
ЗАКЛЮЧЕНИЕ

Работа посвящена актуальному, весьма перспективному и быстро развивающемуся на основе использования автоматного принципа обработки информации направлению в области построения цифровых управляющих устройств и разработки их внутренней структуры для систем автоматического управления технологическими процессами.

Основной целью работы являлось повышение качества функционирования систем автоматического регулирования путем построения и исследования математических моделей адаптивных автоматизированных структурно перестраиваемых систем управления обработки информации и проектирования.

Отметим наиболее важные результаты, полученные в работе.

1. Установлено соответствие между булевой моделью логики перестраиваемых структур, изотропными средами и устройствами управления, заключающееся в том, что их взаимосвязь приводит к построению систем автоматического регулирования, способных функционировать при неполной информации об объекте и автоматически выбирать наилучший закон управления из заданного класса.

2. Разработана булева модель логики перестраиваемых структур для определенных классов булевых функций.

3. Разработаны изотропные среды, обеспечивающие реализацию систем булевых формул из классов неповторных упорядоченных и неупорядоченных булевых функций, а также из класса повторных упорядоченных произвольных нормальных булевых формул из букв и систем булевых формул как с пропусками аргументов, так и без них.

4. Разработаны функционально необходимые модели регуляторов, в том числе с перестраиваемой структурой, с широкими возможностями и большей приспособленностью для практической реализации.

5. Обеспечено повышение качества регулирования и расширение функциональных возможностей систем управления.

6. Разработано специализированное программное обеспечение структурно перестраиваемых автоматизированных систем управления технологическим процессом, в том числе система автоматизированного проектирования приборов и оборудования.

Рекомендации. Дальнейшие исследования могут развиваться в следующих направлениях:

1) исследование предельных возможностей разработанных ячеек в матричных и иерархических структурах;

2) разработка новых многофункциональных логических модулей и изотропных сред, охватывающих полный класс булевых функций;

3) создание высокоинтеллектуальных управляющих устройств с перестраиваемой структурой;

4) создание так называемых «кинематических ячеистых автоматов» – наноустройств, состоящих из идентичных блоков. Такие автоматы могут наращивать функциональность путем подключения дополнительных блоков, т.е., по сути, собирать себя сами.

В итоге системы получают способность менять свою структуру и подключать к решению задачи необходимые узлы в зависимости от возникающей ситуации. Возможно, именно это и есть те пути, по которым техника будет продвигаться к созданию сплошных эволюционирующих и адаптирующихся вычислительных сред, управляющих технологическими процессами.

Приложение 1

ТАБЛИЦЫ ДЛЯ ФОРМУЛ S-СТРУКТУРЫ И Ф-ИМПЛИКАНТ

Таблица П.1.1

Ячейка 1		Ячейка 2		f
α_1	S_1	α_2	S_2	
0	0	0	0	0
0	0	0	1	0
0	0	1	0	A_1A_2
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	A_2
0	1	1	1	0
1	0	0	0	A_1
1	0	0	1	A_1
1	0	1	0	$A_1 \vee A_2$
1	0	1	1	A_1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	A_2
1	1	1	1	0

Таблица П.1.2

S_1	S_2	S_3	α_1	α_2	α_3	f	S_1	S_2	S_3	α_1	α_2	α_3	f
1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	$A_1A_2A_3$	1	0	0	0	0	1	A_2A_3
0	0	0	0	1	0	A_1A_2	1	0	0	0	1	0	A_2
0	0	0	0	1	1	$A_1A_2 \vee A_3$	1	0	0	0	1	1	$A_2 \vee A_3$
0	0	0	1	0	0	A_1	1	0	0	1	0	0	0

Продолжение табл. П.1.2

1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	1	0	1	$A_1 \vee A_2 A_3$	1	0	0	1	0	1	$A_2 A_3$
0	0	0	1	1	0	$A_1 \vee A_2$	1	0	0	1	1	0	A_2
0	0	0	1	1	1	$A_1 \vee A_2 \vee A_3$	1	0	0	1	1	1	$A_2 \vee A_3$
0	0	1	0	0	0	0	1	0	1	0	0	0	0
0	0	1	0	0	1	0	1	0	1	0	0	1	0
0	0	1	0	1	0	$A_1 A_2$	1	0	1	0	1	0	A_2
0	0	1	0	1	1	$A_1 A_2$	1	0	1	0	1	1	A_2
0	0	1	1	0	0	A_1	1	0	1	1	0	0	0
0	0	1	1	0	1	A_1	1	0	1	1	0	1	0
0	0	1	1	1	0	$A_1 \vee A_2$	1	0	1	1	1	0	A_2
0	0	1	1	1	1	$A_1 \vee A_2$	1	0	1	1	1	1	A_2
0	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	0	0	0	1	$A_1 A_3$	1	1	0	0	0	1	A_3
0	1	0	0	1	0	0	1	1	0	0	1	0	0
0	1	0	0	1	1	$A_1 \vee A_3$	1	1	0	0	1	1	A_3
0	1	0	1	0	0	A_1	1	1	0	1	0	0	0
0	1	0	1	0	1	$A_1 \vee A_3$	1	1	0	1	0	1	A_3
0	1	0	1	1	0	A_1	1	1	0	1	1	0	0
0	1	0	1	1	1	$A_1 \vee A_3$	1	1	0	1	1	1	A_3
0	1	1	0	0	0	0	1	1	1	0	0	0	0
0	1	1	0	0	1	0	1	1	1	0	0	1	0
0	1	1	0	1	0	0	1	1	1	0	1	0	0
0	1	1	0	1	1	0	1	1	1	0	1	1	0
0	1	1	1	0	0	A_1	1	1	1	1	0	0	0
0	1	1	1	0	1	A_1	1	1	1	1	0	1	0
0	1	1	1	1	0	A_1	1	1	1	1	1	0	0
0	1	1	1	1	1	A_1	1	1	1	1	1	1	0

Таблица П.1.3

Ф-ИМПЛИ- КАНТЫ	$A_1 A_2 A_3$	$A_1 A_4 A_6$	$A_4 A_7 A_8 A_9$	$A_5 A_7 A_8$	$A_9 A_{10} A_{12} A_{14}$	$A_9 A_{13}$	$A_{10} A_{11} A_{12}$	$A_{13} A_{14}$
1	2	3	4	5	6	7	8	9
φ_1	1	0	1	0	0	0	1	1
φ_2	1	0	0	1	1	0	0	0
φ_3	1	0	0	1	0	1	0	0

Продолжение табл. П.1.3

1	2	3	4	5	6	7	8	9
φ_4	1	0	0	1	0	0	1	1
φ_5	0	1	0	0	1	0	0	0
φ_6	0	1	0	0	0	1	0	0
φ_7	0	1	0	0	0	0	1	1

Таблица П.1.4

Ф-импли- каны	$A_1A_4A_6$	$A_5A_7A_8$	$A_9A_{10}A_{12}A_{14}$	A_9A_{13}
φ_2	0	1	1	0
φ_3	0	1	0	1
φ_4	0	1	0	0
φ_5	1	0	1	0
φ_6	1	0	0	1
φ_7	1	0	0	0

Приложение 2

СТРУКТУРНАЯ СХЕМА АЛГОРИТМА, ТЕКСТ ПРОГРАММЫ И ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ ДЛЯ РАСЧЕТА ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ПИ-РЕГУЛЯТОРА

Алгоритм расчета (рис. П.2.1) реализован на языке MatLab, так как данная среда программирования является наиболее удобной. Код программы представлен в виде m-файла. Программа выводит на дисплей график зависимости оптимальных параметров настройки (ОПН) регулятора согласно заданным требованиям, предъявляемым к системе, на котором необходимо выбрать точку, соответствующую желаемому интегральному критерию качества регулирования (максимум – первый, 95% максимума вправо – второй). Затем рассчитывается замкнутая система автоматического регулирования с выбранными параметрами настройки регулятора. В завершении работы выводится график переходного процесса замкнутой АСР.

Текст программы

```
%=====
%Задание параметров объекта регулирования
%=====
K=6; %Коэффициент усиления
T=97.7; %Постоянная времени
tau=23.4;%Величина запаздывания
%=====
%Заданные требования к системе регулирования
%=====
m=0.305; %Степень колебательности
%=====
%Расчет ОПН ПИ-регулятора
%=====
w=[0:0.001:0.03];%Частота для расчета
[TDddd ST]=size(w);%Определения количества ...
...рассчитываемых точек
```

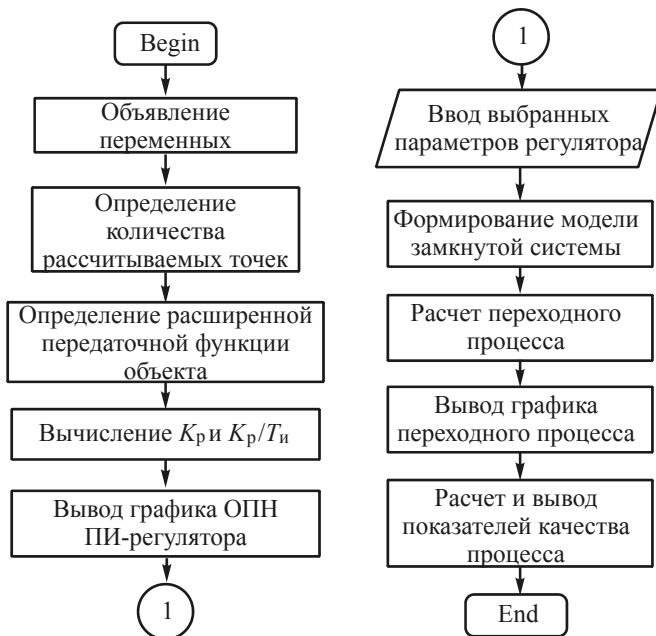


Рис. П.2.1. Структурная схема алгоритма

```

WmXobPI=(K*exp(-tau*(1i-m).*w))./(T*...
... ((1i-m).*w)+1);% Расширенная передаточная...
... функция объекта
C1=0;
C0=0;
C1=(-m.*imag(WmXobPI)-real(WmXobPI))./ ...
... (abs(WmXobPI).*abs(WmXobPI))); ...
...%Коэффициент усиления регулятора
%=====
%Цикл для вычисления Kp/Ti
%=====
for jj=1:ST
    if (imag(WmXobPI(jj))<=0)
        C0(jj)=((-m*m+1).*imag(WmXobPI(jj)).* ...
...w(jj))./(abs(WmXobPI(jj)).*abs(WmXobPI(jj))));
    else
    
```

```

        C0(jj)=((- (m*m+1) .* ...
...imag(WmXobPI(jj)).*w(jj))./ ...
... (abs(WmXobPI(jj)).*abs(WmXobPI(jj))));
    end;
end;

plot(C1,C0,'g');grid on; %Вывод результата...
... расчета ОПН ПИ-рег
%-----
%Интерактивный ввод результатов вычисления
%с графика
%-----
C0=input('Kp/Tu= ?');
C1=input('Kp= ?');
Kрпи=C1;
Типи=Kрпи/C0; %Расчет постоянной...
... времени изодрома

%-----
%Расчет замкнутой АСР с найденными близкими
% к оптимальным параметрами ПИ-регулятора
%-----
disp('----- Start System Computation -----');
%передаточная функция ПИ-регулятора в ЛТИ форме
sys1=tf(Kрпи*[Типи 1],[Типи 0], ...
...'inputname','1in','outputname','1out');
%передаточная функция объекта в ЛТИ форме
sys2=tf(K,[T 1],'td',tau,'inputname', ...
...'2in','outputname','2out');
sys=append(sys1,sys2);%агрегатированная ...
... модель без перекрестных связей
Q=[2 -1 0;1 2 0];%матрица связей
inputs=[2];%матрица внешних входов
outputs=[2];%матрица внешних выходов
sysc=connect(sys,Q,inputs,outputs);
%модель в пространстве состояний с...
...учетом перекрестных связей между блоками
[y x]=step(sysc); %расчет переходной...

```



```
... характеристики замкнутой системы
plot(x,y); %вывод переходной...
... характеристики замкнутой системы

%-----
%Определение показателей качества процесса
%регулирования
%-----
[si si2]=size(y);
[A1 c1]=max(y);
[A2 c2]=min(y(c1:1:si));
[A3 c3]=max(y(c1+c2:1:si));
sigma=(abs(A2-y(si))/(A1-y(si)))*100
%перерегулирование
psi=((A1-y(si))-(A3-y(si)))/(A1-y(si))
%степень затухания
FIdin=(A1-y(si))
%максимальная динамическая ошибка
YY=y-y(si);
delta=0.05*y(si);
zz=abs(YY)-delta;
[zn zx]=min(abs(zz));
Treg=x(zx)
%Время регулирования
```

Приложение 3

СТРУКТУРНАЯ СХЕМА АЛГОРИТМА, ТЕКСТ ПРОГРАММЫ И ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ ИНТЕГРАЛЬНОГО РЕГУЛЯТОРА С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ

Алгоритм функционирования интегрального регулятора с перестраиваемой структурой (рис. П.3.1) реализован на языке MatLab, так как данная среда программирования является наиболее удобной и более прикладной для решаемой задачи. Код программы представлен в виде двух m-файлов, реализующих соответственно две функции.

Первая представляет из себя так называемую *S*-функцию среды MatLab, которые предназначены для построения пользовательских блоков системы Simulink и позволяют описывать части системы или всю систему на алгоритмическом языке. Реализованная функция непосредственно выполняет алгоритм представленные в разделе 5.6, формируя управляющее воздействие из множества значений $\{1, 0, -1\}$.

Вторая функция реализует МЛМ *L*-структуры и вызывается первой для вычисления ряда систем булевых формул.

S-функция носит название `isps_mlm` и в качестве аргументов принимает массив значений состоящий из коэффициента наклона прямой в фазовом пространстве c , постоянных величин служащих для формирования внутренних переменных $V1$, $V2$ и постоянных времени $T0$, $T1$ двух линейных структур участвующих при решении дифференциального уравнения.

Текст программы

```
function [sys,x0,str,ts] = isps_mlm(t, x, u, ...
... flag, c, V1, V2, T0, T1)
switch flag,
    % Инициализация%
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes;
    % Вычисление вектора выхода %
    case 3,
        sys=mdlOutputs(t,x,u,c,V1,V2,T0,T1);
    % Пустые флаги%
    case {1, 2, 4, 9}
```

```

sys=[];
% Ошибочный вход %
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
    
```

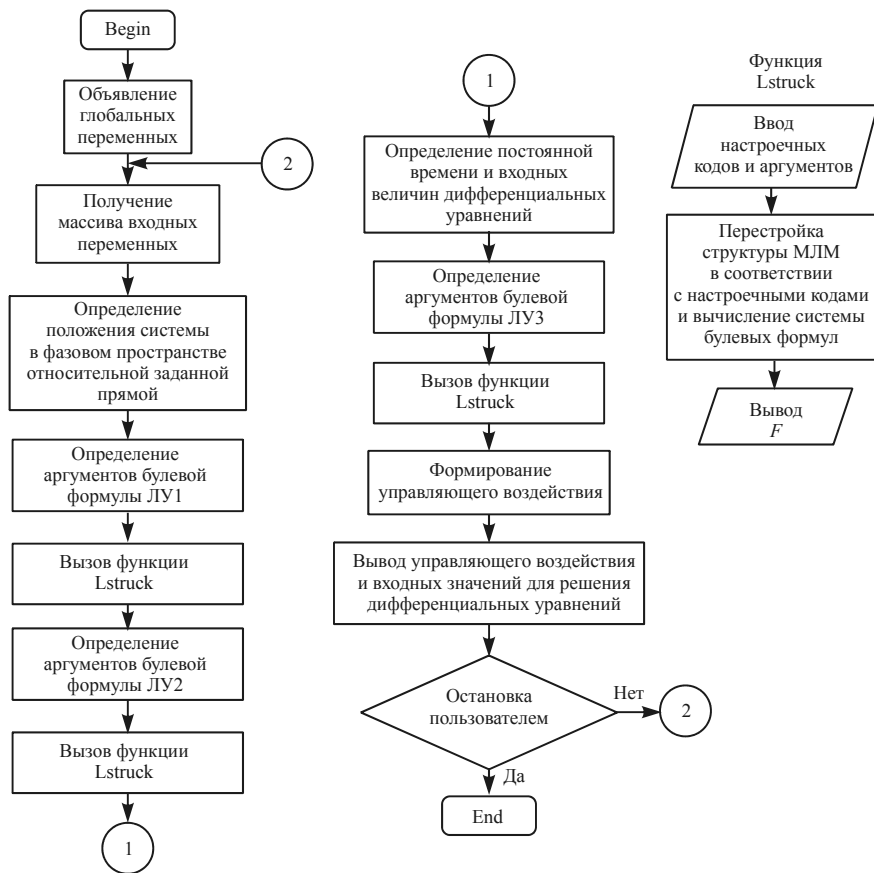


Рис. ПЗ.1. Структурная схема алгоритма

```

% Функция инициализации
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
    
```

```

sizes.NumContStates = 0; %число непрерывных ...
... состояний
sizes.NumDiscStates = 0; %число дискретных ...
... состояний
sizes.NumOutputs     = 4; %число выходов
sizes.NumInputs      = 4; %число входов
sizes.DirFeedthrough = 1; %число прямых проходов
sizes.NumSampleTimes = 1;

sys = simsizes(sizes); %определение размера ...
... векторов переменных модели
x0 = []; %установка начальных значений для ...
... всех переменных состояний
str = []; %пустая матрица
ts = [-1 0]; %Инициализация матрицы периодов ...
... квантования
%Объявление переменных
global s, b, ResetTemp, sigma1, sigma2, ...
... DeltaT, ro, U1, U2, U3, U4, U5, gamma, ...
... vv, tetta, fi, nyu, ksi, SIGMA, ALPHA, ...
... ВЕТТА, lyamda, Tsig, eps, ss, dzita;
%Начальные условия
eps = 0.00528;
Tsig = 5;
[s ss b ResetTemp sigma1 sigma2 DeltaT ro U1 ...
... U2 U3 U4 U5 gamma vv tetta fi nyu ksi ...
... SIGMA ALPHA ВЕТТА lyamda dzita] = 0
% Окончание функции инициализации
%-----
% Функция вывода
%-----
function sys=mdlOutputs(t,x,u,c,V1,V2,T0,T1)
%с - коэффициент прямой в фазовой плоскости
%V1, V2 - константы
%T0,T1 - постоянные времени
tetta=u(3); %решение дифференциального ...
... уравнения 1

```

```
fi=u(4); %решение дифференциального уравнения 2
%Уравнение прямой в фазовой плоскости
ss=u(2)+c*u(1);
s=sign(ss)*u(1);
ss=s*sign(u(1));
%Вычисление вспомогательных переменных
U1=~(sign(abs(sign(s)-1)));
nyu=U1*u(1);
U2=~SIGMA;
ksi=U2*nyu;
if tetta>eps
    dzita=1;
else
    dzita=0;
end
%Решение системы булевых формул с помощью ...
... МЛМ L-структуры
ResetTemp=Lstruck(~abs(sign(s)),dzita, ...
... 0,0,1,0,0);
U4=ResetTemp(1);
ALPHA=U4*fi;
lyamda=ALPHA+ВЕТТА;
if lyamda>eps
    SIGMA=1;
else
    SIGMA=0;
end
%Решение системы булевых формул с помощью МЛМ...
... L-структуры
ResetTemp=Lstruck(abs(sign(nyu)), ...
... ~SIGMA,0,0,1,0,0);
U3=ResetTemp(1);
%Входная величина для решения ...
... дифференциального уравнения 1
gamma=U3*V1;
U5=SIGMA;
ВЕТТА=U5*fi;
%Постоянная времени в дифференциальном ...
```

```

... уравнении 2
Tsig=T0*(~SIGMA)+T1*SIGMA;
%Входная величина для решения ...
... дифференциального уравнения 2
vv=U2*V2;
%Аргументы логического устройства
ResetTemp=0;
ResetTemp(1)=sign(sign(ksi+eps/2)+1);
ResetTemp(2)=sign(sign((eps/2)-ksi)+1);
%Решение системы булевых формул с помощью МЛМ...
... L-структуры
ResetTemp=Lstruck(~ResetTemp(2),ResetTemp(2), ...
... ~ResetTemp(1),ResetTemp(1),0,0,1);
%Формирование выходной величины принимающее ...
... значение {1,0,-1}
ro=(ResetTemp(1)+(ResetTemp(3)*(-1)));
%Выход регулятора
sys(1)=ro;
sys(2)=gamma;
sys(3)=vv;
sys(4)=Tsig;
% Окончание функции вывода

%Модель L-структуры
%% [F1,F2,F3]=Lstruck(x1,x2,x3,x4,z1,z2,z3)
function rez=Lstruck(y1,y2,y3,x,z1,z2,z3)
% Система булевых формул МЛМ L-структуры
F1=y1&(~z1&z2|z3)|(y1|z1|z2)&x&(~z3);
F2=y1&z1&z2&(~z3)|y2&z1&(~z2)&z3|(y1|(~z1)|z2)&y
2&(~z3)|(y2|z1|z2)...
...&x&z3;
F3=y3&(~z3)|(y2|(~z1)|(~z2))&y3&z3|y2&(~z1)&z2&z
3;
%Вывод результата
rez=[F1; F2; F3]; return;

```

СПИСОК ОСНОВНЫХ СОКРАЩЕНИЙ

- АИПС – автоматизированная информационно-поисковая система
АСР – автоматическая система регулирования
БФ – булева формула
БФу – булева функция
ДКВ – динамический критерий выдачи
ДНФ – дизъюнктивная нормальная форма
ИП-автомат – информационно-поисковый автомат
ИПС – информационно-поисковая система
ИПЯ – информационно-поисковый язык
КНФ – конъюнктивная нормальная форма
М-автомат – многофункциональный автомат
МЛМ – многофункциональный логический модуль
НЛР – нечеткий логический регулятор
ОС – однородная среда
ОУ – объект управления
ПИ-регулятор – пропорционально-интегральный регулятор
ПИД-регулятор – пропорционально-интегро-дифференциальный регулятор
ПОД – поисковый образ документа
ПОЗ – поисковый образ запроса
РАФЧХ – расширенная амплитудно-фазочастотная характеристика
СДНФ – совершенная дизъюнктивная нормальная форма
СПС – система с перестраиваемой структурой
ТИ – таблица истинности
УЛМ – универсальный логический модуль
УУ – устройство управления
XML (eXtensible Markup Language) – расширяемый язык разметки

**Шидловский
Станислав**

Викторович

Родился в 1981 г. в Томске. В 1997 г. поступил, а в 2002 г. окончил Томский политехнический университет по специальности «Автоматизация технологических процессов и производств».

В 2004 г. досрочно закончил аспирантуру с защитой кандидатской диссертации по специальности «Автоматизация и управление технологическими процессами и производствами».

С 2005 г. – доцент кафедры информационно-измерительной техники Томского государственного университета систем управления и радиоэлектроники. Обеспечивает учебный процесс кафедры по двум дисциплинам: «Теория автоматического управления» и «Автоматизация технологических процессов и производств».

Автор более 40 научных трудов, в том числе ряда патентов РФ. Лауреат Всероссийского конкурса «Ползуновские гранты» (2005 г.), победитель федеральной целевой научнотехнической программы Роснауки (2006 г.).



**Shidlovskiy
Stanislav**

Viktorovich

Was born in 1981 in Tomsk. In 1997 passed the entering exams successfully and in 2002 graduated from Tomsk polytechnical university on speciality «Automation of technological process and productions».

In 2004 he finished a postgraduate studying ahead of schedule with the defense of the candidate's thesis on speciality «Automation and control of technological process and productions».

Since 2005 – the associate professor of the faculty of information-measuring technical devices of Tomsk state university of control systems and radio electronics. He provides the educational process at the chair on two disciplines: «the Theory of automated control» and «Automation of technological and production process».

The author of more than 40 scientific works, including some patents of the Russian Federation. The laureate of the All-Russian competition «Polzunov's grants» (2005), the winner of the federal purpose-oriented scientific and technical program of Russian science (2006).

***Готовится к изданию
во втором квартале 2007 г.***

Шидловский С.В. Автоматическое управление. Перестраиваемые структуры в системах с распределенными параметрами. – Томск: Томский государственный университет, 2007.

Монография посвящена комплексному исследованию перестраиваемых структур в системах управления с распределенными параметрами и является продолжением монографии автора «Автоматическое управление. Перестраиваемые структуры».

В книге освещены вопросы логического управления, изложены методы и алгоритмы автоматизированного проектирования, моделирования и обработки информации систем с распределенными параметрами.

Наряду с классическими методами анализа и синтеза систем управления с распределенными параметрами предложен автоматико-структурный метод, который реализуется как совокупность аппаратно-программных фрагментов вычисления процедур, составляющих алгоритмы управления из заданных классов. Метод позволяет строить высокоэффективные модульно наращиваемые структуры с массовым параллелизмом, обеспечивающие требуемую производительность, технически и экономически оправданные, для конкретного ряда приложений.

Книга предназначена для специалистов в области управления системами с распределенными параметрами.

Научное издание

Станислав Викторович Шидловский

**АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ.
ПЕРЕСТРАИВАЕМЫЕ СТРУКТУРЫ**

Редактор *К.Г. Шилько*
Корректор *Н.А. Афанасьева*
Оригинал-макет *С.В. Шидловского*
Дизайн обложки *В.Г. Караваева*

Подписано в печать 17.08.2006 г.
Формат 60×84¹/₁₆. Бумага мелованная.
Печать офсетная. Гарнитура «Таймс».
Усл. печ. л. 16,74. Уч.-изд. л. 18,75.
Тираж 500 экз. Заказ

Томский государственный университет
634050, г. Томск, пр. Ленина, 36
Участок оперативной ризографии и офсетной печати
Редакционно-издательского отдела ТГУ

ISBN 5-94621-186-2



9 785946 211864