

## СИСТЕМА ОПЕРАТИВНОГО МОНИТОРИНГА ФИЗИЧЕСКИХ ДАННЫХ В СЕРИИ ЭКСПЕРИМЕНТОВ NA-48, NA-48/1 И NA-48/2

*Е. А. Гудзовский, Д. Т. Мадигожин, Ю. К. Потребеников*

Объединенный институт ядерных исследований, Дубна

Приводится описание системы оперативного мониторинга физических данных, использованной в сеансах набора данных экспериментов NA-48, NA-48/1 и NA-48/2 на ускорителе SPS в ЦЕРН. Система являлась наиболее автоматизированной самостоятельной подсистемой перечисленных экспериментов, позволила неоднократно оперативно обнаружить проблемы в работе используемой аппаратуры и программного обеспечения, а также применялась для калибровки детекторов и настройки экспериментальных условий.

A system of physics data monitoring in real time mode used in the experiments NA48, NA48/1, and NA48/2 at CERN SPS is described. The system was the most automatized subsystem of the above experiments. It often allowed fast detection of problems with the experimental setup and software, and was also used for detector calibration and tuning of the experimental conditions.

PACS: 29.85.+c

### ВВЕДЕНИЕ

Оперативный мониторинг (далее — просто «мониторинг») качества физических данных, получаемых на основе накапливаемой в процессе сеанса экспозиции экспериментальной установки информации, позволяет быстро реагировать на возникающие проблемы и тем самым увеличивать статистику событий, пригодных для окончательного физического анализа. Решение такой задачи в серии экспериментов NA-48 [1], NA-48/1 [2] и NA-48/2 [3], выполненных в рамках программы по каонной физике на ускорителе SPS в ЦЕРН в 1997–2004 г., было возложено на группу сотрудников Лаборатории физики частиц Объединенного института ядерных исследований. Система мониторинга физических данных, наряду с отдельными детекторами экспериментальной установки, системой сбора данных и системой контроля и управления медленными процессами (такими как изменения температур, напряжений и газовых потоков), являлась одной из самостоятельных подсистем указанных экспериментов.

Экспериментальная установка NA-48 представляет собой типичный форвард-спектрометр, предназначенный для регистрации распадов каонов, происходящих в вакуумном распадном объеме. Основными элементами установки являются расположенные последовательно магнитный спектрометр, сцинтилляционный годоскоп, жидкокриптоновый электромагнитный калориметр, адронный калориметр и мюонный детектор.

## 1. ПОСТАНОВКА ЗАДАЧИ

В ходе набора экспериментальных данных требовалось контролировать как сами характеристики записываемых событий (например, параметры реконструированной кинематики событий), так и их стабильность во времени (процентное соотношение восстановленных событий различной топологии, стабильность восстановленных эффективных масс, стабильность загрузок детекторов и т.п.). Сложность практической реализации такой задачи заключалась

- в разнообразии и большом объеме контролируемых данных;
- в необходимости обработки данных в режиме реального времени параллельно с набором экспериментальной информации;
- в необходимости максимальной устойчивости работы мониторинга в режиме реального времени в условиях эксперимента.

Входная информация для системы мониторинга поставлялась системой триггера третьего уровня экспериментов (L3), которая представляла собой программное средство, производящее обработку и предварительную фильтрацию «сырых» («raw») данных для нескольких физических задач с формированием соответствующих потоков данных, впоследствии записываемых на магнитные ленты. За сброс ускорителя («burst»), происходивший приблизительно четыре раза в минуту, система L3 получала один файл в формате сырых данных размером около 500 МБ, содержащий около  $5 \cdot 10^4$  событий. За годовой сеанс экспозиции установки обычно записывалось 200–300 тыс. сбросов. Обработка сырых данных от каждого сброса поручалась одному из 50–70 процессов L3, асинхронно работавших на компьютерной ферме ЦЕРН общего назначения. По завершении обработки очередного файла сырых данных каждый процесс L3 записывал во временном дисковом пространстве по одному файлу для каждого выходного потока, откуда эти файлы стирались после сохранения их копий на лентах. Типичное время хранения данных на дисковом пространстве составляло порядка нескольких часов.

Мониторингу подвергался поток данных специальной компактной формы («compact»), предназначенный для решения основных физических задач эксперимента. Система мониторинга должна была обработать соответствующий файл за время существования его на диске. Она не имела права вносить задержку в процесс сбора данных, однако имела право пропустить часть файлов. Для каждого обработанного файла процесс L3 дописывал строку в конец общего специального текстового файла-журнала (в дальнейшем — просто журнала), в которой содержался, в частности, путь к записанному файлу данных в общей для компьютерной фермы файловой системе. Обработка данных, полученных за сброс ускорителя, завершалась системой L3 не в порядке их получения по времени. Задержка могла составлять от 20 мин до нескольких часов, в зависимости от загрузки процессора, на котором работал конкретный экземпляр процесса L3.

В своей окончательной форме система мониторинга работала под управлением операционной системы Linux на компьютерной ферме ЦЕРН общего назначения.

## 2. РАБОТА СИСТЕМЫ МОНИТОРИНГА

Система мониторинга физических данных состояла из набора управляющих программ (скриптов) на языках perl и csh, программы чтения данных, написанной на языках C

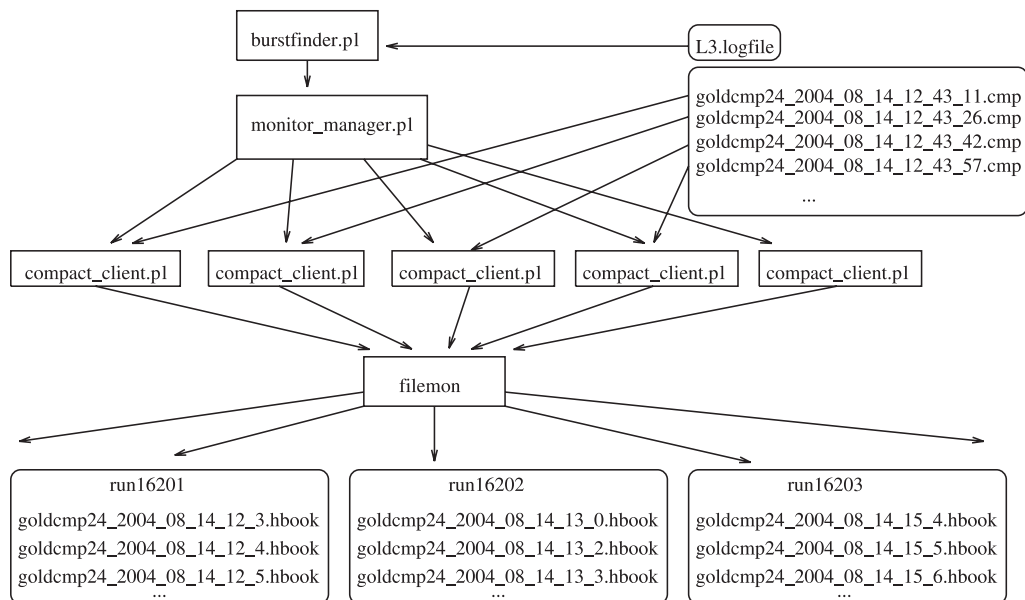


Рис. 1. Схема взаимодействия компонент системы мониторинга

и FORTRAN, и собственно мониторинговой программы, написанной на языке C++. Схема взаимодействия компонент системы, которая будет пояснена ниже, приводится на рис. 1. Отдельно запускались наблюдающие процессы, демонстрирующие результаты работы мониторинга на двух терминалах в комнате дежурного персонала эксперимента, а также обеспечивающие, в случае необходимости, их трансляцию во внешнюю сеть. Система работала на 10–20 компьютерах одновременно.

**2.1. Автоматическая система поддержки целостности.** Запуск системы осуществлялся «демонами» *acrontab*, которые с регулярностью в 10 мин запускали на каждой из машин по процессу одного и того же стартового скрипта. Запущенный процесс устанавливал необходимые переменные окружения и запускал скрипт-«реаниматор». Последний опознавал компьютер, на котором оказался запущен, и проверял специальный общий файл — список процессов, в котором содержалась информация о том, на каком из компьютеров какие процессы и в каком количестве должны работать, а также какое время отсутствия отклика от каждого процесса должно считаться доказательством его зависания или «гибели». Пример списка процессов, работа которых будет описана ниже, соответствующий периоду конца сеанса 2004 г., приведен в таблице.

Получив список процессов, предусмотренных для «своего» компьютера, «реаниматор» проверял, все ли они работают в действительности, и запускал недостающие. Таким образом, вся система мониторинга запускалась автоматически через некоторое время после полного завершения всех процессов или перезагрузки операционной системы.

Работа системы нарушалась как из-за отсутствия некоторых процессов, так и при запуске их лишних экземпляров. Опыт нескольких лет эксплуатации привел к созданию

**Пример списка процессов, поддерживаемых системой мониторинга на различных компьютерах фермы общего назначения**

Компьютер	Имя программы	Кол-во процессов	Время задержки, с
lxplus001	burstfinder.pl	1	120
lxplus001	monitor_manager.pl	1	120
lxplus001	filemon	1	300
lxplus003	compact_client.pl	2	1800
lxplus004	compact_client.pl	2	1800
lxplus005	compact_client.pl	2	1800
lxplus006	compact_client.pl	2	1800
lxplus008	compact_client.pl	2	1800
lxplus011	compact_client.pl	2	1800
lxplus015	compact_client.pl	2	1800
lxplus026	compact_client.pl	2	1800
lxplus042	compact_client.pl	2	1800
lxplus043	compact_client.pl	2	1800
lxplus044	compact_client.pl	2	1800
lxplus055	compact_client.pl	2	1800

описанного ниже механизма контроля и поддержки целостности системы мониторинга. При запуске каждый процесс записывал в специальную контрольную директорию метку старта — пустой файл с именем, содержащим слово «started», системное имя своего компьютера и системный идентификатор процесса. Кроме того, при завершении каждого цикла своей работы каждый процесс перезаписывал там же метку работы — файл, аналогичный метке старта, но со словом «alive» вместо слова «started» в имени. Наличие любой из этих меток считалось признаком работы процесса. «Реаниматор» проверял для каждого из процессов время последней перезаписи метки работы и, если оно было слишком давним, подавал процессу приказ завершения работы путем уничтожения его метки старта. Каждый процесс, если он все-таки работал и просто задержался на чем-то необычно долго, в своем главном цикле проверял наличие своей метки старта и, если ее не было, завершал работу, уничтожая и свою метку работы. Работа процесса завершалась тем же путем, если он не был предусмотрен для данного компьютера в общем списке процессов.

Такой механизм позволял производить изменения числа работающих процессов и миграцию процессов на другие компьютеры без остановки системы, модифицируя список процессов. Запуск новых процессов вместо остановленных происходил при последующих циклах работы «реаниматора», инициированных `acrontab`, и только при отсутствии вышеописанных признаков работы остановленных процессов, что давало время для их нормального завершения.

**2.2. Управляющие скрипты.** Обнаружение новых входных файлов для мониторинга было задачей скрипта-«поставщика» `burstfinder.pl`. Он занимался регулярной проверкой обновлений в журнале `L3`, в котором хранилась информация обо всех обработанных сырых данных. В ранних версиях системы проверка осуществлялась простым чтением дописываемых к концу файла строк. Однако позже потребовалось, чтобы мониторинг самостоятельно реагировал на замену журнала, которая, для ограничения размера журнала,

периодически осуществлялась персоналом, сопровождающим систему L3. Для этого «поставщик» в случае десятиминутного отсутствия новых выработанных L3-файлов заново открывал журнал и проверял по содержимому первой строки, не был ли журнал заменен. Более того, «поставщик» автоматически реагировал на изменения формата записи журнала (при которых необходимая информация перемещалась в другие поля), а также на смену некоторых переменных окружения. В таких исключительных случаях проводилась автоматическая адаптация системы мониторинга, о которой эксперты уведомлялись по электронной почте.

В каждой обнаруженной новой строке журнала L3 «поставщик» отыскивал путь к файлу, в имени которого содержалось имя потока, подлежащего мониторингу. После этого в специальную директорию меток необработанных файлов добавлялся файл-метка с таким же именем, как и у файла данных (состоящим из имени потока и времени записи сброса ускорителя системой сбора данных). Этот файл-метка содержал полный путь к файлу данных сброса в файловой системе.

«Поставщик» сохранял текущее состояние (прежде всего, счетчик уже обработанных строк журнала L3) в служебных файлах. Это позволяло ему восстановить свое состояние и продолжить работу после автоматического перезапуска в случае его гибели.

Метки необработанных сбросов, создаваемые «поставщиком», использовал скрипт управления обработкой `monitor_manager.pl`, который распределял файлы для обработки между асинхронно работающими на различных компьютерах 10–30 экземплярами скрипта-«обработчика» под названием `comract_client.pl`. Число работающих экземпляров «обработчика» подбирались таким образом, чтобы мониторинг не отставал от L3, и могло меняться без остановки работы системы. Чтобы минимизировать задержку информации для дежурной смены, распределение необработанных сбросов всегда начиналось с тех данных, которые были получены самыми последними. Для поиска свободного «обработчика» скрипт управления проверял метки работы процессов этого типа, а также специфические для них файлы — метки загрузки, которые имели в имени слово «data» и системный идентификатор процесса и содержали путь к обрабатываемому файлу. Если метка работы процесса существовала, а метки загрузки не было, управляющий скрипт создавал метку загрузки для этого обработчика, передавая ему таким образом конкретный файл данных. Метка загрузки создавалась путем переименования метки необработанного файла, и таким образом файл исключался из числа необработанных. По окончании обработки сброса эта метка загрузки уничтожалась уже самим процессом-«обработчиком» и процесс снова становился доступным для управляющего скрипта.

Каждый процесс-«обработчик» при запуске создавал временную рабочую директорию на локальном диске своей машины (в директории `\tmp`), куда, для минимизации загрузки сети, записывалась копия исполняемого файла читающей программы `comract`, а также все временные файлы, возникающие в процессе обработки данных. Эта временная директория уничтожалась при завершении работы процесса путем уничтожения его метки старта. Если перед очередным запуском обрабатываемой программы обнаруживалось, что временная рабочая директория уничтожена (чего нельзя исключить при использовании файловой системы `\tmp`), она создавалась заново. В своем главном цикле «обработчик» проверял наличие приготовленной для него метки загрузки и запускал программу чтения данных на обработку файла, указанного в этой метке.

**2.3. Программа чтения данных.** Программа чтения данных состояла из управляющего модуля, который осуществлял распаковку индивидуального события, после чего

вызывал служебные модули. Служебные модули, содержащие алгоритмы контроля аппаратуры и анализа данных, разрабатывались отдельно для каждого из экспериментов NA-48, NA-48/1 и NA-48/2 с привлечением экспертов по детекторам, а также специалистов по анализу всех стоящих в эксперименте физических задач. Каждый пользователь мог включить в программу свой служебный модуль. Помимо указанных модулей, существовал специальный служебный модуль, функциями которого был подсчет количества прочитанных сбросов, обработанной статистики и событий, прошедших определенные наборы критериев отбора.

Результатами обработки каждого сброса являлся набор гистограмм, записываемый в файл формата HBOOK [4], имя которого содержало время записи сброса. Этот файл помещался из локальной директории компьютера в поддиректорию системы мониторинга, предназначенную для входных данных мониторинговой программы. Если по окончании работы читающей программы результата обработки не было, процесс-«обработчик» сохранял файл-журнал работы программы чтения в отдельной директории для «неудачных» сбросов. В любом случае, по окончании обработки сброса «обработчик» уничтожал метку своей загрузки.

Модули мониторинга, отвечавшие за основную физическую задачу эксперимента в сеансах 1998–2001, а также 2003–2004 гг., были разработаны сотрудниками ЛФЧ ОИЯИ.

**2.4. Мониторинговая программа.** Задачи объединения информации с отдельных сбросов, построения временных зависимостей, хранения и представления итоговых данных по периодам работы эксперимента выполняла программа-монитор `filemon`, которая сохраняла на диске информацию о каждом существенном шаге своей работы и поэтому после любого перезапуска всегда была способна восстановить свое последнее рабочее состояние.

В главном цикле работы эта программа проверяла содержимое директории, предназначенной для новых файлов гистограмм, куда они помещались процессами-«обработчиками». Имя файла гистограмм содержало время записи сброса экспериментальной установкой, и этого было бы достаточно для построения временных зависимостей любых интересующих величин. Однако более удобным было сочтено применение двухуровневой временной шкалы. Экспериментальный сеанс делился на периоды с относительно однородными условиями, так называемые прогоны («run»). Информация обобщалась отдельно для каждого прогона, а внутри него — по 10-минутным интервалам времени.

Монитор начинал очередной цикл своей работы с того, что выбирал 10-минутный интервал, содержащий наибольшее число необработанных им файлов гистограмм, либо просто самый недавний интервал, в котором есть новые файлы (в зависимости от конфигурации, выбранной при компиляции). Все односбросовые файлы гистограмм из этого интервала читались и уничтожались, и все их гистограммы добавлялись к соответствующим гистограммам объединенных 10-минутных файлов, накапливаемых для каждого интервала. Эти объединенные файлы создавались в директориях, соответствующих каждому прогону сеанса. После этого для прогона, о котором поступила новая информация, на основе 10-минутных файлов гистограмм обновлялись все временные зависимости, хранившиеся в виде специальных гистограмм в каждой директории прогона. Также обновлялись суммарные файлы гистограмм для каждого прогона. Таким образом, работа монитора совершенно не зависела от порядка поступления файлов гистограмм.

Демонстрация результатов мониторинга в комнате дежурного персонала эксперимента осуществлялась для последнего прогона, о котором поступила информация от L3. С по-

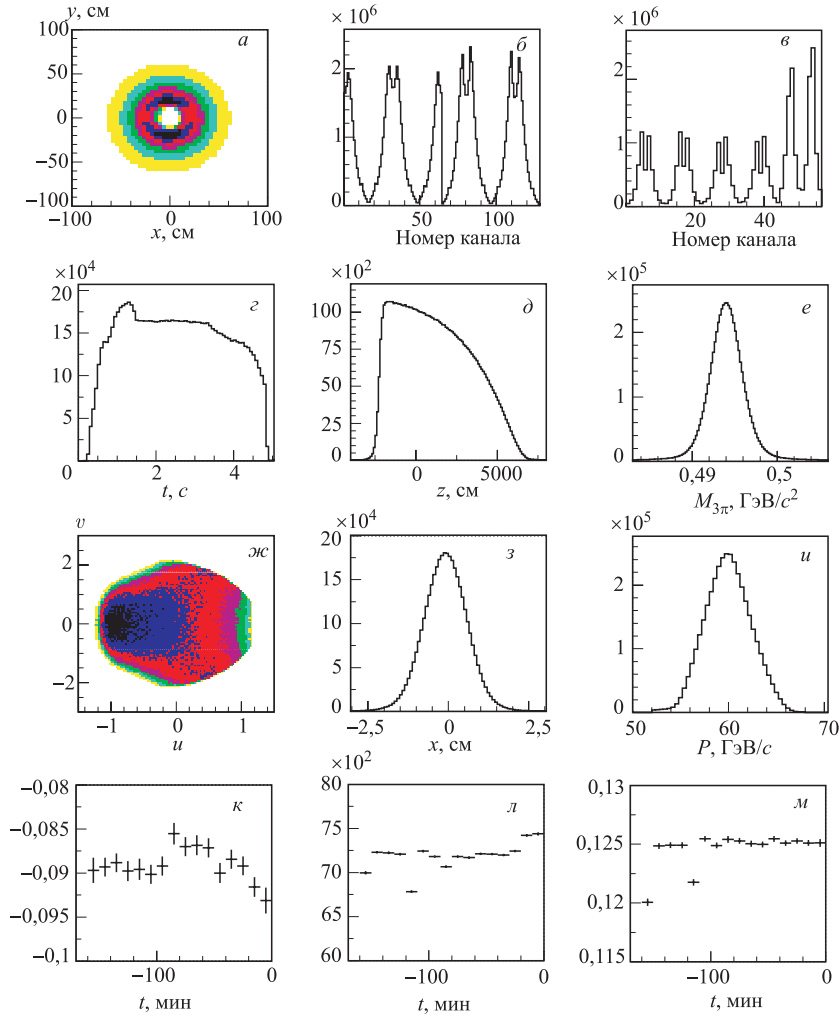


Рис. 2. Примеры гистограмм, формируемых системой мониторинга: а) загрузка одной из дрейфовых камер; б) счетчик срабатываний каналов годоскопа; в) счетчик срабатываний каналов мюонного детектора; г) распределение записанных событий по времени внутри сброса; д) распределение  $z$ -координат реконструированных вершин; е) восстановленная масса заряженного каона; ж) распределение событий  $K_{3\pi}^{\pm}$  по кинематическим переменным; з) профиль пучка в  $x$ -плоскости; и) импульсный спектр пучка; к) временная зависимость  $x$ -положения пучка; л) временная зависимость выхода событий  $K_{3\pi}$  за сброс; м) временная зависимость доли реконструированных  $K_{3\pi}$  среди записанных событий

мощью разработанного во FNAL пакета визуализации данных HISTOSCOPE [5] демонстрировались регулярно обновляемые суммарные гистограммы прогона в процессе накопления в них статистики, а также графики временных зависимостей таких величин, как эффективности триггеров, реконструированные импульсы и положения пучков, набираемая статистика по наиболее интересным физическим процессам. После запуска

очередного прогона дежурная смена считала, что все в порядке, только когда через 10–20 мин наблюдала несколько десятков распределений по реконструированным энергиям, координатам и массам частиц, а также распределения координат точек срабатывания в основных детекторах для данного прогона. Участники эксперимента могли подключить свой клиент HISTOSCOPE и наблюдать за ходом сеанса в удаленном режиме. Также можно было изучать и сами гистограммы в формате HBOOK, открытые для чтения в файловой системе. Примеры формируемых гистограмм приведены на рис. 2.

Был разработан ряд инструментов для автоматического и интерактивного суммирования информации мониторинга, которые применялись, в частности, для достаточно точной оценки набранной статистики. В сеансах 2003 и 2004 гг., когда в соответствии со спецификой физической задачи эксперимента было необходимо соблюдать баланс статистики, собранной при различных конфигурациях магнитных полей спектрометра и пучковой линии, решения о переключении конфигураций принимались именно на основе данных мониторинга.

### **3. РЕЗУЛЬТАТЫ РАБОТЫ**

Мониторинг физических данных устойчиво работал в течение семи лет во всех сеансах набора данных экспериментов NA-48, NA-48/1 и NA-48/2, начиная с сеанса 1998 г. Ежегодно производилась доработка и пересмотр обрабатывающей программы, чтобы обеспечить мониторинг наиболее существенных для текущего сеанса величин и распределений. Контроль подвергалось до 200 гистограмм и 10–15 временных зависимостей различных параметров. Около 30 графиков постоянно демонстрировались в комнате дежурного персонала эксперимента.

Мониторинг неоднократно позволял оперативно обнаружить проблемы в работе аппаратуры и программного обеспечения экспериментов, которые без его применения были бы обнаружены либо значительно позже, либо лишь на этапе физического анализа данных. Таким образом, мониторинг внес существенный вклад в увеличение полезной статистики всех перечисленных выше экспериментов. В ряде случаев гистограммы, полученные в мониторинге, использовались для оперативной калибровки аппаратуры, в частности, мюонного детектора. В сеансе 2004 г. мониторинг использовался для настройки системы прицеливания пучков.

После получения калибровочных данных в рассматриваемых экспериментах производилась переобработка данных с помощью процессов L3 с образованием выходных потоков, аналогичных тем, что записывались в ходе сбора данных. Процесс переобработки также мог контролироваться разработанной системой мониторинга. В 1999 г. работало одновременно две системы мониторинга, одна из которых контролировала сеанс на ускорителе, а другая — переобработку данных прошлого года, которая шла в то же время.

Мониторинг физических данных был наиболее автоматизированной системой серии экспериментов NA-48, поэтому инструкции по управлению им дежурной сменой сводились к рекомендации в случае любой проблемы закрыть демонстрационные окна HISTOSCOPE (чаще всего проблемы были связаны именно с ним или с соответствующим терминалом), и, в крайнем случае, — завершить процесс мониторинга. После этого система возобновляла свою работу в течение не более чем 10 минут.



**Благодарности.** Авторы выражают особую благодарность А. Л. Ткачеву и А. Н. Гапоненко, внесшим значительный вклад в разработку системы мониторинга физических данных на начальной стадии ее создания, В. Д. Кекелидзе за полезные обсуждения и постоянное внимание к работе, а также всем участникам коллабораций NA-48, NA-48/1 и NA-48/2, внесшим вклад в развитие прикладных возможностей системы.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Barr G. D. et al.* Proposal for a Precision Measurement of  $\epsilon'/\epsilon$  in  $CP$ -violating  $K_0 \rightarrow 2\pi$  decays. CERN-SPSC-90-22. 1990.
2. *Batley R. et al.* Addendum 2 to Proposal P253/CERN/SPSC. A High Sensitivity Investigation of  $K_S$  and Neutral Hyperon Decays Using a Modified  $K_S$  Beam. CERN/SPSC 2000-002. 2000.
3. *Batley R. et al.* Addendum 3 to Proposal P253/CERN/SPSC. Precision Measurement of Charged Kaon Decay Parameters with an Extended NA48 Setup. CERN/SPSC 2000-003. 2000.
4. HBOOK Reference Manual. CERN Program Library Long Writeup Y250. 1994.
5. Histo-Scope Plotting Widget Set; <http://www.fnal.gov/fermitools/abstracts/plotwidgets/abstract.html>

Получено 20 июня 2006 г.