

УДК 004.624

А.В. Деревянко

*Харьковский национальный университет имени В.Н. Каразина, Харьков, Украина*

## КОНЦЕПЦИЯ ХРАНИЛИЩ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ НАНОТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

*В работе описывается метод построения модуля сбора и хранения данных системы управления нанотехнологическими процессами, построенного на основе программных модулей, способных получать данные из разнородных источников и приводить к унифицированному формату, что ускоряет их дальнейшую обработку. Использование системы позволяет одинаково эффективно работать как с текущими экспериментальными данными, так и с историческими. Описана архитектура построенной системы, протоколы работы её составных частей, а также методы дальнейшей обработки экспериментальных данных.*

**Ключевые слова:** система автоматизированного управления, сбор данных, хранилище данных, ионно-лучевой синтез.

### Постановка задачи

Современные технологические процессы требуют новых подходов к построению систем автоматизированного управления. В первую очередь это связано с нарастающим объёмом данных, поступающих от датчиков параметров таких процессов, зачастую трудных для восприятия оператором, сложностью алгоритмов управления, малыми допустимыми временами для принятия решения на адекватное управляющее воздействие. Интенсивно развивающееся направление нанотехнологий и сопутствующие эксперименты также нуждаются в современных системах автоматизации. Так, например, процесс ионно-лучевого синтеза, представляющий собой один из наиболее перспективных методов получения тонких плёнок [1] и позволяющий получать качественные покрытия заданной стехиометрии и физических свойств, является очень нетривиальным для управления процессом. Получение необходимых наноразмерных плёнок с наперед заданными свойствами требует высокой точности контроля и управления множеством параметров распыляющего пучка, состава и давления газовой среды в технологическом объёме, температурных режимов синтеза и других связанных сложными зависимостями параметров. Для этого строятся системы, в которых с помощью большого количества датчиков собираются данные, по возможности, полно описывая процесс. При накоплении достаточного количества подобных описаний, возможно прогнозирование [2] на основании которого и будет приниматься решение о воздействии на элементы управления для осуществления стабилизации процесса и устранения отклонений от его оптимального проведения. Для осуществления прогнозирования необходимо также иметь возможность в качестве исходных данных использовать не только накопленную эмпирическую информацию, но и сторонние данные, хранимые в раз-

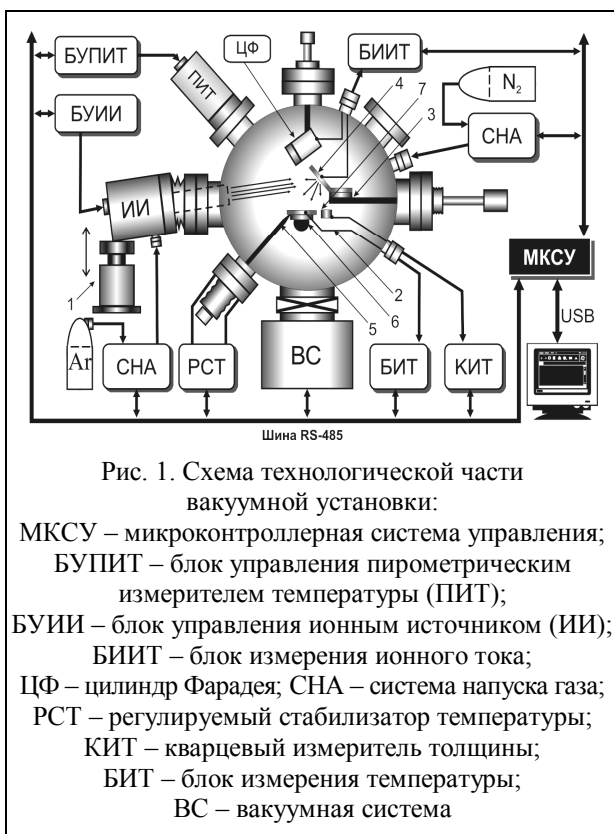
личных форматах, такие, как полученные ранее зависимости, известные табличные данные, математические формулы и т.п. Для того чтобы обрабатывающий их модуль мог получить необходимые результаты, и в то же время был достаточно прост, чтобы работать в режиме реального времени и обеспечивать требуемое быстродействие при анализе, поступающие данные должны быть однородны и не должны требовать дополнительных вычислительных ресурсов. Таким образом, возникает актуальная задача создания системы хранения данных, извлекающей информацию из абсолютно разнородных источников в различной форме и приводящей их к унифицированному виду для последующей оперативной обработки [3].

### Описание системы сбора данных

Для наработки необходимого количества разнородных данных технологическая установка, на которой осуществляется процесс нанесения покрытий, была оснащена набором датчиков и элементов управления, связанных между собой и управляющим центром (рис. 1) [4].

Датчики представляют собой интеллектуальные устройства на базе микроконтроллеров или программируемых логических матриц, имеющие физический интерфейс для связи со средой передачи данных и набор предоставляемых функций, специальный для каждого узла. В качестве физического интерфейса была выбрана промышленная шина RS-485 [5], представляющая собой общую среду передачи данных с последовательным подключением узлов системы.

При выборе логического протокола передачи был сделан обзор наиболее часто используемых решений для подобных систем. Выбор был сделан в пользу протокола Wake [6], работающего одинаково хорошо и с RS-485, и с USB.



Зная специфику работы системы формат фреймов, используемый по умолчанию указанным протоколом был изменён. Для этого было проведено имитационное моделирование работы системы. Варьируя такие параметры, как:

- $T_i$  – период инициации  $i$ -й задачи, т.е. насколько часто задача выставляет признак готовности;
- $T_i^0$  – период опроса  $i$ -й задачи диспетчером;
- $t$  – время передачи одного байта информации;
- $N$  – общее кол-во станций;
- $L$  – максимальный размер пакета,

были получены оптимальные размеры полей кадра [7]. Кроме того, так как передача данных идёт постоянно, то можно воспользоваться аналитическим выражением для подсчёта времени передачи порции данных [8].

Для аналитического расчёта полного времени опроса использовалась формула:

$$t_{\text{Total}} = t \{ 2r_1 + [1/L](r_2 + L) + \delta(\Delta l + r_2) \},$$

где  $\Delta l = 1 - [1/L]L$  – длина последнего неполного пакета при  $l > L$ ;

$$\delta = 0 \text{ при } \Delta l = 0, \delta = 1 \text{ при } \Delta l > 0;$$

$r_1, r_2$  – длина служебных сообщений соответственно формата 1 и 2 ( $r_1 \geq 9, r_2 \geq 10$  байтов);

$$[x] – \text{целая часть числа } x.$$

В качестве значений частоты опроса датчиков выбирались значения, достаточные для получения адекватных моделей. Для этого необходимо вычислить оптимальные числа дискретизации по осям координат  $M$  и  $q$  и оптимальную размерность динамической системы  $n$ .

Для решения этой задачи введём:

1) математическую модель исходных данных в следующем виде:

$$Y_i = (y_0, y_1, \dots, y_i, \dots, y_{m-2}, y_{m-1}) \in \Omega_q^M, \quad (1)$$

где  $y_i \in Q = \{q, q-1\}, 0 \leq i < M-1$ ,

$M$  – целая часть  $T / \Delta t$ ;

2) математическую модель источника системы в виде уравнения нелинейной регрессии порядка  $n$

$$y_{i+1} = f_i(y_i, y_{i+1}, \dots, y_{i+n-1}; p) = f_i, \quad (2)$$

где  $0 \leq i < M-n$  и вектор-параметра

$$p \in \Pi = \{0, q-1\};$$

3) ограничения по уровню и продолжительности исходных данных

$$-\infty < y_{\min} \leq y(t) \leq y_{\max} < \infty, \quad (3)$$

где  $t \in [0, T]$ ;

4) способ приведения всех данных к (1) за счёт их аналого-цифрового преобразования:

$$y_i = \left\lfloor \left\{ \frac{y(i \cdot \Delta t) - y_{\min}[0, T]}{y_{\max}[0, T] - y_{\min}[0, T]} \right\} (q-1) \right\rfloor. \quad (4)$$

Согласно [9] для ряда (1), удовлетворяющего условиям (3), всегда существует математическая модель в виде уравнения (2), с помощью которой по  $n$  начальным выборкам возможно вычисление оставшихся  $M-n$ , при этом при заданном  $T$  указанные параметры  $M, q$  и  $n$  могут быть подобраны достаточно оптимально [10].

Протокол позволяет производить обмен пакетами данных длиной до 255 байт с адресуемыми устройствами, которых может быть до 127. Последовательный канал позволяет работать на скорости до 115200 бод/с. Разработанная система работает при центральном арбитраже, то есть за весь обмен данными отвечает центральный компьютер, таким образом, система работает следующим образом: при необходимости получить значение какого-либо параметра компьютер посылает в сеть запрос, указывая в качестве получателя уникальный номер нужного датчика. Датчик, идентифицировав свой код, посылает результаты измерений обратно. Подобная полудуплексная работа системы стабилизации довольно медлительна, и обладает высокой инерционностью (фактически, чтобы поддерживать давление

на требуемом уровне, центральный компьютер должен постоянно опрашивать датчик давления) и посылать нужные команды узлу напуска, но переход на полнодуплексную связь устройств, при которой система напуска смогла бы самостоятельно опросить датчик давления и принять решение без участия компьютера приводит к появлению коллизий на линии, борьба с которыми, используя даже хорошо отлаженные методы типа CSMA/CD или децентрализованного кодового управления является трудной задачей, что приводит к многочисленным сбоям и задержкам в работе системы [11].

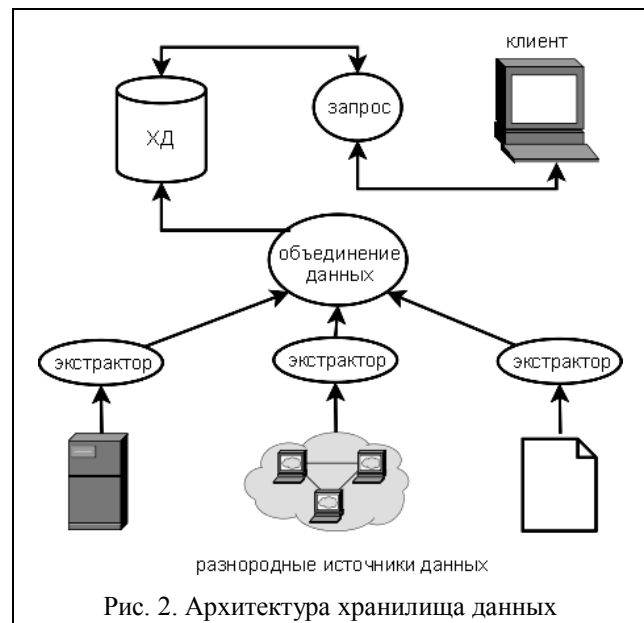
Переданные датчиками данные обрабатываются специально созданным программным обеспечением и сохраняются, как можно более полно описывая процесс.

### Выбор средств хранения информации

Существует множество методов хранения информации, используемых экспериментаторами, и средств их обработки, используемых исходя из предъявляемых требований: начиная от хранения данных в виде текстовых файлов, что быстро в реализации, но неудобно при дальнейшей их обработке, и заканчивая использованием баз данных, со своими СУБД, реализующими часть часто используемых алгоритмов. Множество методов организации хранения данных зачастую приводит к тому, что данные, требуемые для управления процессами находятся в различных источниках с разнообразными средствами работы с ними. Желание же использовать ещё и результаты работы других, созданных ранее, информационно-управляющих систем приводит к необходимости создания некоего промежуточного звена между источником данных и обрабатывающей программой.

Подобный подход используют современные хранилища данных [12]. Хранилищами данных (ХД) называют очень большие предметно-ориентированные информационные базы данных, специально разработанные и предназначенные для подготовки отчётов и анализа процессов. Классические ХД строятся на базе клиент-серверной архитектуры, СУБД и утилит поддержки принятия решений. Данные, поступающие в ХД, становятся доступны только для чтения. Данные из системы работы в реальном режиме времени копируются в ХД таким образом, чтобы анализ не использовал ресурсы системы и не нарушал ее стабильность. Данные загружаются в хранилище с определенной периодичностью, поэтому актуальность данных несколько отстает от системы сбора. Ещё одной важной особенностью ХД является то, что хранимые данные в связи с их большим объёмом зачастую разделяют на подмножества, называемые витринами данных.

Архитектура хранилища может быть схематически представлена рис. 2.



В основе системы лежат различные источники данных. В их роли могут выступать как реляционные БД и документы в стандартных форматах, хранящие архивную информацию, так и системы, работающие с данными в режиме реального времени.

Для того чтобы всю информацию представить в единой форме, для каждого из типов источников данных есть собственный обработчик, извлекающий информацию из своего источника, обрабатывающий её и передающий в центральную часть ХД.

Большинство промышленных ХД используется для работы с корпоративными БД, и основная их цель – принимать участие в анализе бизнес-процессов с целью поддержки принятия решений в организации. При этом стараются сделать систему как можно более обширную, распределённую, нередко с возможностью распараллеливания.

### Реализация системы ХД для поставленной задачи

Было предложено использовать сходную концепцию для управления нанотехнологическими процессами, в частности, технологического процесса ионно-лучевого испарения.

Несмотря на отсутствие в необходимости хранить терабайты данных и передавать их на огромные расстояния, поддерживая целостность общей системы, возможность предоставления пользователю такого, интерфейса, при котором он абстрагирован от реального источника данных, способность обрабатывать запросы, относящиеся одновременно к нескольким разнородным источникам, и помощь в принятии решения касательно оптимальных параметров протекания процесса делают подобный подход оправданным.

Для реализации этой идеи была предложена следующая архитектура: программное обеспечение

работы хранилища было разбито на клиентскую и серверную части, причём сервер обслуживает центральный объединенный массив информации, а все источники являются его клиентами. При поступлении новых данных клиент генерирует сигнал-триггер, обращаясь к хранилищу с предложением добавить в него свои данные. После подтверждения информация пересылается, подхватывается на стороне сервера, проверяется им и складывается. Вся информация, попавшая в центральное хранилище, всегда доступна только для чтения, так как представляет собой исторические данные. Как было сказано выше, эти данные могут быть либо показаниями датчиков и режимов устройств управления, последовательность которых полностью описывает процесс с технологической точки зрения, либо общими данными в графическом, табличном или математическом виде.

Клиентские программы отличаются лишь модулем связи с источником, при этом имеют одну и ту же основу, так как в целом выполняют одну и ту же задачу, а именно: не препятствуя обновлению источника данных, следить за его изменениями и по их обнаружению преобразовывать изменения в общий формат, обрамлять полученный пакет метаданными о времени обновления и типе источника данных.

Подобные экстракторы данных были разработаны для текстовых файлов, таблиц реляционных баз данных в формате Paradox, DB2, dBASE и т.д.

Таким образом, наша реализация концепции ХД может быть представлена следующим образом (рис. 3).

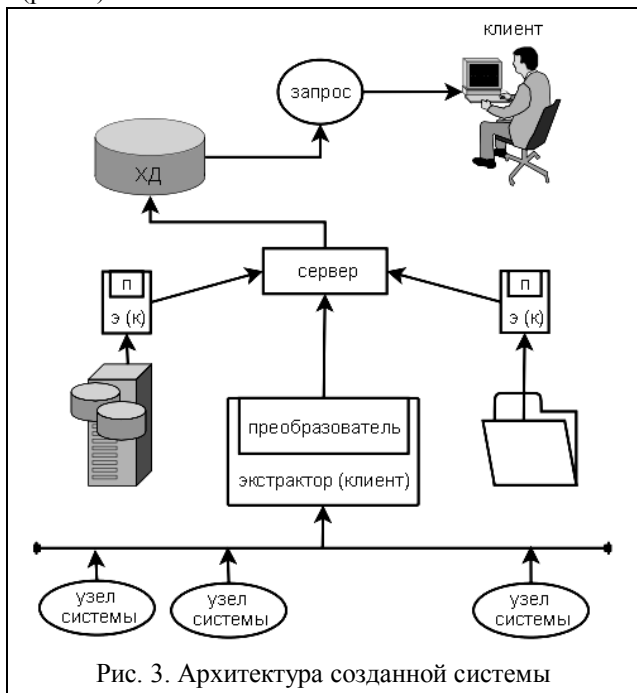


Рис. 3. Архитектура созданной системы

В связи со спецификой задачи преобразователь данных является составной частью экстрактора, что

позволяет при добавлении поддержки нового типа источника не вносить изменений в серверную часть.

Реализация подобной системы привела к возможности использовать удалённые друг от друга источники данных. Действительно, нет существенной разницы, где именно будет находиться клиент. Также были разработаны методы снижения нагрузки на сеть. Для того чтобы была возможность пересылать обновления с места возникновения сбоя, была использована система меток, подобная номерам сегментов, используемых в протоколе TCP. Использование применённого механизма приводит к тому, что формат данных, к которому преобразуют все клиенты, такой, как показано в табл. 1.

Таблица 1

Формат данных, передаваемых сервером

№	Поле	Размер, Б	Описание
1	время	36	Время, когда было замечено появление данных
2	датчик	2	Уникальное имя, не зависящее от адреса в системе сбора данных. В случае если данные имеют общий характер, не используется
3	клиент	1	Логический адрес, используемый для обмена в клиент-серверной архитектуре
4	размер	4	Размер передаваемой порции данных
5	данные		Собственно данные
6	CRC	4	Контрольная сумма
7	метка	4	Уникальная метка посылки

После каждой полученной порции данных, сервер отправляет соответствующее подтверждение, в теле которого находится метка. В случае несовпадения кода CRC, сервер запрашивает требуемые данные заново, используя ту же метку. Данные пересылаются в формате XML, который с одной стороны является общепризнанным стандартом, и работа с ним поддерживается многими системами, а с другой – позволяет любые данные привести к единому виду, создав специфические теги, свои для каждого случая. Данные на сервере также сохраняются в XML-формате, что обеспечивает быстрый поиск, возможность быстро отфильтровать данные, добавлять новые данные и объединять сходные. Первоначально рассматривалась идея сохранять информацию на сервере в БД, но некоторые случаи, как например, появление нового типа данных, корректно обрабатываемые при использовании XML, могут привести к трудностям с БД.

## Решение задачи прогнозирования

Для осуществления прогнозирования для автоматизированного управления процессами в работе предлагается на основании накопленных данных искать функциональные зависимости между параметрами с помощью поиска аппроксимирующих кривых. Аппроксимация является одним из основных методов получения эмпирических зависимостей, позволяющим представлять экспериментальные зависимости очень широким диапазоном функций. Используемая в работе аппроксимация осуществляется с помощью метода наименьших квадратов путем линеаризации требуемых зависимостей. Этот метод позволяет аппроксимировать экспериментальные зависимости с достаточно высокой точностью.

При разработке алгоритмов аппроксимации был сделан упор на максимально широкое использование уже готового, хорошо оптимизированного кода. Для этого были рассмотрены наиболее удачные соответствующие проекты, доступ к исходным кодам которых не представлял трудностей. В качестве альтернатив рассматривались:

- LabPlot;
- QtiPlot;
- Approximator;

- Grace;
- Universal Approximator;
- DataFit.

В связи с хорошей документированностью и высокой скоростью работы в качестве основы для разработки был выбран программный продукт Approximator A.P. Сова [13].

Данное программное обеспечение работает по следующему алгоритму. Прежде чем определять численные значения коэффициентов в выбранной эмпирической формуле, проверяется возможность ее использования методом выравнивания. Лишь после этого происходит переход к отысканию тех значений постоянных коэффициентов, которые дадут наилучшее приближение опытных и вычисленных величин. В программе реализована поддержка аппроксимации большим набором формул, однако как уже указывалось в [14] для получения максимально точных моделей требовалась также поддержка зависимостей. Используя исходные коды программы, её функциональность была изменена для работы с требуемым перечнем функциональных зависимостей. Так, интерфейс подпрограммы поиска зависимостей между двумя факторами эксперимента выглядит, как показано на рис. 4.

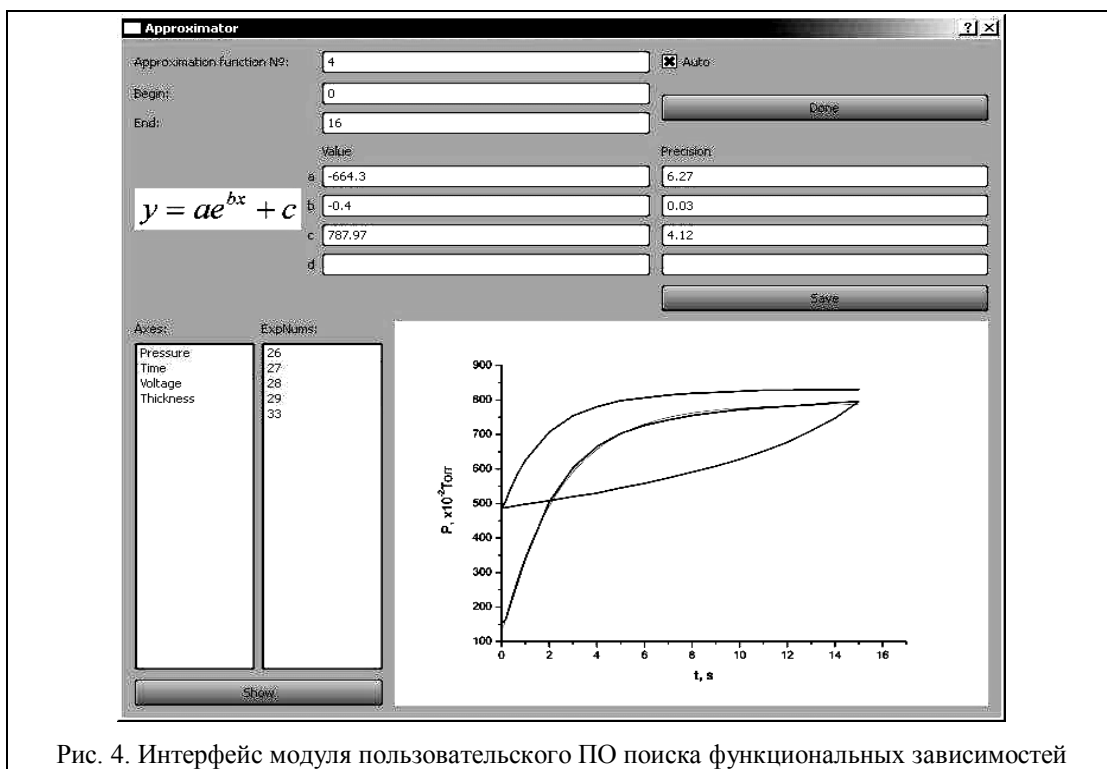


Рис. 4. Интерфейс модуля пользовательского ПО поиска функциональных зависимостей

Здесь после выбора пользователем формулы аппроксимирующей кривой была найдена зависимость

$$P = -664,3 \cdot e^{-x/2,5} + 787,97$$

с соответствующими погрешностями.

Здесь, пользователь выбирает исходные данные, по которым будет построена зависимость, а также

имеет возможность выбрать аппроксимирующую функцию. В данном случае выбор функции осуществляется автоматически, основываясь на корреляционных коэффициентах.

Полученные зависимости затем сохраняются для управления процессом во время активной фазы эксперимента.

## Выводы и перспективы дальнейших исследований

Реализована концепция хранилища данных на примере внедрения в систему управления технологическим процессом ионно-лучевого распыления и осаждения пленок наноразмерного диапазона толщин. Планируется использовать систему для управления другими процессами из области микро и нанотехнологий, что позволит использовать данные, полученные для контроля и стабилизации параметров одного технологического процесса применительно к другим процессам выбранного класса. Несомненно, использование классических СУБД, поддерживающих и распределённость, и возобновление транзакций более удобный и быстрый вариант, однако поставленная задача не может быть решена одними средствами СУБД. Предложенное же решение, хотя и замедляет процесс управления и вносит некую инерционность в систему, добавляя дополнительный уровень, является вполне удобным и простым в использовании. Наряду с использованием механизма поиска функциональных зависимостей между факторами эксперимента система предоставляет возможность построения автоматизированных систем управления, использующих в качестве входных данных как аналитические данные, вводимые пользователем, так и исторические, полученные в ход пассивного эксперимента. В настоящий момент разработанная система проходит также тестирование и адаптацию для технологии ионно-плазменного травления, в ходе которого она уже на данном этапе продемонстрирует удобство в эксплуатации и перспективы развития.

## Список литературы

1. Styervoyedov A. Formation of Ti and TiN ultra-thin films on Si by ion beam sputter deposition / A. Styervoyedov, V. Farenik // *Surface Science*. – 600 (2006). – P. 3766-3769.
2. Рудьнов А.А. Системы автоматического контроля технологических параметров / А.А. Рудьнов, А.И. Беркут. – АСВ, 2005. – 144 с.
3. Шишмарев В.Ю. Автоматизация технологических процессов / В.Ю. Шишмарев. – К.: Academia, 2007. – 352 с.

4. Дерев'янюк А.В. Стабілізація процесу іонно-лучевого осаження ультратонких плінок оксидів і оксинітридів металів / А.В. Дерев'янюк, М.Ю. Силкин, А.Н. Стервєдов // *Радиофизика и электроника: тез. докл. VI Харьк. конф. мол. учёных*. – Х.: ИПЭ, 2007. – С. 58.
5. RS-422 and RS-485 Application Note / B&B Electronics Mfg. Co. Inc [Электронный ресурс]. – Режим доступа: <http://www.bb-elec.com/bb-elec/literature/tech/485appnote.pdf>.
6. Спецификация протокола WAKE [Электронный ресурс]. – Режим доступа: [http://www.spetspribor.com/support/software/wake/dn\\_wake/wake.pdf](http://www.spetspribor.com/support/software/wake/dn_wake/wake.pdf).
7. Дерев'янюк А.В. Система сбора данных и визуализации для управления физическими процессами / А.В. Дерев'янюк, А.А. Новиков, А.Н. Рало // *Сучасні проблеми фізики та хімії на транспорті і в будівництві*. – Х.: ХНАДУ, 2008. – С. 93-95.
8. Гильман А.С. Исследование характеристик протокола канального уровня стандарта ИППС методом имитационного моделирования / А.С. Гильман, М.А. Горбатков // *Имитационное моделирование АСУТП: Сб. науч. тр.* – М., 1986. – С. 50-54.
9. Кирьянов К.Г. Оптимальная дискретизация входных данных для последующей цифровой обработки / К.Г. Кирьянов // *Вестник Нижегородского университета им. Н.И. Лобачевского*. – 2008. – № 1. – С. 39-46.
10. Кирьянов К.Г. Структурная идентификация динамических систем на основе оптимальной дискретизации многоканальных данных / К.Г. Кирьянов // *Вестник Нижегородского университета им. Н.И. Лобачевского*. – 2008. – № 6. – С. 59-69.
11. Operating Rules of IEEE / IEEE 802.3 ETHERNET WORKING GROUP [Электронный ресурс]. – Режим доступа: <http://www.ieee802.org/3/rules/index.html>.
12. Kimball R. The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses / R. Kimball, L. Reeves, M. Ross, W. Thornthwaite. – John Wiley & Sons, 1998. – 780 p.
13. Approximator Homepage [Электронный ресурс]. – Режим доступа: <http://www.aproxim.narod.ru>.
14. Дерев'янюк А.В. Построение эмпирических моделей для управления сложными технологическими процессами / А.В. Дерев'янюк // *Вестник ХНУ имени В.Н. Каразина. Сер. Математическое моделирование. Информационные технологии. Автоматизированные системы управления*. – 2009. – Вып. 12, № 863. – С. 101-110.

Поступила в редколлегию 31.03.2010

**Рецензент:** д-р техн. наук, проф. Г.Н. Доля, Харьковский национальный университет им. В.Н. Каразина, Харьков.

## КОНЦЕПЦІЯ СХОВИЩ ДАНИХ У СИСТЕМІ УПРАВЛІННЯ НАНОТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ

А.В. Дерев'янюк

У роботі описується метод побудови модуля збору та зберігання даних системи управління нанотехнологічними процесами, побудованого на основі програмних модулів, здатних отримувати дані з різномірних джерел і приводити до уніфікованому формату, що прискорює їх подальшу обробку. Використання системи дозволяє однаково ефективно працювати як з поточними експериментальними даними, так і з історичними. Описана архітектура побудованої системи, протоколи роботи її складових частин, а також методи подальшої обробки експериментальних даних.

**Ключові слова:** система автоматизованого управління, збір даних, сховище даних, іонно-променевої синтез.

## THE DATA STORAGE CONCEPT IN THE NANOTECHNOLOGY PROCESSES CONTROL SYSTEM

A. V. Derevianko

The paper describes the constructing method of the collection and data storage module of the nanotechnological processes control system based on the software modules that can receive data from disparate sources and lead to a unified format, which speeds up their further processing. Using the system allows to work effectively both with the current experimental data, and the historical ones. The architecture of the designed system, the protocols of its components, as well as methods of further processing of experimental data are described.

**Keywords:** automatic control, data collection, data storage, ion-beam synthesis.