

СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ КАРЕТКОЙ ОБЛУЧАТЕЛЯ № 1 РАДИОТЕЛЕСКОПА РАТАН-600

В. Н. Черненко, Б. Л. Ерухимов

Описана структура комплекса, аппаратно-программные средства и алгоритм, позволяющие реализовать прозрачный режим эксплуатации системы управления.

The system structure, hardware and software, and algorithm which allows to realize the transparent mode of use of this system are described.

1. Введение

Необходимость качественного повышения уровня научных исследований, в том числе результативности труда радиоастронома-наблюдателя, требует разработки такой аппаратуры и ее программной поддержки, которая минимизирует интеллектуальные затраты исследователя на решение промежуточных задач, обусловленных лишь сложностью инструмента. Такие системы, обычно диалоговые, называют «прозрачными».

Автоматическое управление перемещением первичных рупоров антенны РАТАН-600 необходимо для реализации режима «скольжения» [1], а также предпосылки для безлюдной технологии наблюдательного процесса.

Режим наблюдений с использованием перемещения каретки является для наблюдателя одним из штатных режимов наблюдений. При подготовке пакета источников наблюдателю требуется заполнить набор экранных бланков, каждый из которых соответствует наблюдению конкретного радиоисточника или участка неба [2]. В бланк заносятся имя источника, его координаты из каталога, азимут наблюдения, длины волн используемых радиометров, длина волны радиометра, рупор которого должен быть в фокусе (если рупоров два — то западный, восточный или середина), коэффициент скольжения D , определяющий скорость перемещения каретки. Так, при $D=1$ реализуется режим слежения за источником, при меньших значениях — скольжение, а при $D=0$ — неподвижный режим однократного прохождения.

Для режима прохождения дополнительно указывается временной интервал относительно прямого восхождения источника. В остальных случаях этот интервал рассчитывается автоматически, исходя из возможного времени нахождения изображения источника в поле зрения радиотелескопа.

Заполнение бланков — это процедура, включающая фактически весь объем ручной подготовки наблюдений. Все остальные процедуры, связанные с расчетом времени начала движения, регистрации скорости, числа дискретов записи и управления и т. п., выполняются автоматически и оформляются в виде файлов — заготовок на дисковом носителе.

На облучателях РАТАН-600 первичные рупоры и радиометрические приемники смонтированы на подвижной каретке, общий вес которой вместе с аппаратурой свыше 1000 кг. Решение задачи автоматического перемещения каретки по программно заданной траектории приводит к схеме, аналогичной используемой при числовом программном управлении металлорежущими станками (ЧПУ). Типовая реализация этой схемы использована в системе управления

кареткой облучателя № 2 [3]. Вместе с тем существенные особенности процесса наблюдений, указанные ниже, требуют специфического подхода к разработке системы управления.

1) Траектория движения каретки облучателя в отличие от подвижной части станка должна быть жестко привязана к сетке звездного времени.

2) Система управления кареткой должна органически вписываться в общую структуру системы регистрации и обработки наблюдений и быть «прозрачной» для наблюдателя.

В соответствии с этими требованиями была разработана система управления, базирующаяся на разработанной и внедренной авторами ранее трехуровневой структуре автоматизированной поддержки наблюдательного процесса [2, 4, 5]. Ниже описываются алгоритм функционирования и прикладное аппаратно-программное обеспечение следящей системы.

2. Алгоритм работы следящей системы

При разработке алгоритма отработки траектории движения кроме отмеченных особенностей всей системы учитывались следующие факторы:

1) большие вариации параметров объекта управления, так как масса, центр тяжести, коэффициент трения подвижной каретки меняются при установке и смене радиометров и другой измерительной аппаратуры;

2) существенный дрейф коэффициента передачи исполнительного электропривода при больших вариациях внешней температуры (от -20 до $+40$ °C), влажности (5—98 %) и давления.

Таким образом, разрабатывалась система управления, параметры которой являлись существенно переменными. Поэтому для получения приемлемого качества работы желательным было охватить обратной связью максимальное число звеньев системы.

При выборе места введения управляющего воздействия в контур управления следовало определить, отслеживать ли ошибку по скорости, получаемой из разности координат в дискретные временные отсчеты, или формировать сигнал рассогласования самих координат.

Замыкание обратной связи только по скорости из-за вышеупомянутых факторов требует существенных эксплуатационных затрат, например периодического экспериментального выяснения момента начала движения для устранения систематической координатной ошибки. Выявление этих недостатков на эксплуатировавшейся системе управления кареткой облучателя № 2 [3] потребовало пересмотреть как программно-алгоритмический, так и аппаратный способы реализации системы, обрабатывающей непосредственно координатное рассогласование.

Решение проблемы устойчивости координатной системы при минимуме динамической ошибки было осуществлено известным способом введения форсирующего звена, т. е. звена, предвычисляющего заранее выходную реакцию системы и заставляющего отслеживать координату с опережением. Величина опережения берется достаточно малой, такой, чтобы можно было пренебречь ошибкой предварительного расчета. При этом предрасчет реализуется простой линейной экстраполяцией текущих координат при заданной мгновенной скорости движения.

В такой системе величина управляющего воздействия, подводимого к электроприводу в N -й временной дискрет, рассчитывается по следующей формуле:

$$U(N) = ((X(N)_r + V_r \cdot dT) - X(N)_i) \cdot k1, \quad (1)$$

где $X(N)_r$ и $X(N)_i$ — величины расчетной и измеренной координат; $V_r \cdot dT$ — расчетная скорость, умноженная на временной дискрет; $k1$ — масштабный коэффициент.

Нетрудно заметить, что перегруппировав правую часть, можно получить выражение, аналогичное закону пропорционально-интегрального скоростного

регулятора (ПИД) [6] с нулевым значением коэффициента при дифференциальном члене:

$$U(N) = ((X(N)_r - X(N)_i) \cdot k_1 + V_r \cdot k_2, \quad (2)$$

где разность координат эквивалентна интегральному члену, а k_1 и k_2 — масштабные коэффициенты.

Заметим, что эквивалентность сохраняется только в предположении, что ошибки измерений и вычислений отсутствуют, однако это условие не соблюдается. Ошибка счета интегрального члена в последнем случае накапливается пропорционально времени работы регулятора, вследствие увеличения числа суммируемых членов, а также невысокой точности измерения скорости как разности координат соседних отсчетов при малой скорости движения и фиксированной частоте опроса датчика. Использование же переменной частоты дискретизации [6] для повышения точности вычислений невозможно из-за необходимости точной координатно-временной привязки.

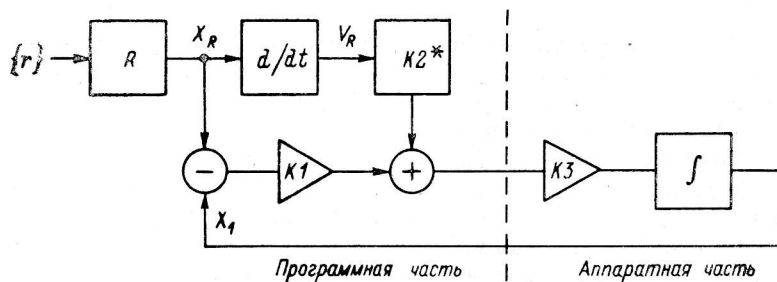


Рис. 1. Структурная схема непрерывного эквивалента системы управления.

Структурная схема непрерывного эквивалента системы представлена на рис. 1. На этой схеме справа от вертикальной штриховой линии изображены эквивалентные звенья электропривода: усилитель K_3 и интегратор, а слева — звено генератора траекторий R , дифференцирующее звено d/dt , масштабные усилители K_1 и K_2 , схема сравнения и сумматор. Все звенья слева реализованы программно в ЭВМ системы управления. В процессе работы системы выборка из множества режимов наблюдений (r) порождает один из траекторных массивов X_r и массив соответствующих скоростей V_r после дифференцирования. В каждый дискретный момент времени на выходе схемы сравнения получается значение координатной ошибки $X_r - X_i$, которое после масштабного преобразования суммируется с мгновенным значением скорости и после цифроаналогового преобразования воздействует на привод как управляющий сигнал. Следует отметить, что звено K_2 оснащено инерционным элементом из-за необходимости защиты привода от больших ускорений.

Расчет оптимальных параметров звеньев в данной работе не описан, поскольку в первом приближении эквивалентен расчету классического ПИД регулятора, основное же внимание уделено реализации конкретной системы управления.

3. Архитектура и аппаратура системы управления

В основу разработки системы управления были положены следующие принципы:

- обеспечение «прозрачного» для пользователя режима эксплуатации;
- комплектование аппаратных средств из набора серийных элементов, использующихся в системах автоматизации научных исследований.

Реализация первого принципа осуществляется включением ЭВМ системы управления в состав двухмашинного комплекса локальной информационно-управляющей структуры (ЛИУС) [2, 4, 5], поддерживающей наблюдательный

процесс на облучателе № 1 РАТАН-600. При этом ЭВМ системы регистрации оснащена терминалом наблюдателя и является ведущей, а ЭВМ системы управления — ведомой.

В обеспечение второго принципа авторам удалось избежать специализированных аппаратных разработок и реализовать алгоритм управления программными средствами, с использованием стандартных КАМАК-модулей.

Таким образом, аппаратура системы управления включает:

- 1) одноплатную микро-ЭВМ НМС-1101.01;
- 2) КАМАК-крейт с крейт-контроллером;
- 3) набор стандартных модулей (ЦАП, выходной регистр, входные ворота);
- 4) четырнадцатиразрядный цифровой датчик угла поворота с устройством кодопреобразователем типа ЗПК-14.

Система управления воздействует на электропривод через упомянутый цифроаналоговый преобразователь (ЦАП). Код обратной связи формируется цифровым датчиком угла поворота, при этом величина линейного перемещения каретки преобразуется в угловую с помощью безлюфтового зацепления зубчатой рейки с шестеренкой на оси датчика. Если использовать терминологию, принятую в теории автоматического регулирования, то можно сказать, что ЭВМ является генератором управляющего воздействия, дискриминатором ошибки и цифровым фильтром, оптимизирующим выходное воздействие для привода.

Аппаратное комплексирование ведомой ЭВМ системы управления с ведущей ЭВМ системы регистрации осуществлено в соответствии с концепцией двухмашинного комплекса ЛИУС [5] через встроенный порт параллельного байтового обмена (ИРПР). Этот порт соединен с таким же портом ЭВМ системы регистрации; синхронизирующий сигнал звездного времени с частотой около 100 Гц вводится по цепи радиального прерывания прямо в канал ЭВМ (к-прт) с выхода таймера системы регистрации.

Таким образом, система управления комплектуется полностью из стандартных заводских изделий, легко дублируется и не требует настройки аппаратуры.

4. Программное обеспечение системы управления

Программное обеспечение системы управления состоит из двух частей, размещаемых соответственно в ведущей ЭВМ системы регистрации и ведомой ЭВМ управления, которую мы будем называть контроллерной.

В ЭВМ системы регистрации — это оболочка из базовых модулей для инициализации контроллерной ЭВМ, загрузки и поддержки управляющей программы, а в контроллерной ЭВМ — загружаемое ядро, вызывающее из своего тела набор процедур, реализующих динамический расчет траектории, и собственно алгоритм автоматического сопровождения по полученным данным.

Все программные модули написаны на языке ассемблера MACRO-11 с использованием библиотеки структурных макрорасширений MACRO-11/SP.

Базовые модули оболочки имеют следующее назначение:

ISSL — подготавливает ведущую ЭВМ к обмену с контроллерной ЭВМ, инициализирует контроллерную ЭВМ и загружает в нее первичный загрузчик;

IFDRL — осуществляет чтение и транзитную передачу загрузочного файла ядра с магнитного носителя центральной ЭВМ по линии связи через ЭВМ системы регистрации и байтовый канал в память контроллерной ЭВМ;

LOACAR, PUTCAR — дополнительные транспортные модули, входящие в состав ассемблерной части программы ведущей ЭВМ, пересылают байты управляющих команд (INPUT, AGAIN, STOP и др.).

Следует отметить, что модули оболочки инвариантны к объему и функциональному содержанию ядра, являясь в этом смысле базовыми и пригодными для любого контроллера на базе ЭВМ НМС-1101.01.

В тексте программы регистрации (на ФОРТРАНе) обращение к модулям оболочки выглядит как вызов функций, возвращающих нулевое значение или код ошибки.

Вызов LOACAR выглядит так:

CALL LOACAR (AREA, NUMB),

где:

1) AREA — имя массива параметров, который передается ведущей ЭВМ в контроллерную;

2) NUMB — размер массива в байтах.

Аналогично для PUTCAR:

CALL PUTCAR (B),

где B — байтовая константа (управляющий символ, например «P» — включение питания, «T» — ввод таблицы параметров, «e» — отработка фиксированной координаты и др.).

4.1. Структура и функции ядра

Ядро состоит из трех программных секций, размещаемых в ОЗУ контроллерной ЭВМ в перечисленном порядке:

1) абсолютная секция, описывающая точки входа процедур прерывания от ИРПР, таймера, ошибок арифметического переполнения, тайм-аута шины и сбоя питания;

2) OTS \times I — перемещаемая секция чистого кода программы;

3) USER \times D — перемещаемая секция постоянных и переменных данных программы, где с последней метки программы (track) размещается расчетная траектория движения.

При таком расположении программных секций загружаемая часть ядра размещается в начальных адресах и занимает менее 5 % объема оперативной памяти, так что функция IFDRL выполняется за доли секунды.

Алгоритмы функционирования основных модулей ядра изображены на рис. 2.

Главным модулем ядра является процедура CONTR, которая после постройки стека и установки бита разрешения прерывания от входного порта попадает в бесконечный цикл ожидания прерываний с проверкой флага. Прерывания могут быть от таймера или от вводимой команды, причем допускается вложение таймерных запросов в командные, но не наоборот. Вложенность команд исключается снятием бита разрешения прерывания по вводу до завершения выполнения текущей команды. Таким образом, двухэлементная командная очередь со стороны оболочки формируется путем простой проверки готовности порта на вывод. На время же таймерных запросов приоритет процессора поднят, что гарантирует жесткую синхронизацию процессов реального времени.

Таким образом, можно считать, что в контроллере параллельно существуют три процесса (рис. 2):

1) фоновый — CONTR;

2) оперативный с низким приоритетом, возбуждаемый портом ввода — RES;

3) оперативный с высоким приоритетом, возбуждаемый таймером — CLOCK.

ТАБЛИЦА 1

Код команды	Имя модуля	Что делает
P	AGAIN	Включает питание привода каретки
a	DONE	Завершает текущий процесс и выключает питание привода каретки
б	ACSLR	Осуществляет пуск и ускоренное движение вправо
ц	ACSLF	То же, влево
д	VELR	Осуществляет пуск и движение по рассчитанной траектории
e	POINS	Осуществляет установку каретки в заданную точку
ф	FINAL	Выключает все внешние устройства, останавливает процессор
г	INPUT	Осуществляет ввод таблицы параметров и рассчитывает начало траектории

Функция первого процесса будет рассмотрена в разд. 3.2, здесь же удобнее начать с процесса RES.

Процесс RES запрещает прерывания от входного порта, сохраняет в стеке необходимые регистры оперативного назначения (РОН) и регистр состояния крейт-контроллера, затем читает содержимое порта и использует его как указатель в таблице процедур (команд), одну из которых вызывает. Завершается

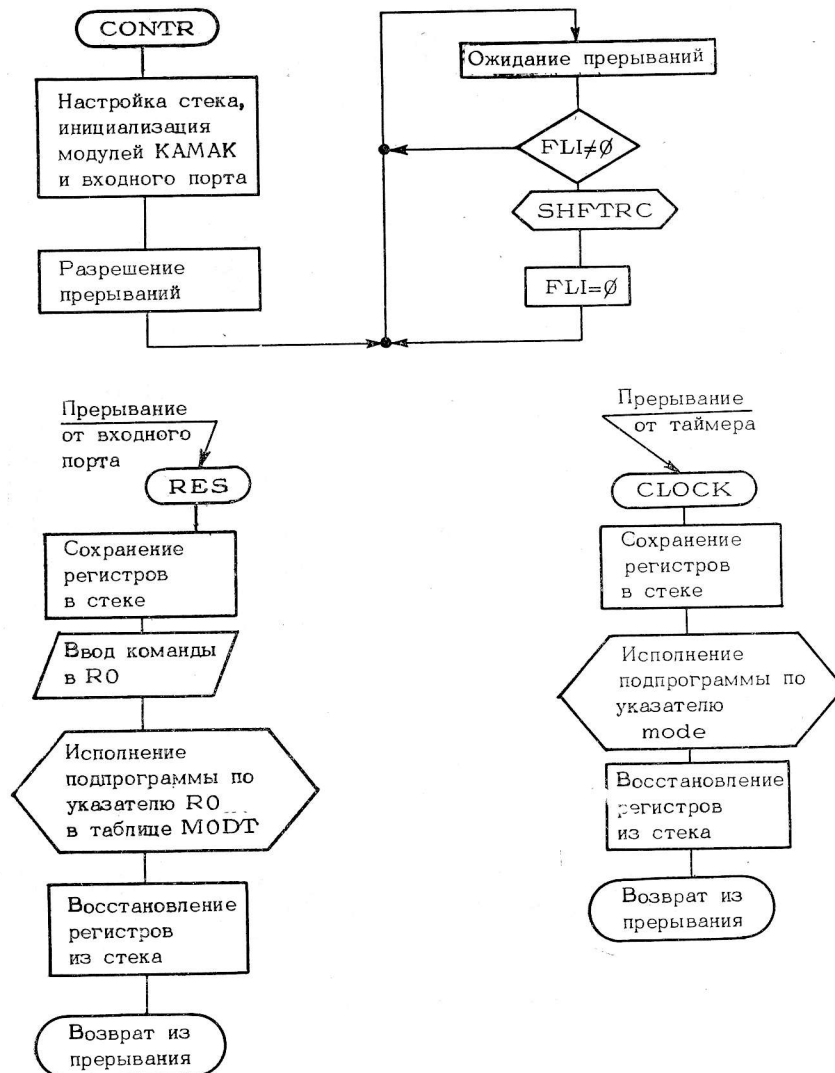


Рис. 2. Алгоритмы функционирования основных модулей ядра.

RES восстановлением сохраненных данных в обратном порядке и установкой бита разрешения прерывания.

RES может вызывать ряд процедур, оформленных в виде компактных модулей. Функции этих модулей иллюстрируются в табл. 1.

Необходимость строгой синхронизации выполняемых процедур реального времени требует такой организации порядка их вызова, при котором в случае приема команды указанная процедура выполняется не как переход в подпрограмму, т. е. немедленно, а планируется на момент прихода импульса от таймера. В операционных системах, выполняющих множество асинхронных задач по обслуживанию нескольких внешних устройств, планирование обычно заключается в записи имени указанного процесса в очередь процессов согласно прио-

ТАБЛИЦА 2

Имя модуля	Вызывает модули	Меняет MODE	Что делает
DMEAS	По BMODE	—	Формирует импульсы опроса датчика и считывает код угла поворота, по сигналу концевого выключателя вызывает процедуру, указываемую BMODE
TIMZER	DMEAS	—	Формирует код величины рассогласования между текущим отсчетом датчика и нулем, при многократном вызове позволяет устанавливать каретку в нулевое положение
POINT FONDLY	DRIVER DRIVER	+ +	То же, для произвольной точки
MEASU	DRIVER	+	Формирует задержку движения для регистрации фонового уровня неба
DRIVER	DMEAS	—	Отрабатывает текущее значение координаты, задаваемой массивом TRACK, по достижении границы массива устанавливает флаг перерасчета траектории
RETRN		—	Измеряет величину рассогласования заданной и действительной траекторий, рассчитывает величину управляющего воздействия для ЦАП
			Пустая операция

риту, обусловленному срочностью выполняемой задачи. В данном случае, поскольку объект управления всего один, планировщик является вырожденным и содержит всего один элемент очереди — ячейку памяти MODE, куда помещается имя (адрес точки входа) планируемой процедуры.

Процесс CLOCK начинает функционировать сразу, как только CONTR оформит стек и снизит приоритет процессора. Структура подпрограммы, реализующей этот процесс, состоит из основного модуля и нескольких ветвей, планируемых процедурами из RES или внешними событиями. Основной модуль имеет точку входа CLOCK, указываемую вектором обработки прерываний от таймера и завершается инструкцией RTI. Внутри, как и в RES, производится сохранение регистров и состояния КАМАК и вызывается процедура, на которую указывает содержимое ячейки MODE. То есть любая из процедур может записать в MODE имя желаемой операции, которая прервет на следующем такте любую другую (в том числе и себя) и гарантированно будет выполняться в течение 10 мс. Аналогичную роль выполняет содержимое ячейки — BMODE, являющееся указателем процедуры, выполняемой при срабатывании датчика начального положения каретки.

Таким образом, несмотря на наличие многопроцессного функционирования, сохранена быстрая реакция на изменения внешних условий, с задержкой не более одного временного дискрета.

4.2. Динамический расчет траектории

Построение координатной системы автоматического регулирования требует безусловного знания необходимого значения координаты в каждый текущий временной отсчет. Эти значения могут быть вычислены заранее и находиться в оперативной памяти в виде списка. Второй способ состоит в расчете нового значения каждый раз по мере отработки предыдущего значения. Если первый способ требует значительного объема оперативной памяти, то второй — высокого быстродействия процессора. Оценки возможностей использования этих способов показали их нереализуемость в чистом виде на доступных микро-ЭВМ. Вследствие этого авторами выбран комбинированный вариант, частично включающий оба способа расчета.

Наблюдения можно проводить неограниченно долго, если рассчитывать траекторию порциями на величину отводимого для этой цели объема памяти. Скорость счета при этом уже не критична и расчет легко реализуется стандартным набором подпрограмм из библиотеки ФОРТРАНа. Расчет может производиться в фоновом режиме и инициироваться по флагу при отработке половины рассчитанной ранее траектории.

В описываемой системе управления такой динамический расчет осуществляется фоновым процессом CONTR по флагу, устанавливаемому оперативной процедурой MEASU (из процесса CLOCK). Флаг устанавливается при достижении указателя обрабатываемой точки середины или конца массива TRACK, причем содержимое флагового слова является указателем физического адреса начала области памяти, куда необходимо поместить новую порцию траектории. Перед началом расчета флаг сбрасывается, вызывается фоновая процедура SHFTRC, затем CONTR вновь переходит в состояние ожидания флага.

Очевидно, что расчет траектории должен быть произведен быстрее, чем указатель достигает контрольной точки в TRACK.

5. Заключение

Прозрачный режим системы управления кареткой закладывался как основной принцип разработки. Прозрачность реализуется использованием базовой трехуровневой иерархии аппаратно-программных средств (МИКРАТ) и прикладной системы поддержки наблюдательного процесса [2, 5], осуществляющей распределение вычислительных задач по сложности и требуемой скорости реакции между тремя процессорами ХОСТ-ЭВМ, спутниковой ЭВМ системы регистрации и контроллерной ЭВМ управления.

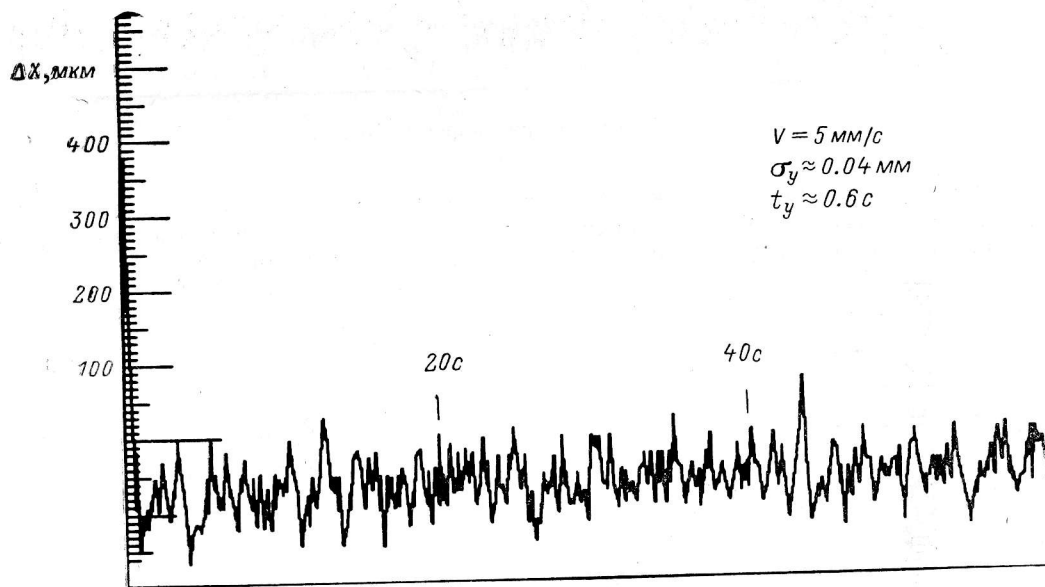


Рис. 3. Экспериментальная запись ошибки слежения при воздействии скачка скорости в 5 мм/с.

Программа регистрации наблюдений загружается в спутниковую ЭВМ системы регистрации (ведущую) средствами МИКРАТ после набора одной команды LOAD. Эта программа в числе модулей поддержки регистрации содержит описанную выше оболочку, которая сразу после загрузки иницирует контроллерную ЭВМ, считывает с диска ХОСТ-ЭВМ файл ядра контроллера и загружает его. Данные, относящиеся к расчету траектории движения, берутся из служебного заголовка (служебной зоны) файла-заготовки, ретранслируются оболочкой контроллеру процедурами LOACAR и PUTCAR, а в ядре принимаются RES и INPUT. Затем оболочка выдает команды включения питания привода и начальной установки каретки на нулевую или заданную точку. Момент начала движения и регистрации отслеживается по программным часам ведущей ЭВМ согласно данным заголовка файла-заготовки, принятого по линии связи с ХОСТ-ЭВМ. По окончании движения питание привода выключается командой DONE.

Работа всех этих процедур происходит вне внимания наблюдателя, для которого необходимость диалога с регистрирующей ЭВМ и тем более с управляющей во время эксперимента практически сведена к нулю. Если необходимо изменить

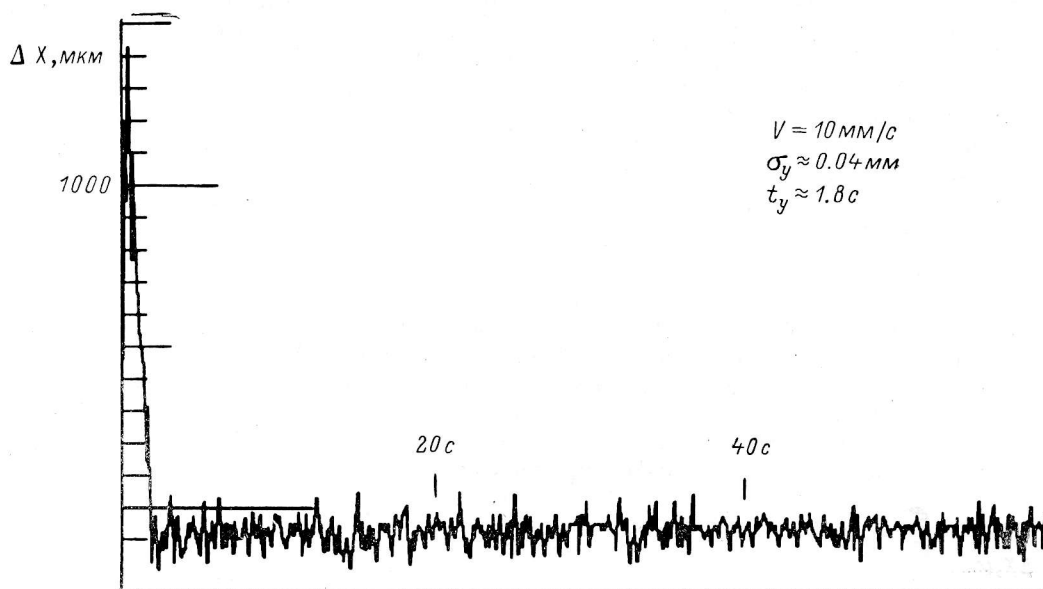


Рис. 4. То же при скорости 10 мм/с.

условия эксперимента, наблюдатель может очень быстро подготовить новое задание (или отредактировать старое), оперируя, опять же, понятными ему физическими величинами: координатами источников, диапазоном волн и др.

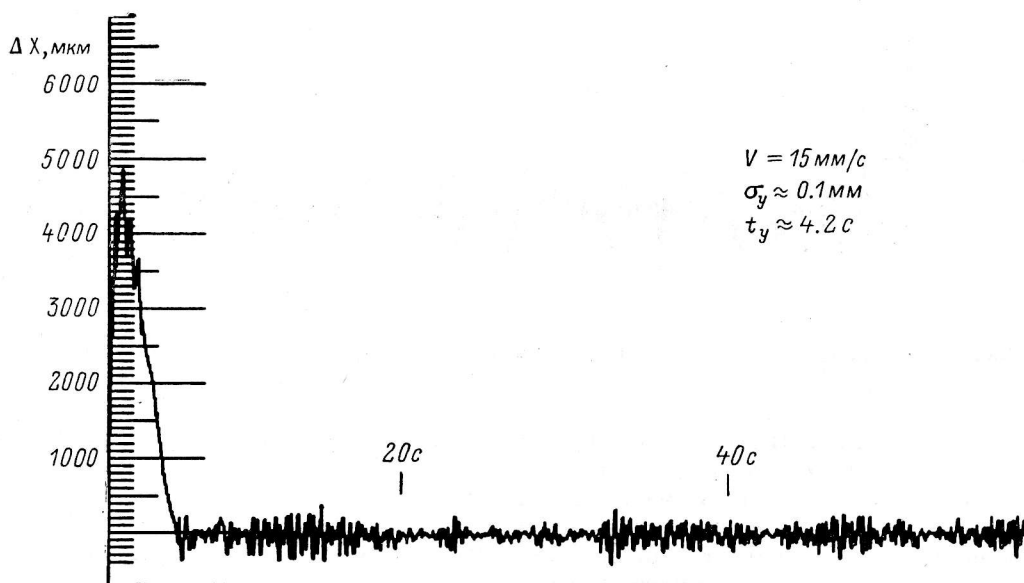


Рис. 5. То же при скорости 15 мм/с.

Рис. 3—5 получены при экспериментальной проверке качества работы системы управления кареткой на облучателе № 1. Они показывают динамику ошибки слежения в микрометрах на минутном интервале времени с начала пуска. Тестовое воздействие представляет собой скачок в изменении скорости от нуля до величин 5, 10 и 15 мм/с (соответственно рис. 3, 4, 5). В любом случае

максимальное время достижения номинальной скорости остается меньше 5 с, а среднеквадратичная ошибка в установившемся режиме — менее 0.1 мм. Проверка статического позиционирования показала точность установки выше 0.1 мм (точность измерения была ограничена точностью нониусной шкалы) при отсутствии перерегулирования.

Авторы благодарят И. В. Госачинского и Е. К. Нижельскую за ценные данные по опыту эксплуатации системы управления кареткой на облучателе № 2 и весьма признательны Ю. Н. Парийскому, М. Г. Мингалиеву и С. А. Трушкину за постоянное внимание к разработке и предоставление собственных наблюдательных программ для проведения опытных наблюдений в автоматизированном режиме.

Л и т е р а т у р а

1. М е т о д «скольжения» на радиотелескопе РАТАН-600 / М. Г. Мингалиев, З. Е. Петров, В. И. Филипенко, Л. Н. Черков // Астрофиз. исслед. (Изв. САО). 1985. 19. С. 76—81.
2. С и с т е м а регистрации и предварительной обработки данных облучателя № 1 РАТАН-600 на основе двухуровневого многомашинного комплекса / В. А. Витковский, Б. Л. Ерухимов, Г. А. Малькова, В. Н. Черненко // Сообщ. САО. 1988. 59. С. 5—17.
3. С и с т е м а управления движением каретки облучателя радиотелескопа РАТАН-600 на базе микро-ЭВМ / Г. С. Голубчин, С. Ф. Гольдшмидт, Э. С. Мучник, Е. К. Нижельская // Астрофиз. исслед. (Изв. САО). 1987. 24. С. 205—209.
4. Е р у х и м о в Б. Л., Ч е р н е н к о в В. Н. Архитектура и аппаратно-программное обеспечение сети измерительно-вычислительных комплексов на радиотелескопе РАТАН-600 // Астрофиз. исслед. (Изв. САО). 1987. 24. С. 184—190.
5. Е р у х и м о в Б. Л., Ч е р н е н к о в В. Н. Трехуровневый аппаратно-программный комплекс для автоматизации научных исследований на радиотелескопе РАТАН-600 // Микропроцессорные средства и системы. 1987. С. 86.

Поступила в редакцию
31 октября 1988 г.