



УДК 004.942

© 2021 г. А.Н. Рыбалев, канд. техн. наук

(Амурский государственный университет, Благовещенск)

РЕАЛИЗАЦИЯ АЛГОРИТМА АДАПТИВНОГО УПРАВЛЕНИЯ С ЭТАЛОННОЙ МОДЕЛЬЮ ДЛЯ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ

Предложены подходы к программной реализации системы адаптивного управления с эталонной моделью и параметрической настройкой регулятора на программируемых логических контроллерах (ПЛК) с апробацией синтезируемых систем на комбинированных моделях, включающих имитационную модель объекта регулирования и виртуальный или реальный контроллер.

Ключевые слова: уравнения в пространстве состояний, адаптивное управление, эталонная модель, имитационная модель, ПЛК, управляющая программа.

DOI: 10.22250/isu.2021.70.30-38

Введение

Системы автоматического регулирования технологических параметров, применяемые в промышленности, по большей части построены по классическому принципу замкнутого контура с типовым (П, ПИ, ПИД) регулятором. Внедрению более «продвинутых» подходов мешают, в частности, трудности с разработкой программного обеспечения для управляющих машин – ПЛК, которое имеет свою специфику. В [1 – 3] рассмотрены вопросы реализации некоторых нетиповых алгоритмов управления в программах для ПЛК: регуляторов с произвольно заданными передаточными функциями, транспортного запаздывания, компенсаторов возмущений. Апробация программ проводилась в комбинированных системах на базе виртуального контроллера 3S CoDeSys SP PLCWinNT и Simulink-моделей. Технологические аспекты разработки таких систем с многочисленными примерами изложены в [4].

В данной работе описан опыт программной реализации алгоритма адаптивного управления с эталонной моделью и параметрической настройкой регулятора. В отличие от систем, описанных в [1 – 3], программное обеспечение было апробировано не только на виртуальном, но и на реальном ПЛК.

Система адаптивного управления

На рис. 1 показана структура адаптивной системы управления с эталонной моделью и параметрической настройкой регулятора [5, 6].

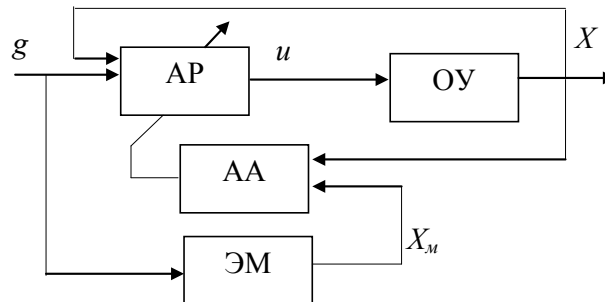


Рис. 1. Структура адаптивной системы регулирования.

Объект управления ОУ описывается в пространстве состояний векторно-матричным уравнением:

$$\dot{X} = AX + Bu,$$

где X – вектор переменных состояния (столбец из n элементов); u – управляющее воздействие (скаляр); A и B – соответственно нестационарная матрица состояния ($n \times n$) и вектор-столбец входа ($n \times 1$).

Все переменные состояния измеряются, т.е.

$$Y = X,$$

где Y – вектор выходных (измеряемых) величин. В дальнейшем эта переменная не используется.

Эталонная модель ЭМ описывается аналогичным уравнением:

$$\dot{X}_m = A_m X_m + B_m g,$$

где g – задающее воздействие.

Управление формируется *адаптивным регулятором* (АР) по закону:

$$u = K_a X + K_b g,$$

где K_a и K_b – настройки адаптивного регулятора соответственно вектор-строка и скаляр, которые вычисляются *алгоритмами адаптации* (АА) согласно системе уравнений:

$$\begin{cases} \dot{K}_a = B_m^T P e X^T R_a; \\ \dot{K}_b = B_m^T P e g R_b, \end{cases}$$

где $e = X_m - X$ – вектор-столбец ошибки адаптации; P – матрица, найденная из решения уравнения Ляпунова:

$$PA_m + A_m^T P = -Q,$$

где Q – симметрическая положительно определенная матрица; R_a, R_b – соответственно диагональная матрица и скаляр с положительными коэффициентами настройки, значения которых выбираются на этапе имитационного моделирования.

В качестве эталонной была выбрана модель, где матрицы состояния и входа имеют вид:

$$A_m = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\omega_0^3 & -2\omega_0^2 & -2\omega_0 \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 \\ 0 \\ \omega_0^3 \end{bmatrix}.$$

Передаточная функция эталонной модели

$$W_m(s) = \frac{\omega_0^3}{s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3}$$

имеет распределение полюсов по Баттерворту – вдоль полуокружности в левой части комплексной плоскости с радиусом, равным ω_0 (параметру, определяющему быстродействие системы). В нашем случае принято $\omega_0 = 0,1$.

Объект управления подвергнут параметрическим рассогласованиям относительно эталонной модели:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0,7\omega_0^3 & -2 \times 1,3 \times \omega_0^2 & -2 \times 0,5 \times \omega_0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1,4\omega_0^3 \end{bmatrix}.$$

Матрицы Q, R_a и скаляр R_b заданы следующим образом:

$$Q = R_a = \begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad R_b = 1.$$

В приложении 1 приведен текст script-файла Matlab, в котором вводятся описанные выше переменные, а также производится расчет матриц P и $B_m^T P$, необходимых для реализации алгоритмов.

Имитационная модель в Simulink

Для моделирования объекта и «образцовой» системы адаптивного управления была разработана Simulink-модель, показанная на рис. 2.

Модель содержит две идентичные подсистемы State Space1 и State Space2, моделирующих объект управления. Первая подсистема управляется

сигналом контроллера, полученным от блока OPC Read, а вторая – сигналом «образцовой» системы регулирования.

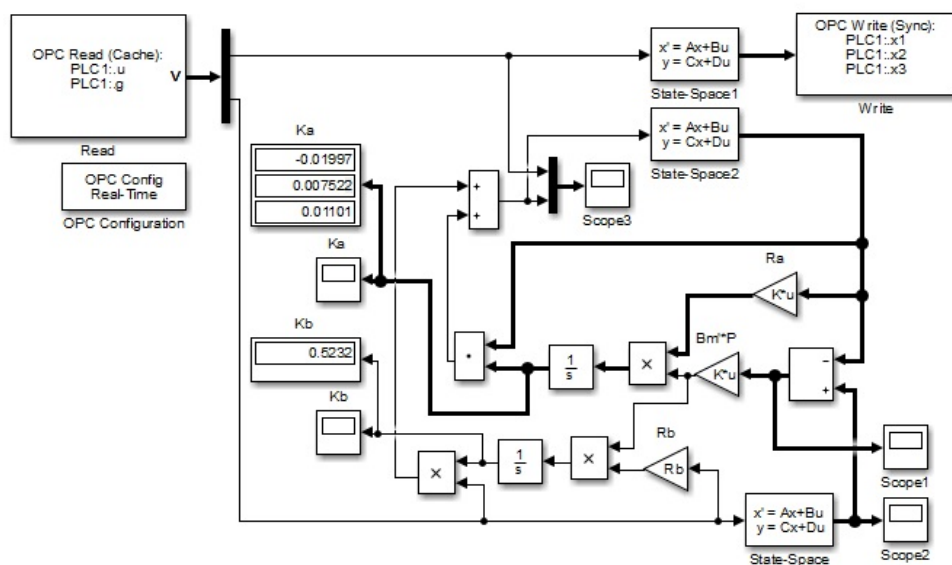


Рис. 2. Simulink-модель для исследования адаптивной системы регулирования.

«Образцовая» система построена непосредственно по уравнениям, приведенным выше.

Программа контроллера

Список глобальных переменных проекта CoDeSys, включающий переменные состояния объекта управления и эталонной модели, задающее и управляющее воздействия, а также сигнал сброса регулятора и эталонной модели, приведен в приложении 2. Часть из этих переменных участвует в обмене с Simulink-моделью (рис. 2).

Программа контроллера состоит из двух задач, к которым привязаны две программы: Model (ЭМ) и AR (АА и АР). Обе выполняются периодически, с интервалом 50 мс.

Для наглядности программы составлены на языке CFC (рис. 3, 4).

```
PROGRAM Model1
VAR
    Integ1, Integ2, Integ3: INTEGRAL;
    Am1:REAL:=0.001; Am2:REAL:=0.02; Am3:REAL:=0.2; Bm3:REAL:=0.001;
END_VAR
```

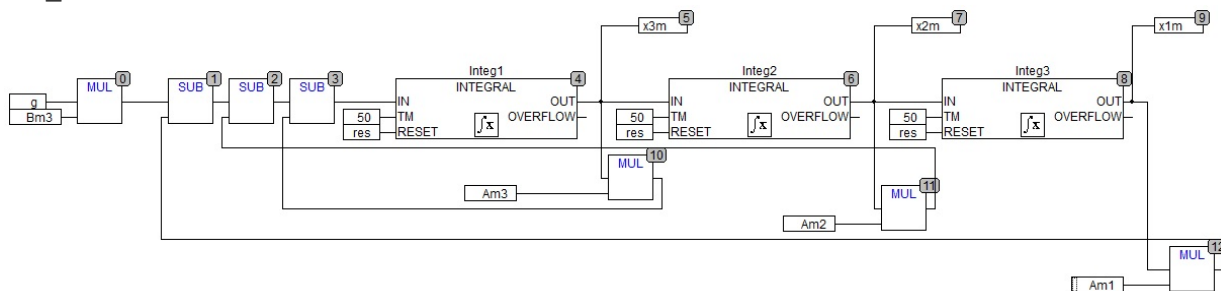


Рис. 3. Программа «Эталонная модель».

```

PROGRAM AR
VAR
Integ1,Integ2,Integ3,Integ4: INTEGRAL;
Ra1: REAL:=0.1; Ra2: REAL:=1; Ra3: REAL:=10; Rb: REAL:=1;
BmP1: REAL:=0.05; BmP2: REAL:=0.7017;BmP3: REAL:=3.5333;
Ka1,Ka2,Ka3,Kb,BmPe: REAL; (*только для программы на ST*)
END_VAR

```

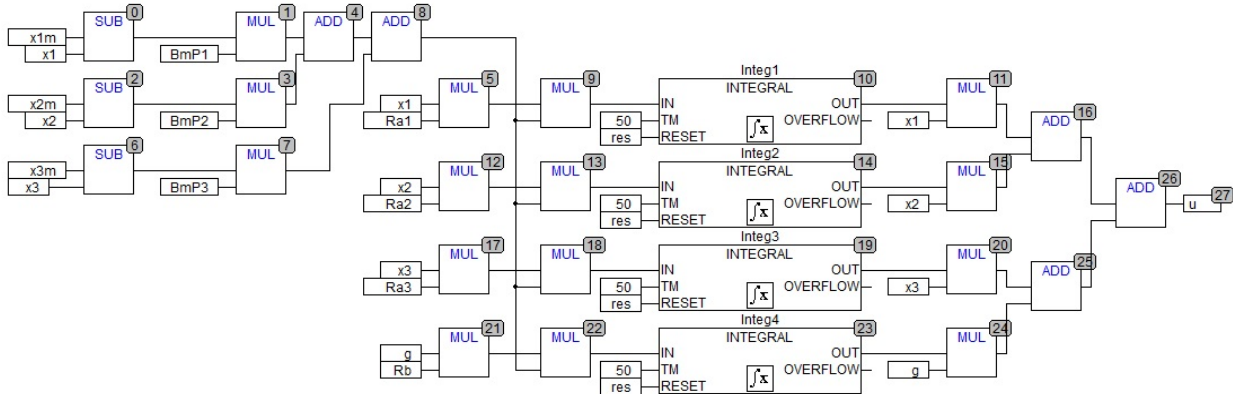


Рис. 4. Программа «Адаптивный регулятор».

В приложениях 3, 4 приведены текстовые версии этих программ на языке ST. Проверка показала, что их поведение полностью идентично поведению графических версий.

Результаты моделирования

Результаты моделирования наблюдались как в блоках Scope Simulink-модели (рис. 2), так и на экране визуализации проекта CoDeSys (рис. 5).

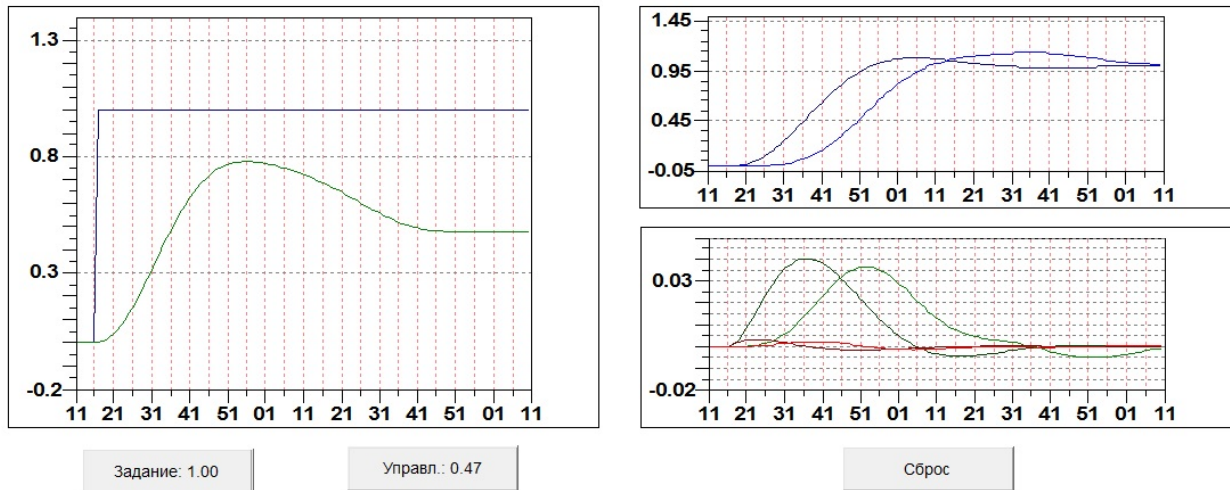


Рис. 5. Экран визуализации.

На левом тренде экрана в реальном масштабе времени отражаются изменения сигналов задания и управления, на трендах справа – координат эталонной модели и объекта управления (сверху – первой координаты, снизу – второй и третьей). С помощью кнопок возможно вводить задания, наблюдать численное значение управляющей величины, сбрасывать регулятор и эта-

лонную модель.

На рис. 6 показан график изменения рассогласований между эталонной моделью и объектом в системе, полностью реализованной в Simulink.

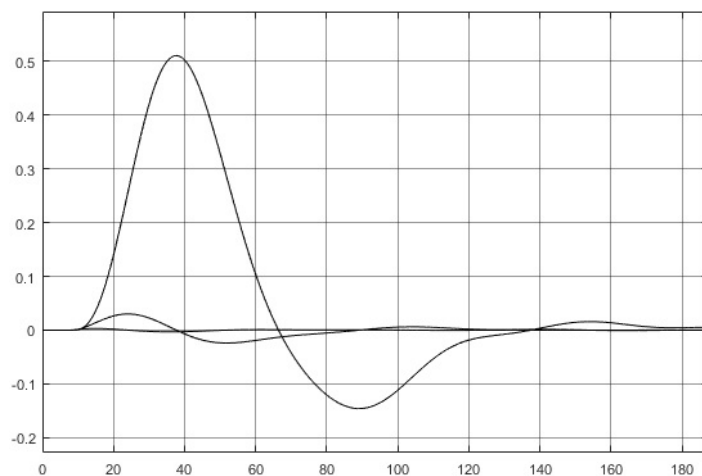


Рис. 6. Рассогласования между координатами эталонной модели и объекта управления (рис. 2, Score 1).

На рис. 7 сравниваются сигналы управления, формируемые виртуальным контроллером и алгоритмом Simulink-модели и подаваемые ими на «свои» объекты.

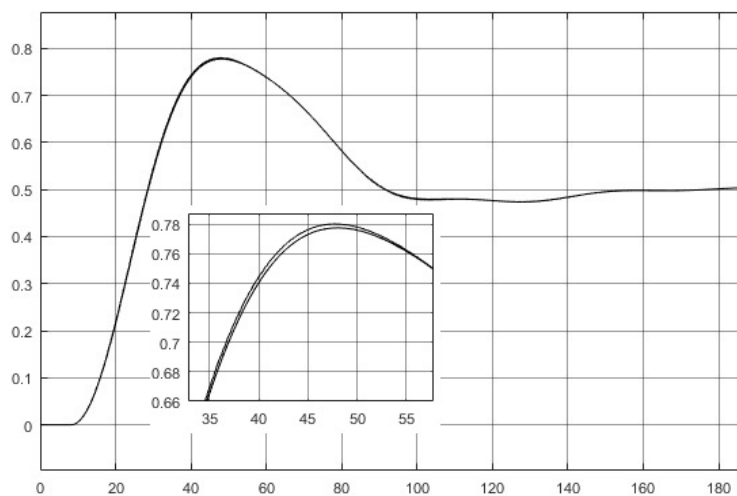


Рис. 7. Управляющие воздействия (рис. 2, Score 3).

Результаты моделирования однозначно отражают как работоспособность систем, так и практически идентичное их поведение, что свидетельствует о корректной программной реализации алгоритма адаптивного управления для ПЛК.

Перенос системы управления на реальный ПЛК

Разработанное программное обеспечение было загружено в программируемый логический контроллер Овен ПЛК 150 ИМ, входящий в состав лабораторного стенда [7].

Предварительно в настройках целевой платформы выбрана конфигурация, соответствующая реальному ПЛК. Полноценно конфигурация ПЛК при этом не «развертывается», т.е. реальные входы и выходы контроллера остаются недоступными. Однако в этом нет необходимости, поскольку программа ПЛК будет по-прежнему взаимодействовать лишь с виртуальным объектом на Simulink-диаграмме. Если потребуется полностью «развернуть» конфигурацию, это можно будет сделать, создав новый проект специально для реального ПЛК и импортировав в него программные единицы из проекта для виртуального.

В настройках соединения был выбран интерфейс связи, который традиционно используется для программирования лабораторного прибора: Ethernet, протокол TCP/IP с фиксированным адресом контроллера. Для компьютера также был установлен фиксированный адрес, причем такой, чтобы он находился с контроллером в одной подсети.

После установки соединения и успешной загрузки проекта в ПЛК был переконфигурирован OPC-сервер CoDeSys, для чего в его настройках был выбран тот же самый интерфейс связи, что задействован и в самой программе CoDeSys. Далее действующий процесс OPC-сервера был уничтожен в Диспетчере Задач Windows для того, чтобы Simulink-модель, будучи запущенной на исполнение, создала новый экземпляр с измененными настройками. Сама Simulink-модель никаких изменений не требует, так как взаимодействует исключительно с OPC-сервером.

Результаты первого запуска полученной системы оказались неудовлетворительными: ее поведение значительно отличалось от поведения системы на базе виртуального контроллера, причем в худшую сторону (наблюдалась сильная колебательность переходных процессов и, соответственно, их затягивание).

Исследования показали, что причиной несоответствия является то, что реальный ПЛК пересчитывает циклически выполняемые программы с частотой, отличной от частоты пересчета в виртуальном контроллере.

Еще в начале исследований, посвященных комбинированным моделям, экспериментально было выявлено следующее обстоятельство. Для корректной работы интеграторов в виртуальном контроллере необходимо установить период вызова соответствующей программы в два раза меньший, чем значение настроечного входа ТМ библиотечного функционального блока INTEGRAL (вход ТМ «информирует» блок о периоде его пересчета). Так, в нашем проекте для виртуального контроллера настройки интервалов вызова были установлены в 25 мс, хотя фактически программы вызывались с пери-

одом 50 мс.

Причины этого обстоятельства неизвестны, возможно, просто ошибка в самой программе виртуального контроллера.

Как выяснилось, реальный контроллер такой ошибки не совершает. После установки периода вызова подпрограмм, равным 50 мс в полном соответствии с настройками интеграторов, проект вновь был загружен в ПЛК.

Результаты запуска системы полностью соответствовали ожиданиям: система на базе реального контроллера продемонстрировала поведение, совпадающее с поведением системы, рассмотренной выше. Поскольку полученные графики практически не отличимы от уже показанных на рисунках, здесь они не приводятся.

Заключение

Рассмотрены вопросы реализации алгоритма адаптивного управления с эталонной моделью и параметрической настройкой регулятора в программах для ПЛК. Программы составлены на графическом (CFC) и текстовом (ST) языках. Апробация проводилась в комбинированной системе, включающей виртуальный контроллер и Simulink-модель объекта. В качестве образца использовалась модель системы управления, реализованная полностью в Simulink. Результаты моделирования продемонстрировали практически полную идентичность поведения систем, что свидетельствует о корректности воспроизведения алгоритмов в ПЛК.

Разработанное программное обеспечение было протестировано в системе с реальным ПЛК. Описан процесс загрузки проекта в контроллер и его конфигурации. Результаты тестирования также показали эффективность реализации алгоритмов.

Приложение 1

Код программы Matlab для расчета матриц для реализации алгоритма адаптивного управления

```
w0 = .1;
Am = [0 1 0; 0 0 1; -w0^3 -2*(w0^2) -2*w0];
Bm = [0; 0; w0^3];

A = [0 1 0; 0 0 1; -w0^3*.7 -2*(w0^2)*1.3 -2*w0*.5];
B = [0; 0; w0^3*1.4];

Q = diag([.1 1 10]);
P = lyap(Am', Q);
BmP=Bm'*P;
Ra=diag([.1 1 10]);
Rb=1;
```


Глобальные переменные проекта CoDeSys

```
VAR_GLOBAL
  x1, x2, x3: REAL;
  x1m, x2m, x3m: REAL;
  g: REAL;
  u: REAL;
  res: BOOL;
END_VAR
```

Приложение 3

Код программы расчета координат эталонной модели на языке ST

```
Integ1(IN:=Bm3*g - Am1*x1m - Am2*x2m - Am3*x3m, TM:=50, RESET:=res,
  OUT=>x3m);
Integ2(IN:=x3m, TM:=50, RESET:=res, OUT=>x2m);
Integ3(IN:=x2m, TM:=50, RESET:=res, OUT=>x1m);
```

Приложение 4

Код программы расчета управляющего воздействия на языке ST

```
BmPe:= BmP1*(x1m - x1) + BmP2*(x2m - x2) + BmP3*(x3m - x3) ;
Integ1(IN:=x1*Ra1*BmPe, TM:=50, RESET:=res, OUT=>Ka1);
Integ2(IN:=x2*Ra2*BmPe, TM:=50, RESET:=res, OUT=>Ka2);
Integ3(IN:=x3*Ra3*BmPe, TM:=50, RESET:=res, OUT=>Ka3);
Integ4(IN:=g*Rb*BmPe, TM:=50, RESET:=res, OUT=>Kb);
u:=Ka1*x1 + Ka2*x2 + Ka3*x3 + Kb*g;
```

ЛИТЕРАТУРА

1. Рыбалев А.Н. Компьютерное моделирование нетиповых законов регулирования для программируемых логических контроллеров // Информатика и системы управления. – 2016. – №4(50). – С.33-43.
2. Рыбалев А.Н. Реализация и компьютерное моделирование алгоритмов регулирования с транспортным запаздыванием для программируемых логических контроллеров // Информатика и системы управления. – 2017. – №2(52). – С.12-24.
3. Рыбалев А.Н. Реализация и компьютерное моделирование алгоритмов компенсации возмущения для программируемых логических контроллеров // Информатика и системы управления. – 2020. – №4(66). – С. 37–48.
4. Рыбалев А.Н. Имитационное моделирование АСУ ТП. – Благовещенск: Амурский гос. ун-т, 2019.
5. Борцов Ю.А., Поляхов Н.Д., Путов В.В. Электромеханические системы с адаптивным и модальным управлением. – Л.: Энергоатомиздат, 1984.
6. Борцов Ю.А., Соколовский Г.Г. Автоматизированный электропривод с упругими связями. – Изд. 2-е, перераб. и доп. – СПб.: Энергоатомиздат, 1992.
7. Рыбалев А.Н. Программируемые логические контроллеры и аппаратура управления: лабораторный практикум. Часть 3. Овен ПЛК 150 и модули МВА8 и МВУ8. Учебное пособие. – Благовещенск: Амурский гос. ун-т, 2010.

Статья представлена к публикации членом редколлегии Е.А. Ереминым.

E-mail:

Рыбалев Андрей Николаевич – atgu_appre@mail.ru.