

Наведено теоретико-множинне обґрунтування системи блочного проектування САК, описано досвід практичної реалізації програмного комплексу, заснованого на принципах вільно розповсюдженого первинного коду, призначеного для проектування та дослідження САК складними системами

Ключові слова: теоретико-множинне обґрунтування, блочне моделювання, системи управління

Приведено теоретико-множественное обоснование системы блочного проектирования САУ, описан опыт практической реализации программного комплекса, основанного на принципах свободно распространяемого исходного кода, предназначенного для проектирования и исследования САУ сложными системами

Ключевые слова: теоретико-множественное обоснование, блочное моделирование, системы управления

There is a set-theoretic justification of ACS unit design, described the experience of practical implementation of software system based on the principles of free source code for the design and the ACS study complex systems

Key words: set-theoretic reasoning, block modeling, control system

СРЕДА РАЗРАБОТКИ И ИССЛЕДОВАНИЯ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ASPC-LAB

В. К. Тытюк

Кандидат технических наук, доцент*
Контактный тел.: (0564) 92-16-92
E-mail: dinalt2006@gmail.com

И. А. Луценко

Доктор технических наук, доцент, заведующий кафедрой*

*Кафедра электроснабжения и ресурсосбережения
Криворожский технический университет
Контактный тел.: (0564) 28-10-97
E-mail: sitsr@e-mail.ua

1. Введение

Предварительный анализ систем автоматического управления технологическими процессами методами математического моделирования является важным этапом разработки таких систем, обеспечения высокого уровня их функциональности, повышения качества их работы. В связи с ростом конструктивной и математической сложности систем управления для их математического моделирования получил широкое распространение объектно-ориентированный подход, при котором САУ представляется в виде набора блоков, соединенных между собой необходимым образом, выполняющих функции отдельных технических элементов.

2. Современное состояние вопроса

В настоящее время разработано и применяется несколько популярных программ, предназначенных для блочного моделирования динамических систем, в том числе и систем управления технологическими

процессами, такие как MATLAB, Mathcad, LabView, AnyLogic и другие. Все перечисленные программы характеризуются рядом особенностей: закрытая архитектура исполнительной части программы; отсутствие в стандартном наборе поставки требуемых для решения задач анализа процессов управления технологическими процессами готовых специализированных компонентов; высокая стоимость лицензий на такие программы.

Закрытая архитектура подсистемы выполнения разработанной модели снижает уровень контроля разработчика над корректностью полученных в ходе математического моделирования результатов, снижает уровень доверия к ним. В [1] прямо указывается, что использование популярных пакетов прикладных программ — MATLAB, Mathcad, Scilab и многих других — может приводить к ошибочным результатам расчета и становится причиной аварий и катастроф, приводя к рекомендациям по поиску и устранению такого рода ошибок.

Любопытно отметить, что именно в задачах синтеза систем управления, доставляющих минимум квадратичным критериям качества, впервые на практике

столкнулись с явлением изменения корректности задачи при эквивалентных преобразованиях — причем столкнулись, не понимая еще сущности явления.

Перечисленные выше программы обладают развитыми возможностями создания пользовательских блоков за счет применения встроенных языков программирования. Это приводит к необходимости постоянного совершенствования этих языков программирования и требует непрерывного контроля работоспособности ранее разработанных моделей в новых версиях программ. Кроме того, встроенные языки программирования являются, как правило, менее гибкими и менее функциональными, имеют более ограниченную техническую поддержку по сравнению с общепризнанными языками программирования C и Pascal, являющимися фактически стандартом de facto для разработчиков программного обеспечения.

3. Цель исследований

Преодоление ограничений известных программ исследования систем автоматического управления, разработка альтернативного подхода к разработке и исследованию процессов управления сложными системами. Актуальным является создание библиотеки свободно-распространяемых макро-блоков систем автоматического управления, обеспечить свободный доступ к системам разработки и анализа САУ, что даст возможность упростить исследование процессов управления и разработку систем автоматического управления сложными системами.

4. Теоретико-множественные аспекты построения блочных моделей

Основой для составления модели является множество классов C – объединение определенных вполне различаемых прототипов механизмов систем автоматического управления, которые обладают некоторым общим набором свойств и различаются значением этих свойств:

$$C_i = (c_1, \dots, c_i)$$

Каждый прототип механизмов САУ обладают множеством параметров $P_m = (p_1, \dots, p_m)$, множеством входных сигналов $I_j = (i_1, \dots, i_m)$ и множеством выходных сигналов $O_k = (o_1, \dots, o_k)$. Значения m, j, k принадлежат множеству натуральных чисел N , а, следовательно, множества P, I, O являются конечными множествами. Отдельный элемент множества прототипов механизмов САУ будем обозначать следующим образом: $c_i(P_m, I_j, O_k)$.

Математическая модель системы автоматического управления состоит из множества некоторых механизмов, соединенных между собой определенным способом, и содержит две основные сущности:

1. Состав модели - множество элементов, входящих в состав САУ совместно с заданными значениями множества параметров каждого элемента;

2. Структура модели - порядок (схема) соединений блоков модели.

Процесс создания модели САУ с математической точки зрения можно описать следующим образом:

Пусть L – некоторое множество, которое мы назовем множеством целочисленных индексов $l, l \in L$.

Для каждого элемента C с помощью специального отображения создается конечное множество V_l , содержащее несколько (N_l) одинаковых элементов C_l : $\forall C \rightarrow V$.

Множество V_l может быть пустым, если элемент типа l не используется в модели. Значение N_l определяет, сколько элементов типа l необходимо для работы модели. Например, как правило, в модели САУ используется несколько сумматоров, источников сигналов и т.п.

Тогда модель системы автоматического управления может быть определена в виде формального математического выражения как обобщенное объединение множества C . Модель САУ является множеством M , определяемым как объединение семейства подмножеств $\{B_k | k \in K\}$, являющееся множеством $\{x | \exists k (k \in K \wedge x \in B_k)\}$. Это множество обозначают так: $\bigcup_{k \in K} B_k$. Размерность модели САУ можно вычислить по формуле $K_c = \sum_{v \in C} N_l$

Когда состав модели определен в виде множества M , можно задать структуру модели. Для этого вначале определим множество всех входных сигналов модели AI и множество всех выходных сигналов модели AO . Эти множества могут быть вычислены как объединения множеств входных и выходных сигналов соответственно для всех блоков модели: $AI = \bigcup_{v \in M} I$, $AO = \bigcup_{v \in M} O$.

Структура модели устанавливает взаимно однозначное соответствие между множествами входных AI и выходных AO сигналов модели, причем к каждому входу может быть непосредственно подключен только один выход какого-либо блока. Таким образом, структура модели S является однозначным отображением (функцией) множества выходных сигналов модели во множество входных сигналов модели: $S = AO \rightarrow AI$. Структуру модели S можно трактовать как отношение между множествами AO и AI и отождествлять его с некоторым подмножеством декартова произведения $AO \times AI$, называемым графиком отображения S и определяемым следующим образом

$$S = \{(x, y) | x \in AO \wedge y \in AI \wedge S(x) = y\}$$

5. Прикладные концепции и практическая реализация

Разрабатываемый программный комплекс ASPC-Lab функционально состоит из двух основных частей: из набора блоков, строительных блоков для создания моделей систем автоматического управления и управляющего ядра, обеспечивающего сборку модели, ее выполнение и визуализацию полученных результатов.

Реализация отдельных блоков - механизмов систем автоматического управления выполнена в виде подключаемых модулей – плагинов, практически реализуемых в виде динамически подключаемых библиотек (DLL). Такой подход позволяет в конечном итоге избавиться от ограничений на используемые языки программирования, позволит вести разработку различных блоков широкому сообществу пользователей, независимо друг от друга и от разработчиков исполнительного ядра программы.

Управляющее ядро программы при этом подходе выполняет функции определения состава модели, соз-

дания таблицы соединений модели, исполнения модели и визуализации полученных результатов. Эта часть программы не содержит в себе реализации конкретных блоков. Управляющая программа разрабатывается с использованием концепции открытого кода с использованием свободно распространяемых программных продуктов. В частности, авторами был использован свободно распространяемый компилятор Lazarus Free Pascal.

Программа содержит список всех доступных и зарегистрированных в программе блоков-прототипов. Для хранения информации о структуре модели, параметрах блоков и таблицы соединений использована свободно распространяемая база данных Firebird. Такое решение является более продуктивным по сравнению с разработкой специализированных файлов для хранения параметров модели. Пользователь имеет возможность создавать свою модель через серию последовательных шагов, сохраняя промежуточные состояния модели, что заметно упрощает разработку и отладку модели. В процессе описания структуры модели пользователь создает список экземпляров классов, которые будут входить в модель. Каждый блок в пределах модели получает уникальное, автоматически формируемое имя. В специальных таблицах для каждого блока, включенного в модель, необходимо задать числовые значения параметров блока и при необходимости – значения начальных условий для динамических блоков.

После создания структуры модели можно переходить к созданию таблицы соединений модели. Сведения о соединениях модели хранятся в отдельной таблице базы данных. К каждому входу блока может быть подключен выход только одного блока. Не допускается существование неподключенных входов. После задания числовых значений параметров и создания таблицы соединений разрабатываемая модель готова к выполнению.

На уровне управляющего ядра программы отдельный блок представлен в виде экземпляра общего уни-

версального класса TModule, реализующего функции подключения к DLL блока и выполнения стандартных функций, определяемых протоколом обмена. Таким образом, все общие сведения о прототипе блока, такие как количество входных и выходных сигналов, условные названия входов и выходов, сведения о параметрах блока определены внутри DLL блока. Наиболее важной функцией DLL является функция Run, реализующая прикладную функциональность блока. Функция Run вычисляет значения выходных сигналов блока на момент времени $t + \Delta t$ по известным входным сигналам блока на момент времени t .

На уровне управляющего ядра программы модель изучаемой системы представляется в виде динамического массива экземпляров класса TModule. При создании каждого экземпляра ему передаются сведения об используемой этим экземпляром DLL, что полностью определяет его видовой функциональности, и присваиваются числовые значения параметров. После создания динамического массива модели запускается главный цикл программы, выполняющий следующие действия: - вычисляются выходные сигналы блоков-источников сигналов (безвыходных блоков); вычисляются сигналы на входах прочих блоков в соответствии с созданной на этапе конструирования модели таблицей соединений; методом Run вычисляются выходные сигналы блоков; выходные сигналы выводятся на графики и в протокол работы модели. Протокол работы модели позволяет сохранять результаты работы в файлы, совместимые с Microsoft Excel, что повышает удобство при обмене данными и расширяет возможности визуализации полученных результатов.

На рис. 1. представлено главное окно программы ASPC-Lab. На рисунке изображены результаты разработки модели системы автоматического управления процессом нагрева.

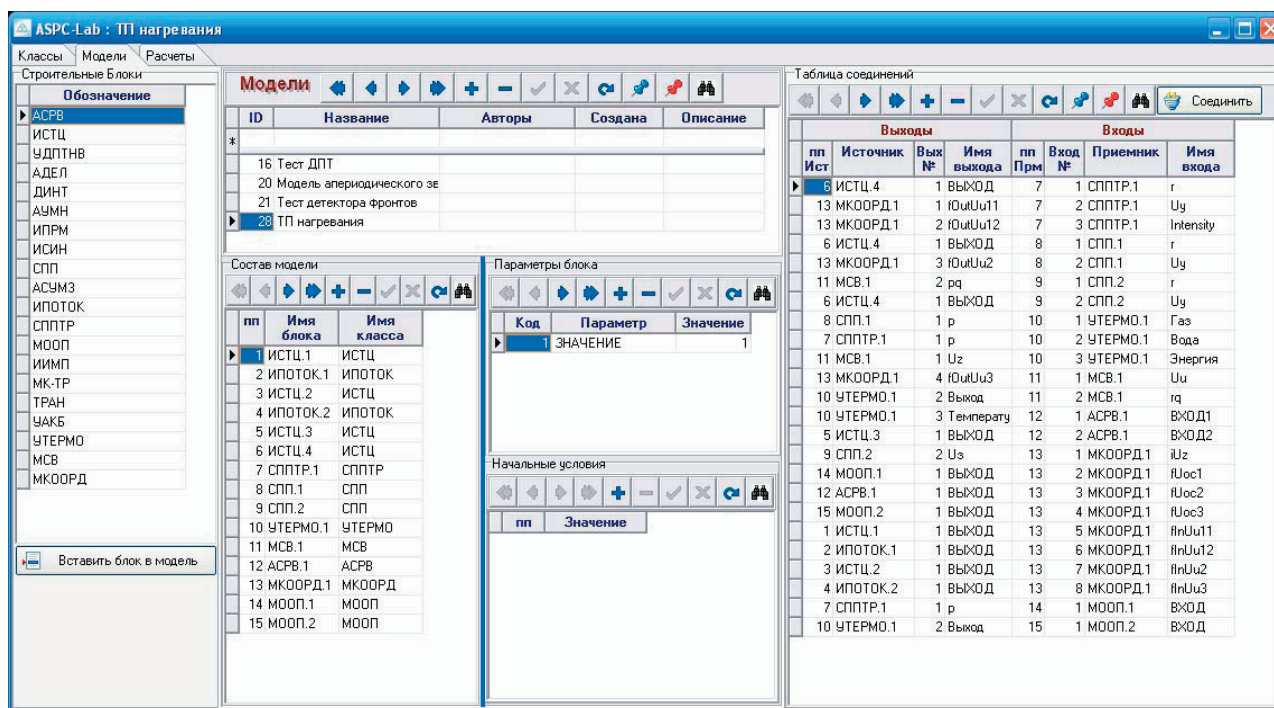


Рис. 1. Главное окно программного комплекса ASPC-Lab

Разработана стартовая версия исполнительного ядра ASPC-Lab, предложены первые версии технической документации для разработчиков новых блоков, разработан стартовый набор блоков, включающий в себя типовые блоки источников сигналов, арифметико-логические блоки, элементарные динамические звенья. Общая работоспособность предложенной концепции проверялась на простейших моделях.

Для расширенной проверки работоспособности предложенной программы была разработана математическая модель системы управления технологическим процессом нагрева.

Технологический процесс нагрева включает в себя механизмы подачи нагреваемой заготовки, механизм подачи термоагента (электроэнергии, газа), механизм выгрузки нагретой заготовки, механизм нагрева, механизм координации, синхронизирующий последовательное выполнение различных стадий процесса.

На рис. 2. представлена разработанная схема полной модели САУ процессом нагрева.

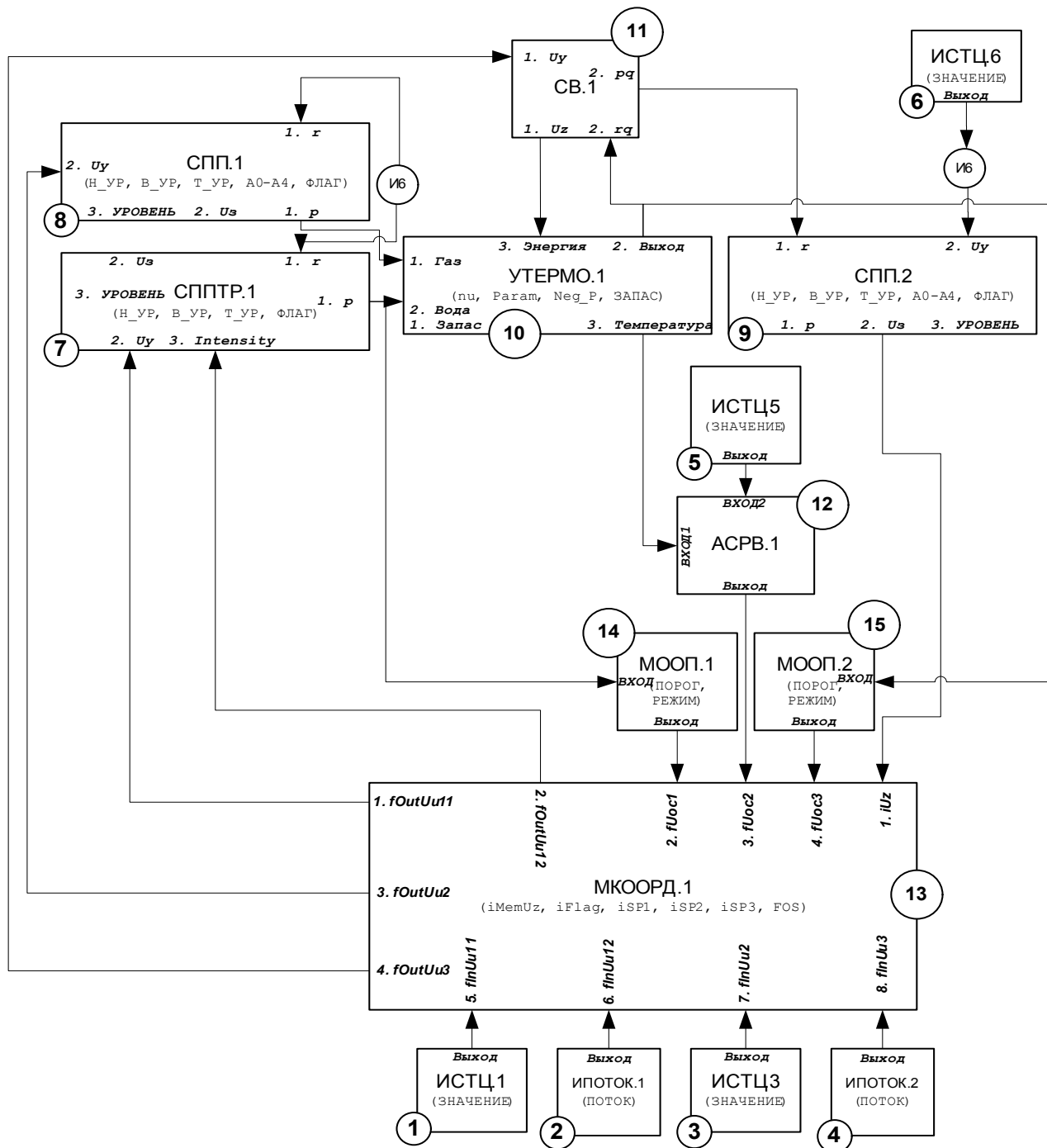
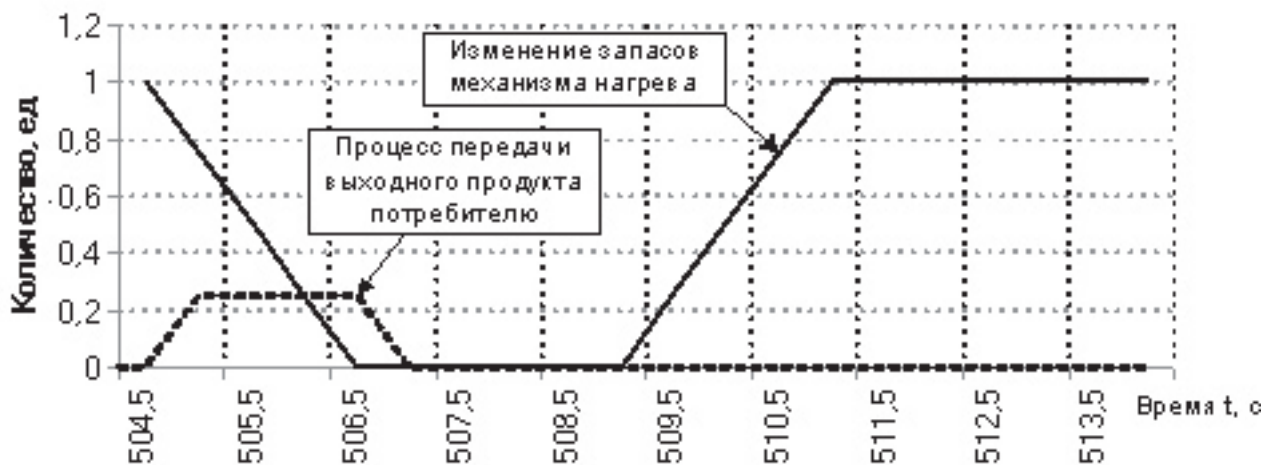
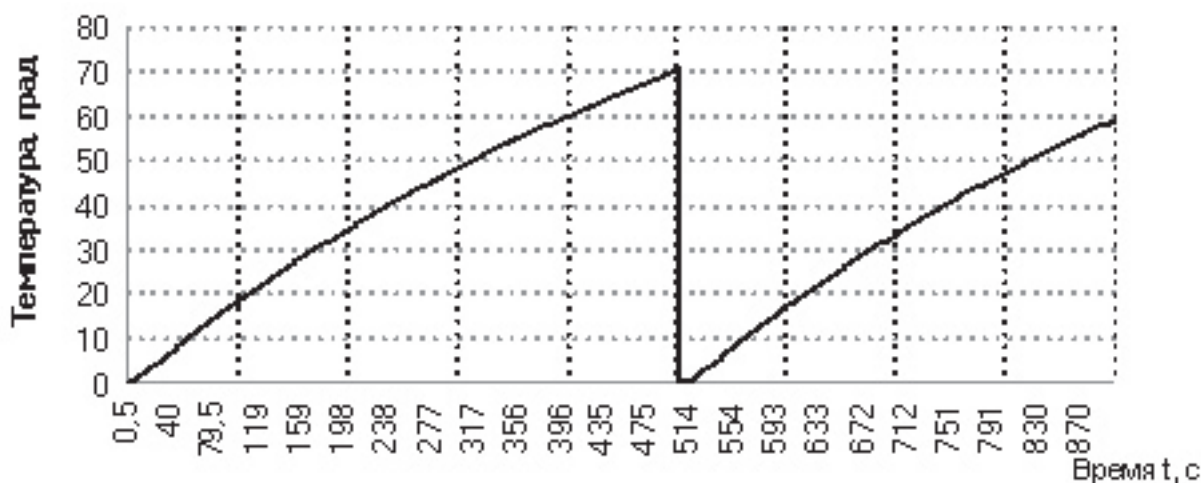


Рис. 2. Структурная схема САУ системы управления нагревом с партионной подачей входных продуктов в системе ASPC-Lab, разработанная в MS Visio

На рис. 3. изображены временные диаграммы изменения учетных (а) и качественного (б) параметров технологического процесса нагрева.



а)



б)

Рис . 3. Временные диаграммы процессов изменения учетных параметров а) и качественного параметра б) технологического процесса нагрева с партионной подачей входных продуктов

6. Выводы

Предложена теоретико-множественная концепция построения блочных систем моделирования, на основе которой разработана блочная система проектирования и исследования систем автоматического управления сложными процессами, работоспособность разработанного программного комплекса проверена на системе автоматического управления технологическим процессом нагрева с партионной подачей входных продуктов.

Литература

1. Петров Ю. П., Петров Л. Ю. Неожиданное в математике и его связь с авариями и катастрофами. — 4-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2005. — 240 с: ил. ISBN 5-94157-543-2.
2. Н. Бурбаки. Теория множеств. Пер. с французского под ред. В. А. Успенского. Изд. «МИР», Москва, 1965, 455с.