

МОДЕЛИРОВАНИЕ ЭКСТРЕМАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ В СРЕДЕ MATLAB И SIMULINK КАК СРЕДСТВО АНАЛИЗА ДИНАМИКИ⁶

**В.Г. РУБАНОВ
Д.А. БУШУЕВ**

*Белгородский
государственный
технологический
университет
им. В. Г. Шухова*

e-mail: untame@list.ru

Рассмотрена возможность значительного упрощения процесса исследования динамики экстремальных систем управления с дрейфующими параметрами путем моделирования их в среде Matlab и Simulink. Показана трудоемкость аналитического исследования на примере системы с запоминанием экстремума. Приведены возможные варианты задания экстремальных объектов управления и алгоритмов экстремальных регуляторов в среде SIMULINK. Описана модель системы экстремального управления с динамической частью первого порядка, работающей по методу запоминания экстремума, и результаты, получаемые при ее моделировании в среде SIMULINK и MATLAB.

Ключевые слова: экстремальная, управление, моделирование, Matlab, система, Simulink, дрейф, оптимизация, S-функция.

Повышение энергоэффективности технологических процессов в промышленности часто приводит к необходимости текущей оптимизации, заключающейся в отыскании управляющего воздействия, при котором обеспечивается экстремум некоторых технологических параметров или величин (критерия оптимальности функционирования объекта) отдельных агрегатов. Зачастую такая оптимизация проходит в условиях минимального количества априорной информации, в результате чего необходимо использовать автоматические поисковые системы или системы экстремального управления.

Например, можно рассматривать в качестве экстремальной систему управления величиной вибрации помольно-смесительного агрегата с автоматической балансировкой [1]. Управляющим воздействием является перемещение противовеса, изменяющее уравнивающее воздействие, оптимизируемым параметром – величина вибрации. Алгоритм управления такой системы закладывается в микроконтроллер. Судя по исследованиям на физической модели агрегата [2] ее статическая характеристика является унимодальной и дрейфующей.

Системы экстремального управления представляют собой нелинейные динамические системы, структуру которых в большинстве случаев можно представить в виде, показанном на рис.1.

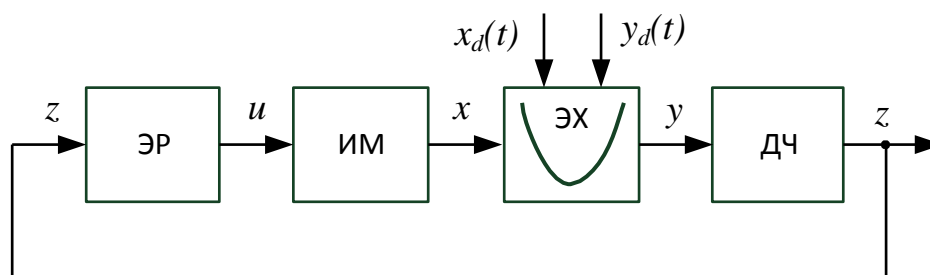


Рис. 1. Структура системы экстремального управления

Экстремальный регулятор (ЭР) воздействует через исполнительный механизм (ИМ) на объект управления, представленный в виде нелинейной экстремальной статической характеристики ЭХ, которая при дрейфе является также функцией времени. Инерционные свойства объекта и измерительных устройств отражает динамическая часть ДЧ. В случае, когда исполнительный механизм воздействует на объект управления через звено, обладающее инерционностью, тогда перед объектом появляется еще одна динамическая часть.

⁶ Работа выполнена в рамках гранта Минобрнауки РФ «Методы проектирования управляемых мобильных логистических средств, обладающих свойством живучести»



Аналитическое исследование процессов в таких системах представляет немалый интерес [3]. Покажем это на примере анализа системы экстремального управления, работающей по методу запоминания экстремума [4].

Экстремальный регулятор системы поиска минимума работает в соответствии со следующими правилами:

1. Запоминающее устройство фиксирует только уменьшающееся значение входного сигнала z , на увеличивающееся значение оно не реагирует;
2. Условие переключения сигнум-реле записывается в виде:

$$u = k_1 \text{sign}(z_n + z_{\min} - z),$$

где z_{\min} – минимальное значение, запомненное в запоминающем устройстве;
 z_n – зона нечувствительности сигнум-реле.

3. После срабатывания сигнум-реле, запомненное значение z_{\min} сбрасывается, и запоминание начинается снова.

Для упрощения анализа работы такой системы аналитическим методом [4], представим динамическую часть звеном первого порядка с постоянной времени T_1 , и пренебрежем инерционностью исполнительного механизма постоянной скорости k_1 .

В этом случае дифференциальное уравнение объекта запишется в виде:

$$T_1 \frac{dz}{dt} + z = f(x, t).$$

При осуществлении подстановки $\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt}$, где $\frac{dx}{dt} = \pm k_1$ в уравнение (2), получим следующее уравнение

$$\pm T_1 k_1 \frac{dz}{dx} + z = f(x, t).$$

Уравнение экстремальной характеристики типа «парабола» имеет вид:

$$y = f(x, t) = k(t)(x - x_d(t))^2 + y_d(t),$$

где $k(t)$ – отражает наклон ветвей параболы, $x_d(t)$ – горизонтальный дрейф экстремальной характеристики, $y_d(t)$ – вертикальный дрейф экстремума.

Таким образом, уравнение движения записывается в виде:

$$\pm T_1 k_1 \frac{dz}{dx} + z = k(t)(x - x_d(t))^2 + y_d(t).$$

Если регулятор включил исполнительный механизм на увеличение входного сигнала объекта управления, тогда в левой части уравнения (5) будет знак плюс.

При отсутствии дрейфа параметров k , x_d и y_d уравнение (5) является линейным неоднородным дифференциальным уравнением вида:

$$y' + P(x)y = Q(x)$$

Общее решение такого уравнения записывается в виде:

$$y(x) = e^{-\int P(x)dx} \left(\int Q(x)e^{\int P(x)dx} dx + C_0 \right)$$

Алгоритм построения переходного процесса следующий:

1. Получаем решение уравнения (5) по формуле (7)

$$z(x) = C_0 e^{-\frac{x}{T_1 k_1}} + y_d + k \left[x^2 - x(2x_d + 2T_1 k_1) + 2T_1^2 k_1^2 + 2T_1 k_1 x_d + x_d^2 \right],$$

2. Находим произвольную постоянную C_0 из начальных условий в момент времени $t=0$, т.е. подставляя $x=x_0$, $z=z_0$ в уравнение (8).

$$C_0 = e^{\frac{x_0}{T_1 k_1}} (z_0 - y_d - k \left[x_0^2 - x_0(2x_d + 2T_1 k_1) + 2T_1^2 k_1^2 + 2T_1 k_1 x_d + x_d^2 \right]).$$

3. Получаем кривую движения $z(x)$ до момента реверса.

4. Определяем момент $x=x_e$, когда система достигает минимума, для этого находим x_e из уравнения $\frac{dz}{dx} \Big|_{x=x_e} = 0$ при постоянной C_0 .

5. Находим $z=z_e$, путем подстановки найденного в п.4 значения x_e в уравнение $y=f(x)$.

6. Определяем момент $z=z_r$ реверса $z_r = z_e - z_n$.

7. Определяем x_r , подставив z_r , в уравнение (8) при постоянной C_0 .

8. Повторяем алгоритм, начиная с п.1., но уже в качестве начальных условий берем полученные значения x_r, z_r и учитываем, что после реверса меняется знак в левой части уравнения (5), что влечет за собой получение нового решения уравнения (5) по формуле (7), либо использование уравнения (8), тогда с момента реверса необходимо откладывать значения x в сторону отрицательной оси абсцисс.

Таким образом, на двулистной фазовой плоскости $x-z$ по этапам циклично строится переходный процесс, пока в результате построения не получится замкнутая петля, т.е. до установления режима периодических колебаний вокруг экстремума.

Например, если нелинейная часть объекта описывается уравнением

$y = f(x) = 0.2(x - 5)^2 + 10$, а динамическая часть представляет собой аperiodическое звено с постоянной времени 100 с, то, выполняя выше описанный алгоритм, получим следующую траекторию движения системы (рис.2). Движение началось с начальных условий $x=0.5, y=f(x)=14.05$. В результате цикл алгоритма до установления автоколебаний повторяется минимум 7 раз.

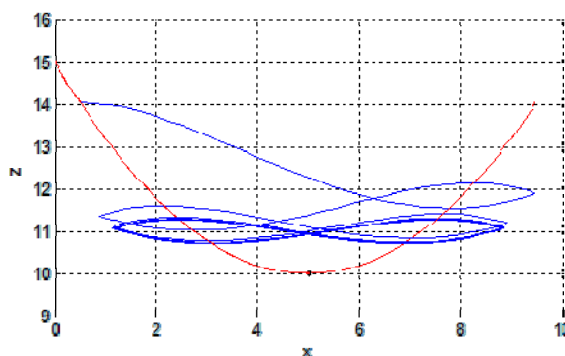


Рис. 2. Траектория движения системы на фазовой плоскости «вход-выход» объекта

Как видно, аналитическое исследование экстремальных систем даже в случае симметричной (относительно минимума) статической характеристики и с динамической частью первого порядка представляет собой весьма трудоемкую процедуру. Следует отметить, что исследование динамики экстремальных систем с помощью приближенных методов (например, методом Эйлера [4]) требует решения такого же количества дифференциальных уравнений и выбора начальных условий, как и в вышеописанном алгоритме.

Если же порядок динамической части системы повышается, статическая характеристика является несимметричной, добавляется дрейф параметров от времени и усложняется алгоритм управления, то задача исследования динамики становится действительно сложно разрешимой даже численными методами.

Современное средство имитационного моделирования MATLAB Simulink позволяет значительно упростить анализ динамики сложных экстремальных систем при различных алгоритмах работы экстремальных регуляторов. Такая возможность обеспечивается благодаря тесной интеграции MATLAB и Simulink, расширяющей их функциональные возможности и использованию технологии S-функций, при которой пользователем программируется на одном из языков программирования (MATLAB, C, C++, Ada, Fortran) собственный библиотечный блок «S-function», за счет чего можно задавать алгоритмы любой степени сложности [5].

Ниже представлен пример модели (рис.3), созданной в среде Simulink, для исследования динамики экстремальных систем, а также отражены варианты задания экстремальных объектов управления и алгоритмов управления экстремальных регуляторов на примере системы с запоминанием экстремума.

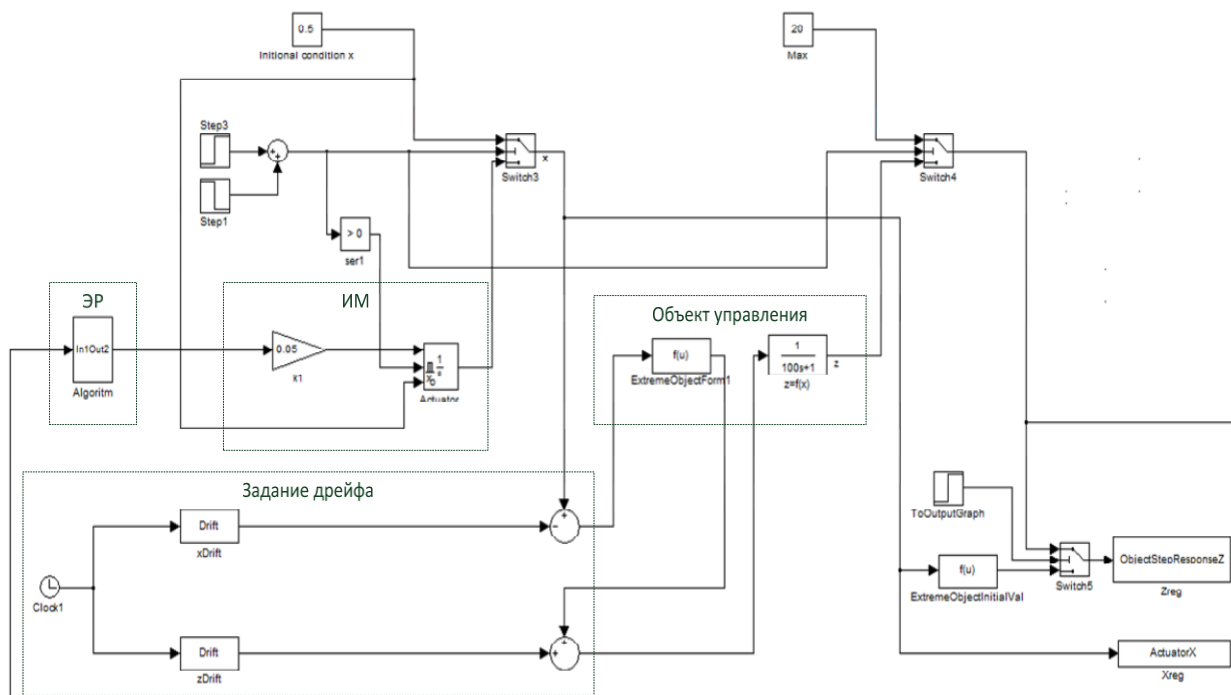


Рис. 3. Модель системы экстремального управления в среде Simulink

В данном примере экстремальная статическая характеристика объекта управления имеет симметричную относительно экстремума зависимость $y = f(x) = 0.2(x - 5)^2 + 10$, которая может быть реализована выражением на языке C с помощью блока «Fcn» раздела User Functions (Блок «ExtremeObjectForm1» на рис.3). Если же статическая характеристика несимметрична относительно экстремума и необходимо «кусочное» задание функции (например, для топки левая ветвь статической характеристики до экстремума имеет вид параболы, а правая – вид гиперболы), то в этом случае необходимо программировать собственный библиотечный блок «S-function», используя технологию S-функций. Динамическая часть объекта управления $W(s)$ формируется из стандартных блоков Simulink. Она может быть любого порядка и располагаться в любом месте относительно нелинейности $y=f(x)$. В рассмотренной модели (рис.3) динамическая часть представлена звеном

$$\text{первого порядка } W(s) = \frac{1}{Ts + 1}.$$

Исполнительный механизм представлен в виде интегратора со сбросом (Блок «Actuator»), скорость которого задается усилителем $k1$ (см. рис. 3).

В модели реализовано включение экстремального регулятора и исполнительного механизма через определенное время после начала работы системы (или моделирования), для того чтобы исключить переходные процессы, обусловленные его динамической частью. Это реализуется посредством блоков step1 и step3, задающих время включения регулятора. Начальные условия (начальное положение ИМ) x_0 для движения задается в блоке «Initial Condition x».

Дрейф параметров $x_d(t)$ и $y_d(t)$ можно задать любой сложности также посредством программируемых пользователем блоков «S-function». В данном примере линейный временной дрейф с однократной сменой направления дрейфа реализован в блоках «xDrift» и «zDrift», представляющих собой S-функцию, написанную на языке программирования MATLAB. Механизм маскирования подсистем позволяет оформить S-функцию как полноценный библиотечный блок, т.е. снабдить подсистему собственным окном задания параметров, пиктограммой, справочной системой и атрибутами. За счет этого механизма в блоках «xDrift» и «zDrift» параметры дрейфа задаются в соответствующих окнах диалога. Время t , определяющее дрейф, поступает с блока Clock1.

Экстремальный регулятор можно реализовывать с помощью стандартных блоков Simulink, как например, реализован регулятор, работающий по методу запоминания экстремума (рис. 4).

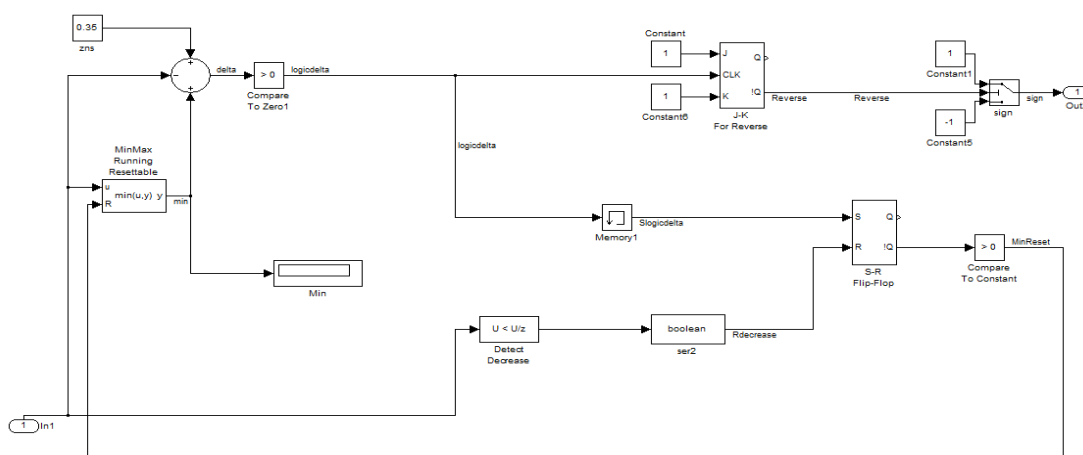


Рис. 4. Задание алгоритма работы экстремального регулятора, работающего по методу запоминания экстремума с помощью стандартных блоков Matlab

Такой регулятор имеет элемент сравнения, задающий условие переключения (1), запоминающее минимум устройство со сбросом «MinMax Running Resettable», сигнал-реле, реализованное на J-K триггере, работающем в режиме Т-триггера и блок сигнатуры sign, осуществляющего переключение направления вращения вала исполнительного механизма. Запоминание только уменьшающегося значения и сброс запоминающего устройства при реверсе реализовано на элементе DetectDecrease, элементе памяти Memory1, RS-триггере и вспомогательных блоках для преобразования типов данных и выполнения логических операций. Алгоритм работы регулятора помещен в подпрограмму «Algorithm» (Рис.3).

Применение программируемых блоков «S-function» позволяет облегчить задание алгоритма работы экстремального регулятора. Описание механизма создания таких блоков приводится в [5]. Основная процедура ответного вызова (так называемый callback-метод), определяющая выходной сигнал блока, выглядит следующим образом (с комментариями):

```
function sys=mdlOutputs(t,x,u)
% Извлечение значений предыдущего шага
LetsWork = get_param(gcb,'UserData');
%Запоминаем только уменьшающиеся z
if u<LetsWork.zpred
    LetsWork.zmin = u;
End
%Условие переключения
sign = zn+LetsWork.zmin-u;
if sign<0 && LetsWork.f1 == false
    LetsWork.f1 = true;
%Выполнение реверса ИМ
if LetsWork.upred == 1
    LetsWork.upred = -1;
else
    LetsWork.upred = 1;
end
LetsWork.zmin = 1e+307; %Сброс запомненного значения
end;
%Восстановление флага Values.f1 для того чтобы можно было опять
%реверсировать
if sign>0
    LetsWork.f1 = false;
end
LetsWork.zpred = u; %Вывод
sys = LetsWork.upred;
%Сохраняем значения для следующего шага
set_param(gcb,'UserData',LetsWork);
end
```

Следует отметить, что за счет описанного выше механизма маскирования параметр зоны нечувствительности сигнум-реле zn задается в диалоговом окне блока и используется в S-функции.

Применение функций `set_param` и `get_param` позволяет сохранять и считывать любые значения в область `UserData` пользовательского блока. В данном случае сохраняется структура `LetsWork`, которая содержит предыдущее значение направления перемещения `upred` (до входа в блок), минимальное значение `zmin` и т.д.

Как видно, задание алгоритма поиска с помощью технологии S-функций намного проще, чем с помощью стандартных блоков (рис.4), однако следует отметить, что работа алгоритмов экстремального регулятора с помощью стандартных блоков при одинаковом значении максимального шага моделирования происходит быстрее, чем при использовании блоков `Matlab S-function`. Быстродействие можно повысить применяя программируемые пользователем S-функции более высокого уровня, например, написанные на языках `C` или `Fortran` и откомпилированные в исполняемый машинный код.

Итак, модель экстремальной системы создана, выходными данными модели будут вектор значений исполнительного механизма X , вектор выходного сигнала экстремальной системы Z и вектор времени T , которые в виде соответствующих структур передаются в рабочую область (`workspace`) среды `Matlab` посредством блоков «`ToWorkspace`» (см. рис.3) для дальнейшего исследования и построения переходных процессов.

После выполнения моделирования автоматически запускается `m-file` из среды `Matlab`, который строит соответствующие смоделированные переходные процессы, например, для рассмотренного выше примера (рис.5) и статические характеристики с учетом дрейфа параметров и определением минимального значения.

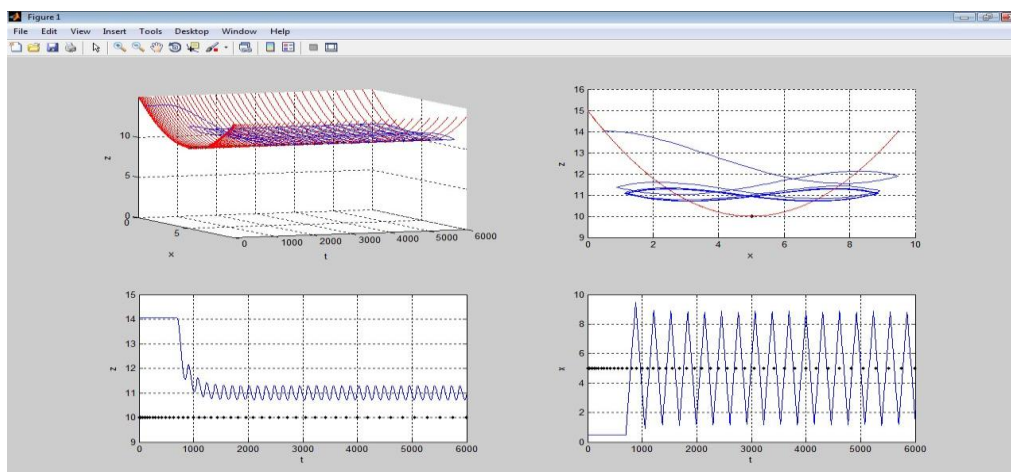


Рис.5. Построение графиков переходного процесса по результатам моделирования в среде `Simulink` ($y=f(x)=0.2(x-5)+10$, $T_1=100$, дрейф отсутствует)

Отображение результатов в трехмерном пространстве (в системе координат xtz) с указанием точек минимума статической характеристики наглядно демонстрирует динамику системы при дрейфующих параметрах статической характеристики (рис.6).

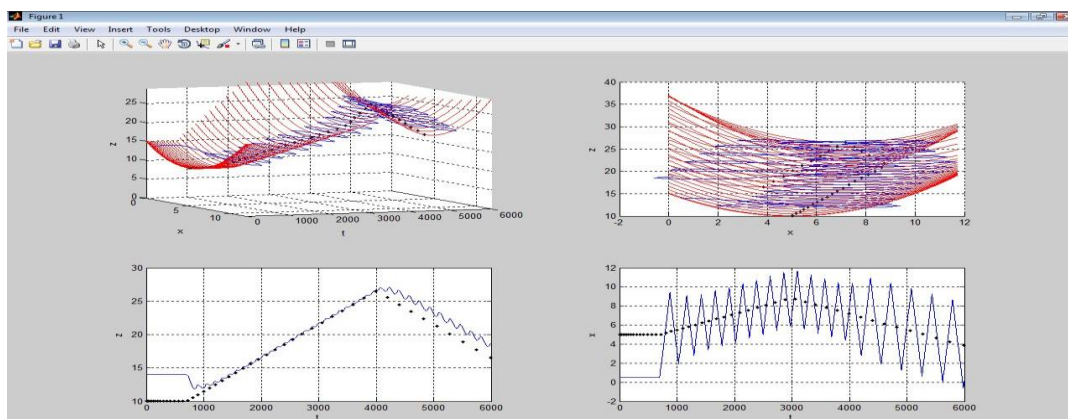


Рис. 6. Построение графиков переходного процесса по результатам моделирования в среде `Simulink` ($y=f(x)=0.2(x-5)+10$, $T_1=100$, линейный дрейф параметров $xd(t)$ и $yd(t)$)

мальной системы: потери на поиск, амплитуда автоколебаний и т.д.

Таким образом, исследование экстремальных систем управления при наличии дрейфа, несимметричности статической характеристики и высоком порядке динамической части представляет собой значительные сложности, однако использование среды Simulink и Matlab благодаря применению программируемых пользователем блоков «S-function» позволяет существенно облегчить задачу системного анализа и сделать результаты более наглядными. Кроме того использование расширения GUI (Graphical User Interface) позволяет сделать конечный программный продукт независимым (возможность работы без установки среды MATLAB) с возможностью добавления разнообразных инструментов ввода и вывода информации [6]. В результате получается мощный инструмент для анализа динамики экстремальных систем с интуитивно понятным графическим интерфейсом.

Литература

1. Пат. 114875 Российская Федерация, В 02С 17/00. Помольно-смесительный агрегат с автоматической балансировкой / Рубанов В.Г., Севостьянов В.С., Уральский В.И., Стативко А.А., Бушуев Д.А., Стативко С.А.; заявитель и патентообладатель БГТУ им. В.Г. Шухова
2. Стативко, С.А. / Исследование вибрации центробежного помольного агрегата в зависимости от загрузки / Стативко С.А., Рубанов В.Г., Севостьянов В.С., Уральский В.И. // Актуальные проблемы науки: сб. докл. Междунар. науч.-практ. конф., Кузнецк, 30 сент. 2011 г. / Кузнецкий инст. информ. и управл. техн. – Кузнецк: ИП Тугушев С.Ю., 2011.-Т.3.- С. 99-102.
3. Справочник по теории автоматического управления // Под ред. А.А.Красовского.- М.: Наука, 1987.
4. Казакевич, В.В., Системы автоматической оптимизации. / Казакевич В.В., Родов А.Б. – М.: Энергия, 1977.- 288 с.
5. Черных, И.В. SIMULINK: среда создания инженерных приложений// Под общ. Ред. к.т.н. В.Г. По темкина. – М.: ДИАЛОГ-МИФИ, 2003. – 496 с.
6. Рубанов, В. Г. Применение среды MATLAB и расширения GUI для идентификации сложного объекта / В. Г. Рубанов, В. А. Порхало // Известия ТулГУ. – 2011. – Вып. 6. – Ч. 2. – С.14 – 20.

SIMULATION OF EXTREMUM SEEKING CONTROL SYSTEMS USING MATLAB AND SIMULINK AS A MEAN OF DYNAMICS ANALYSIS

V.G. RUBANOV D.A. BUSHUEV

Belgorod state technological university named after V.G. Shoukhov

*e-mail:
untame@list.ru*

The possibility of considerable simplification of the process of studying the dynamics of extremum seeking control systems with drifting parameters by simulation them in MATLAB and SIMULINK is discussed. The complexity of the analytical studies in systems with extremum memorizing is illustrated. The possible versions of setting extremum objects and algorithms of extremum regulators in the SIMULINK are presented. Here a model of the extremum seeking control system with dynamic part of the first order, which uses the method of extremum memorizing and the results obtained in its modeling in SIMULINK and MATLAB are described.

Keywords: extremum, control, simulation, MATLAB, system, SIMULINK, drift, optimization, S-function.