal Builder* можно формировать сразу несколько сигналов. В нашем случае на первом выходе блока будет присутствовать единичный скачок (сигнал по умолчанию), а на втором выходе наш сформированный сигнал.

Удалить текущий сигнал можно, выполнив на графике сигнала контекстное меню *Delete*, как показано на рисунке 6. Теперь на выходе блока будет только один сигнал.

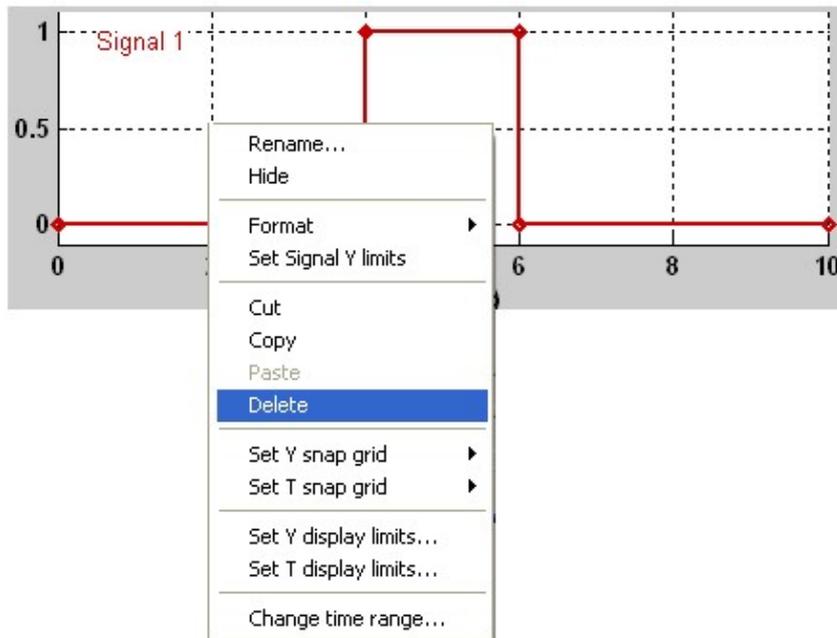


Рисунок 6. Удаление текущего сигнала

Такой сигнал можно подать на вход передаточной функции, как показано на рисунке 7. При этом время моделирования желательно установить равным конечному значению вектора X . На рисунке 5 показан сигнал при времени моделирования, равном 10.

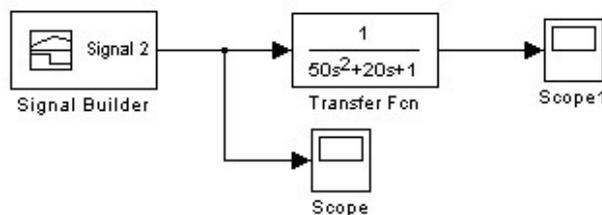


Рисунок 7. Модель с произвольным входным сигналом

Использование блока **From file**

Блок *From File* также находится в стандартной библиотеке *simulink* в разделе *Sources* и служит для получения сигнала из файла.

Модель с использованием данного блока приведена на рисунке 8.

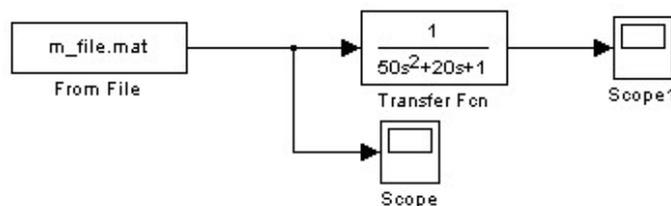


Рисунок 8. Модель с блоком *From file (Model2)*

Чтобы создать файл, выполним в рабочем окне MATLAB следующие команды:

- укажем путь к папке с моделью, например, как `Cd d:\asutp\model;`
- создадим из векторов X , Y двумерный массив, выполнив команду $M = [X; Y];$

- сохраним массив в файле, выполнив команду *Save 'm_file' M*, где *'m_file'* – имя файла. *M* – имя сохраняемого массива;
- кликнем по блоку *from file* и укажем имя файла, как показано на рисунке 9.

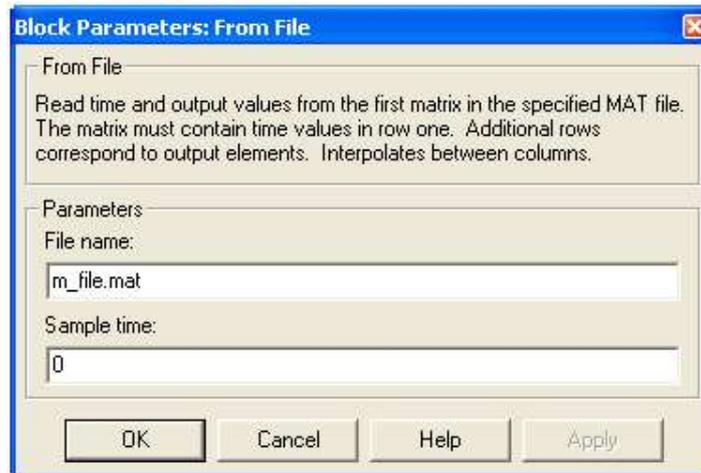


Рисунок 9. Настройка блока *from file*

Обратим внимание, что входной сигнал зависит от параметра *Sample time*. Если значение *Sample time* равно нулю, то входной сигнал будет выглядеть, как на рисунке 10. Заметим, что сигнал отличается от сигнала, приведенного на рисунке 5. Углы получаются сглаженными, а сигнал после последней точки продолжает нарастать.

Если значение *Sample time* = 1, то сигнал будет выглядеть, как на рисунке 11.

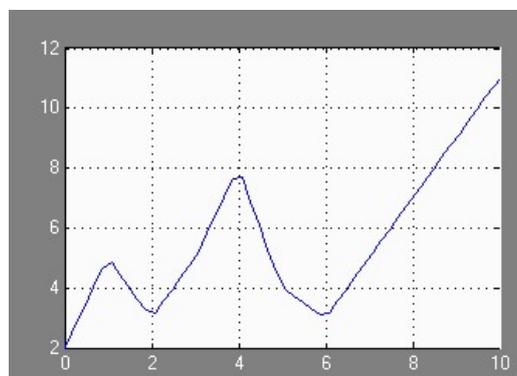


Рисунок 10. Входной сигнал при *Sample time* = 0

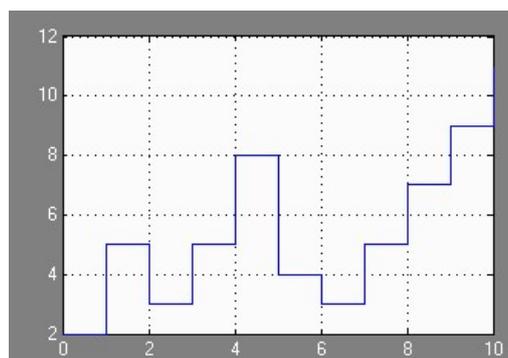


Рисунок 11. Входной сигнал при *Sample time* = 1

Использование блока *From workspace*

Блок находится на той же вкладке, что и ранее рассмотренные блоки.

Если уже имеется двумерный массив, то нет смысла сохранять его в файле. Используя блок *From Workspace* можно (рис. 12) указать непосредственно имя массива.

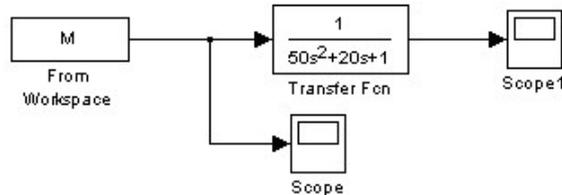


Рисунок 12. Модель с блоком *From Workspace* (*Model3*)

Настройка блока показана на рисунке 13.

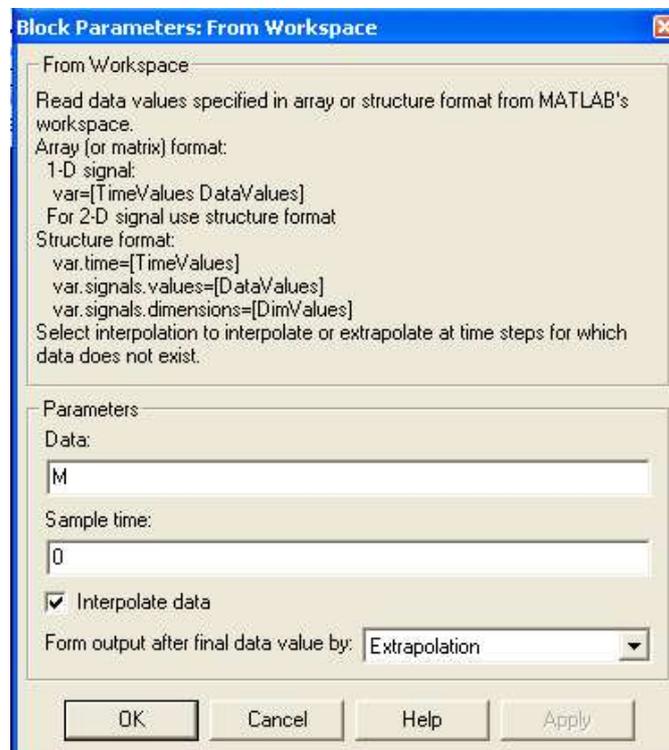


Рисунок 13. Настройка блока *From Workspace*

Если включена опция *Interpolate data* и в раскрывающемся списке выбрана опция *Extrapolation*, то сигналы будут соответствовать сигналам, приведенным на рисунке 10 (при *Sample Time* = 0) и на рисунке 11 (при *Sample Time* = 1).

Значение, выбираемое в раскрывающемся списке «*From output after final data value by*», влияет на «конец» сигнала, если время моделирования больше количества дискретных точек. В частности, если включена опция *Interpolate data* и *Sample Time* = 1, то параметры, выбираемые в раскрывающемся списке, имеют следующий смысл:

- при выборе *Extrapolation* – значение соответствует значению последнего элемента массива;

- при выборе *Setting to Zero* – значение равно нулю;
- при выборе *Holding final value* – значение соответствует значению последнего элемента массива.

Использовавшие команды *Lsim*

Команда *Lsim* позволяет получить сигнал на выходе системы, при изменении входного сигнала по какой-либо известной математической зависимости.

Сначала необходимо отметить, что все команды MATLAB, используемые для построения графиков систем управления (такие как *step*, *impulse*, *nyquist*, *bode* и другие) можно использовать и как функции [5]. Например, если в командном окне MATLAB выполнить строки:

```
W = tf([1], [50 20 1]);
Step(W), grid
```

то построится график, но если в качестве второй строки указать

```
[Y, X] = step(W)
```

то в результате получатся массивы этого графика (массив ординат указывается на первом месте).

Формат команды *Lsim* имеет вид *Lsim(W, U, T)*, где *W* – передаточная функция системы, на вход которой необходимо подать сигнал; *U* – массив ординат входного сигнала; *T* – массив ординат времени.

Рассмотрим пример, когда на ход объекта необходимо подать перевернутую параболу.

Для формирования параболы выполним следующую последовательность команд:

```
X1 = linspace(0,100,100);
A = 0.01; B = 1; Y1 = -A * X1^2 + B * X1;
X = linspace(0,200,200);
Y2 = linspace(0,0,100);
Y = [Y1, Y2];
plot(X, Y), grid
```

В результате получим график, приведенный на рисунке 14.

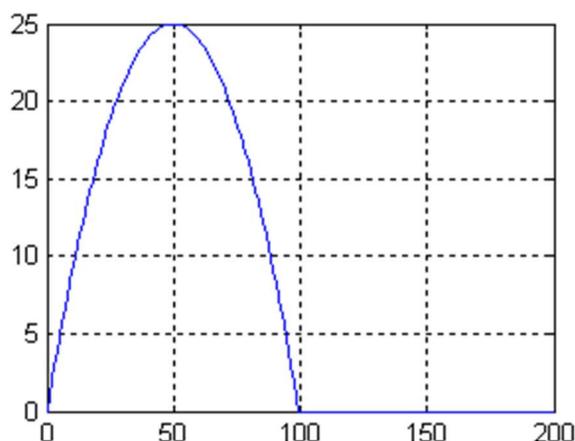


Рисунок 14. Входной сигнал системы

Далее сформируем передаточную функцию и выполним команду *Lsim*:

```
W = tf([1], [50 1]);  
lsim(W, Y, X), grid
```

В результате получим графики, приведенные на рисунке 15.

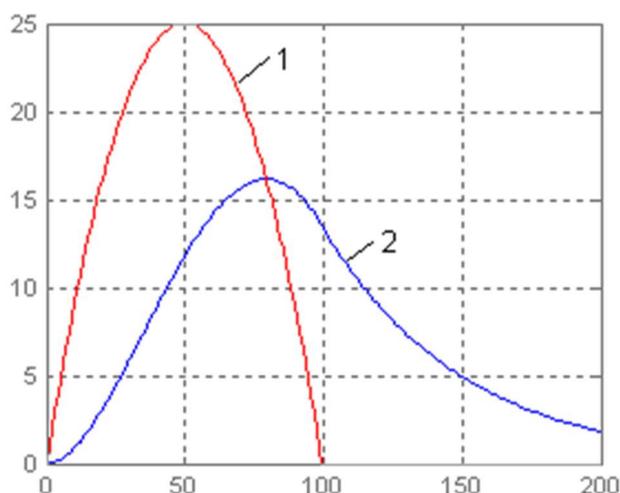


Рисунок 15. Сигналы на входе (1) и выходе (2) системы

Пример использования пакета по многокритериальному анализу систем управления

В заключении рассмотрим формирование случайного сигнала в пакете по многокритериальному исследованию систем управления, разработанного на кафедре АТП БФ ПНИПУ [5]. Данный пакет позволяет работать с различными версиями MATLAB [6].

Для формирования случайного сигнала в пакете необходимо выполнить следующую последовательность действий:

- запустим пакет;
- в поле « $T(K)$ » на форме идентификации укажем необходимый разброс сигнала, например 10;
- выполним меню «Файл» > «Создать» > «Случайный сигнал»;
- кликнем по панели «Дин. Характеристика», чтобы посмотреть на форму сигнала;
- выполнить меню «Файл» > «Передать в MATLAB» > «Оба столбца» и введем имя массива, например *SysXY*;
- перейдем на форму «Работа с моделями *simulink*»;
- выполним меню «Модель» > «Открыть» и откроем модель «*Zaton \ Model3*»;
- запустим модель;
- на форме в поле *Scope* введем *ScopeData2*;
- нажмем кнопку «График».

В результате получим график выходного сигнала, приведенный на рисунке 16



Рисунок 16. График выходного сигнала модели, при случайном сигнале на входе

Из рисунка видно, что чем больше инерционность передаточной функции, тем меньше колебаний в выходном сигнале.

Сохранение рабочей области и переменных в файлах

И последнее, на что стоит обратить внимание, это как можно сохранить созданные сигналы для последующей работы. Для этого можно использовать две возможности MATLAB – это сохранение рабочей области и сохранение переменных.

Сохранение рабочей области подразумевает автоматическое сохранение всех переменных, используемых в данном сеансе работы. Сохранение рабочей области осуществляется из окна рабочей области (*Workspace*) с помощью кнопки «*Save Workspace As*». Восстановление рабочей области осуществляется по кнопке «*Load Data File*».

Сохранение переменных подразумевает сохранение в файле конкретной переменной (вектора, массива и т.д.).

Сохранения переменной, осуществляется командой *Save 'm_file' M*, где *'m_file'* – имя файла, *M* – имя сохраняемой переменной. В результате переменная сохраняется в формате **.mat*. Чтение переменной из файла осуществляется командой *Load M*, где *M* – имя переменной.

Выводы

Подача на вход *Simulink* моделей различных сигналов позволяет осуществлять разностороннее тестирование моделей на предмет выявления различных ошибок и их устойчивой работы.

Описанные выше подходы многократно использовались при исследовании систем управления на практике. Разработанный на кафедре пакет позволяет применять удобный интерфейс, для управления *Simulink* моделями различных версий MATLAB [6]. Так в частности была разработана и протестирована достаточно сложная модель процесса грануляции, насчитывающая более 700 блоков библиотеки *Simulink* [7]. Также, различные сигналы подавались на вход модели сепарации губчатого титана, для выявления ее правильной работы [8].

Список использованных источников и литературы

1. Гайдук Ф. Р., Беляев В. Е., Пьявченко Т. А. Теория автоматического управления в примерах и задачах с решениями в MATLAB. – М.: Лань, 2011. – 464 с.
2. Бильфельд Н. В. Моделирование в реальном режиме времени в среде MATLAB // *Juvenis scientia*. – 2016. № 2. – С. 16-18.
3. Бильфельд Н. В. Пассивная идентификация объектов управления средствами TOOLBOX IDENT // *Juvenis scientia*. – 2016. № 3. – С. 4-7.
4. Дьяконов В. П. MATLAB6/6/1/6/5+SIMULINK 4/5. – М.: Солон-Пресс, 2002. – 768 с.
5. Бильфельд Н. В., Многокритериальное исследование систем управления. – Пермь : ПНИПУ, 2015. – 436 с.
6. Бильфельд Н. В., Митюков Е. А. Одновременная работа с различными версиями пакетов MATLAB как серверов OLE-автоматизации // *Новый университет. Серия: Технические науки*. – 2015. № 9-10 (43-44). – С. 67-70.
7. Затонский А. В. Программные средства глобальной оптимизации систем автоматического регулирования – М: ИЦ РИОР, 2013. – 136 с.
8. Затонский А. В., Бильфельд Н. В. Применение самоорганизующихся систем при управлении сложными процессами // *Проблемы теории и практики управления*. – 2007. №12. – С.70-74.

List of references

1. Gaiduk F. R., Belyaev V. E., Pjavchenko T. A. Theory of automatic control in examples and problems with solutions in MATLAB. - M.: Lan, 2011. – 464 p.
2. Bilfeld N. V. Modeling in real time in the environment of MATLAB // *Juvenis scientia*. – 2016. № 2. – P. 16-18.
3. Bilfeld N. V. Passive identification of asset management tools TOOLBOX IDENT // *Juvenis scientia*. – 2016. №3. – P. 4-7.
4. Dyakonov V. P. MATLAB6/6/1/6/5+SIMULINK 4/5. – M.: Solon-Press, 2002. – 768 p.
5. Bilfeld N. V., Mnogokriterial'noye issledovaniye sistem upravle-niya. – Perm: PNIPU, 2015. – 436 p.
6. Bilfeld N. V., Mityukov Ye. A. Odnovremennaya rabota s razlichny-mi versiyami paketov MATLAB kak serverov OLE-avtomatizatsii // *Novyy universitet. Seriya: Tekhnicheskiye nauki*. – 2015. № 9-10 (43-44). – P. 67-70.
7. Zatonsky A. V. Software for global optimization of automatic control systems – M: IC RIOR, 2013. – 136 p.
8. Zatonsky A. V., Bilfeld N. V. The use of self-organizing systems in the management of complex processes // *Problems of control theory and practice*, – 2007. №12. – P. 70-74.