ПРОГРАММИРОВАНИЕ В СРЕДЕ ТРЕЙС МОУД

♦ ИЗДАТЕЛЬСТВО ТГТУ ♦

Министерство образования и науки Российской Федерации

Государственное образовательное учреждение
высшего профессионального образования
«Тамбовский государственный технический университет»

ПРОГРАММИРОВАНИЕ В СРЕДЕ ТРЕЙС МОУД

Методические указания к выполнению лабораторных работ
по дисциплинам «ИТ проектирования ЭС», «САУ»

Тамбов
Издательство ТГТУ
2006

УДК 004.4'22
ББК ⌘973-018.3я73-5
К12

Утверждено Редакционно-издательским советом университета

Рецензент

Доктор технических наук, профессор
В.Н. Шамкин

Составитель

А.А. Кабанов

К12 Программирование в среде Трейс Моуд: Метод. указания /

Сост. А.А. Кабанов. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2006.
32 с.

Методические указания предназначены для получения навыков проектирования систем автоматического регулирования в отечественной SCADA системе ТРЕЙС МОУД, знакомства с языками отладки ТРЕЙС МОУД – Техно FBD, Техно IL и выполнения анализа качества работы исследуемых систем.

Предназначены для студентов специальности 210200 по дисциплинам: «ИТ проектирования ЭС», «САУ».

УДК 004.4'22
ББК ⌘973-018.3я73-5

Учебное издание

ПРОГРАММИРОВАНИЕ В СРЕДЕ ТРЕЙС МОУД

Методические указания

Составитель
КАБАНОВ Алексей Анатольевич

Редактор Т.М. Глинкина
Инженер по компьютерному макетированию Т.А. Сынкова

Подписано к печати 16.05.2006.
Формат 60×84/16. Бумага газетная. Печать офсетная.
Гарнитура Times New Roman. Объем: 1,86 усл. печ. л.; 1,80 уч.-изд. л.
Тираж 100 экз. С. 270

Издательско-полиграфический центр
Тамбовского государственного технического университета
392000, Тамбов, Советская, 106, к. 14

ВВЕДЕНИЕ

Современные микропроцессорные управляющие устройства позволяют реализовывать регуляторы с широкими функциональными возможностями, причем качество работы при этом напрямую зависит от правильности выбора настроек.

На сегодняшний день существует множество SCADA систем разработки АСУТП, одной из отечественных наиболее популярных в этой области является инструментальная программа ТРЕЙС МОУД. Недостаточное освещение проблемы имитационного проектирования регуляторов с визуализацией контроля делает актуальным их построение в среде ТРЕЙС МОУД. Задачи регуляторов многогранны и успешное их решение играет огромную роль в различных технологических процессах и в экономике народного хозяйства.

Пакет программ ТРЕЙС МОУД позволяет оперативно создавать электронные макеты систем управления, отлаживать их в режиме реального времени и проводить сравнительный анализ с другими видами регуляторов. Его база данных содержит необходимые компоненты для построения АСУТП. Основу инструментальной среды ТРЕЙС МОУД составляет редактор базы каналов и редактор представления данных. В редакторе базы каналов создается математическая основа системы управления: описываются конфигурации всех рабочих станций, контроллеров и устройств согласования с объектом, используемых в системе управления, настраиваются информационные потоки между ними, описываются входные и выходные сигналы и их связь с устройствами сбора данных и управления. В этом редакторе задаются периоды опроса или формирования сигналов, настраиваются законы первичной обработки и управления, технологические границы, структура математической обработки данных, а также устанавливается, какие данные и при каких условиях сохранять в различных архивах ТРЕЙС МОУД, и настраивается сетевой обмен данными. Кроме того, редактор позволяет описывать задачи управления архивами, документированием, коррекции временных характеристик системы управления (периоды опроса параметров, время цикла системы и пр.) и решать некоторые другие задачи.

Результатом работы в этом редакторе являются математическая и информационная структуры проекта АСУТП. Эти структуры включают в себя набор баз каналов и файлов конфигурации для всех контроллеров и операторских станций (узлов) проекта, а также файл конфигурации всего проекта.

В редакторе представления данных разрабатывается графическая часть проекта системы управления. При этом создается статичный рисунок технологического объекта, а затем поверх него размещаются динамические формы отображения и управления. Среди этих форм присутствуют такие, как поля вывода численных значений, графики, гистограммы, кнопки, области ввода значений и перехода к другим графическим фрагментам и т.д. Все формы отображения информации, управления и анимационные эффекты связываются с информационной структурой, разработанной в редакторе базы каналов.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

Классификация регуляторов. Регуляторы классифицируются по назначению, принципу действия, конструктивным особенностям, виду используемой энергии, характеру изменения регулирующего воздействия и т.п.

По принципу действия они подразделяются на регуляторы прямого и непрямого действия. Регуляторы прямого действия не используют внешнюю энергию для процессов управления, а используют энергию самого объекта управления (регулируемой среды). Примером таких регуляторов являются регуляторы давления. В регуляторах непрямого действия для его работы требуется внешний источник энергии.

По роду действия регуляторы делятся на непрерывные и дискретные. Дискретные регуляторы, в свою очередь, подразделяются на релейные, цифровые и импульсные.

По виду используемой энергии они подразделяются на электрические (электронные), пневматические, гидравлические, механические и комбинированные. Выбор регулятора по виду используемой энергии определяется характером объекта регулирования и особенностями автоматической системы.

По закону регулирования они делятся на двух- и трехпозиционные регуляторы, типовые регуляторы (интегральные, пропорциональны, пропорционально-дифференциальные, пропорционально-интегральные и пропорционально-интегрально-дифференциальные регуляторы – сокращенно И-, П-, ПД-, ПИ- и ПИД-регуляторы), регуляторы с переменной структурой, адаптивные (самонастраивающиеся) и оптимальные регуляторы. Двухпозиционные регуляторы нашли широкое распространение, благодаря своей простоте и малой стоимости.

По назначению регуляторы подразделяются на специализированные (например, регуляторы уровня, давления, температуры и т.д.) и универсальные с нормированными входными и выходными сигналами и пригодные для управления различными параметрами.

По виду выполняемых функций регуляторы подразделяются на регуляторы автоматической стабилизации, программные, корректирующие, регуляторы соотношения параметров и другие.

Выбор типа регулятора. Для того, чтобы выбрать тип регулятора и определить его настройки, необходимо знать:

1) статические и динамические характеристики объекта управления;
2) требования к качеству процесса регулирования;
3) показатели качества регулирования для серийных регуляторов;
4) характер возмущений, действующих на процесс регулирования.

Выбор типа регулятора обычно начинается с простейших двухпозиционных регуляторов и может заканчиваться самонастраивающимися микропроцессорными регуляторами. Заметим, что по требованиям технологического регламента многие объекты не допускают применения релейного управляющего воздействия.

Рассмотрим показатели качества серийных регуляторов. В качестве серийных предполагаются непрерывные регуляторы, реализующие И-, П-, ПИ- и ПИД-законы управления.

Теоретически, с усложнением закона регулирования качество работы системы улучшается. Известно, что на динамику регулирования наибольшее влияние оказывает величина отношения запаздывания к постоянной времени объекта τ/T. Эффективность компенсации ступенчатого возмущения регулятором достаточно точно может характеризоваться величиной динамического коэффициента регулирования, а быстродействие – величиной времени регулирования R_d.

Минимально возможное время регулирования для различных типов регуляторов при оптимальной их настройке определяется табл. 1.

Таблица 1

Закон регулирования	П	ПИ	ПИД
$t_р/\tau$	6,5	12	7

где $t_р/\tau$ – время регулирования; τ – запаздывание в объекте.

Теоретически, в системе с запаздыванием, минимальное время регулирования $t_{р\,min} = 2\tau$.

Руководствуясь табл. 1, можно утверждать, что наибольшее быстродействие обеспечивает П-закон управления. Однако, если коэффициент усиления П-регулятора $K_р$ мал (чаще всего это наблюдается в системах с запаздыванием), то такой регулятор не обеспечивает высокой точности регулирования, так как в этом случае велика величина статической ошибки. Если $K_р$ имеет величину равную 10 и более, то П-регулятор приемлем, а если $K_р < 10$, то требуется введение в закон управления интегральной составляющей.

Наиболее распространенным на практике является ПИ-регулятор, который обладает следующими достоинствами:

1 Обеспечивает нулевую статическую ошибку регулирования.

2 Достаточно прост в настройке, так как настраиваются только два параметра, а именно коэффициент усиления $K_р$ и постоянная интегрирования T_i. В таком регуляторе имеется возможность оптимизации $K_р/T_i \rightarrow$ max, что обеспечивает управление с минимально возможной среднеквадратичной ошибкой регулирования.

3 Малая чувствительность к шумам в канале измерения (в отличие от ПИД-регулятора).

Для наиболее ответственных контуров можно рекомендовать использование ПИД-регулятора, обеспечивающего наиболее высокое быстродействие в системе. Однако следует учитывать, что это условие выполняется только при его оптимальных настройках (настраиваются три параметра). С увеличением запаздывания в системе резко возрастают отрицательные фазовые сдвиги, что снижает эффект действия дифференциальной составляющей регулятора. Поэтому качество работы ПИД-регулятора для систем с большим запаздыванием становится сравнимо с качеством работы ПИ-регулятора. Кроме этого, наличие шумов в канале измерения в системе с ПИД-регулятором приводит к значительным случайным колебаниям управляющего сигнала регулятора, что увеличивает дисперсию ошибки регулирования и износ исполнительного механизма. Таким образом, ПИД-регулятор следует выбирать для систем регулирования, с относительно малым уровнем шумов и величиной запаздывания в объекте управления. Примерами таких систем являются системы регулирования температуры.

При выборе типа регулятора рекомендуется ориентироваться на величину отношения запаздывания к постоянной времени в объекте τ/T. Если $\tau/T < 0{,}2$, то можно выбрать релейный, непрерывный или цифровой регуляторы. Если $0{,}2 < \tau/T < 1$, то должен быть выбран непрерывный или цифровой, ПИ- или ПИД-регулятор. Если $\tau/T > 1$, то выбирают специальный цифровой регулятор с упредителем, который компенсирует запаздывание в контуре управления. Однако этот же регулятор рекомендуется применять и при меньших отношениях τ/T.

Алгоритмы ПИ и ПИД-регулирования. Наиболее распространенными алгоритмами цифрового регулирования являются ПИ и ПИД. При правильной настройке эти алгоритмы обеспечивают достаточно хорошее качество управления для большинства объектов промышленной технологии.

Рассмотрим алгоритм ПИД-регулятора (2) для объекта первого порядка, модель динамики которого в дискретной форме имеет вид:

$$y(t_{n+1}) = \alpha\, y(t_n) + \beta u(t_n)\,, \quad t_{n+1} = t_n + \delta t\,, \quad n = 0, 1, 2, ...;$$

$$\alpha = e^{a\delta t}\,, \quad \beta = -\frac{b}{a}(1 - e^{a\delta t})\,, \qquad (1)$$

где a, b – параметры модели объекта; δt – шаг дискретизации; $y(t_n)$, $u(t_n)$ – текущие значения фазовой координаты и управления на n-м шаге.

$$u_{р}(t_n) = K_р \Delta y(t_n) + \frac{K_р}{T_и} \sum_{t_i = t_0}^{t_i = t_n} \Delta y(t_i) + \frac{K_р}{T_d} \sum_{t_i = t_0}^{t_i = t_n} \Delta y(t_i)\,, \qquad (2)$$

где $K_р$ – коэффициент усиления регулятора; $T_и$ – постоянная интегрирования; T_d – постоянная дифференцирования; $\Delta y(t_n)$ – текущее рассогласование выходной величины объекта и заданного значения на n-м шаге.

Три параметра: $K_р$, $T_и$, T_d подбирают в процессе настройки регулятора таким образом, чтобы максимально приблизить алгоритм функционирования системы к желаемому виду (рекомендации приведены в прил. Б).

В зависимости от типа объекта управления может быть достаточным применение более простого П-регулятора:

$$u_р(t_n) = K_р \Delta y(t_n) \tag{3}$$

или ПИ-регулятора:

$$u_р(t_n) = K_р \Delta y(t_n) + \frac{K_р}{T_и} \sum_{t_i=t_0}^{t_i=t_n} \Delta y(t_i), \tag{4}$$

которые являются частными случаями ПИД-регулятора при соответствующем выборе постоянных интегрирования и дифференцирования.

Энергозатраты будут определяться по формулам:

$$I_э = \int_{t_0}^{t_к} u^2(t)dt \quad \text{или} \quad I_э = \sum_{t_i=t_0}^{t_i=t_к} u_i^2(t_i)\delta t, \tag{5}$$

ГДЕ ПЕРВОЕ ВЫРАЖЕНИЕ – ФУНКЦИОНАЛ В НЕПРЕРЫВНОЙ ФОРМЕ, А ВТОРОЕ ВЫРАЖЕНИЕ – В ДИСКРЕТНОЙ.

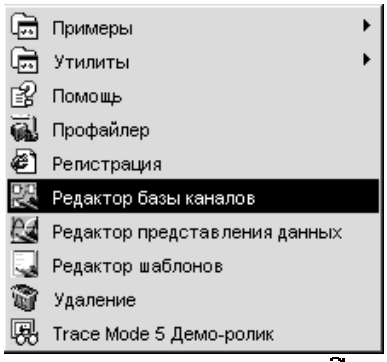
2 ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТ

При подготовке к работе рекомендуется ознакомиться с прил. А.

Подготовка к работе заключается в активации рабочего файла, специально сконфигурированного для выполнения лабораторных работ.

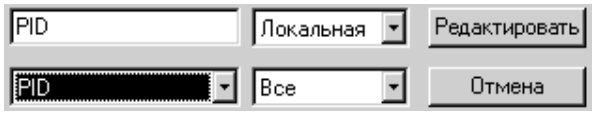
Последовательность действий при этом следующая:

1) из меню «Пуск» компьютера необходимо запустить редактор базы каналов ТРЕЙС МОУД (РБК);



2) при помощи функциональной клавиши РБК «Загрузить» необходимо открыть файл «lab.ctm»;

3) при помощи клавиши РБК «Программы» выбрать FBD программу PID, подтвердив выбор нажатием клавиши «Редактировать».



Описание назначения клавиш меню FBD

1) Переход в режим редактирования связей позволяет настроить структуру обработки данных в программе.

2) Переход в режим размещения новых блоков, любое нажатие левой кнопки мыши на свободном месте рабочего поля окна размещает новый функциональный блок.

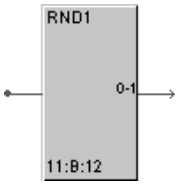
3) Эмуляция в пошаговом режиме.

4) Эмуляция в непрерывном режиме. 

Описание диаграммы блоков

На экспериментальном поле Вы видите набор функциональных блоков связанных между собой определенным образом. Опишем их предназначение:

1) ГЕНЕРАТОР СЛУЧАЙНЫХ ЧИСЕЛ – ПЕРВАЯ СТУПЕНЬ ЭМУЛЯЦИИ ДИНАМИЧЕСКИХ ВНЕШНИХ ВОЗДЕЙСТВИЙ (ОТ 0 ДО 1).



2) Расчетный модуль – вторая ступень эмуляции динамических внешних воздействий:

PID (In) – значение на выходе ПИД-регулятора;

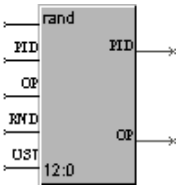
OP (In) – значение на выходе оптимального регулятора (OP) (не используется);

RND (In) – выходное значение блока RND1;

UST(In)* – установка границ случайных воздействий;

PID (Out) – синтезированное входное значение для ПИД-регулятора с учетом внешних воздействий;

OP (Out) – синтезированное входное значение для OP с учетом внешних воздействий (не используется).

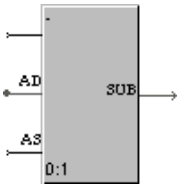


3) Элемент сравнения, применяется для расчета рассогласования в ПИД-регуляторе:

AD (In)* – требуемое значение выхода ПИД-регулятора;

AS (In) – текущее значение выхода ПИД-регулятора;

SUB (Out) – значение рассогласования.

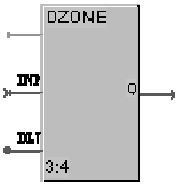


4) Определитель зоны нечувствительности в ПИД-регуляторе:

INP (In) – контролируемый сигнал;

DLT (In) – значение зоны нечувствительности;

Q (Out) – значение рассогласования.



5) МОДУЛЬ ПИД-РЕГУЛЯТОРА:

INP (In) – значение входа на текущем такте пересчета;

KP (In)* – коэффициент при пропорциональной составляющей;

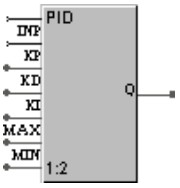
KD (In)* – коэффициент при дифференциальной составляющей;

KI (In)* – коэффициент при интегральной составляющей.

MAX (In)* – верхняя граница регулирования;

MIN (In)* – нижняя граница регулирования;

Q (OUT) – ЗНАЧЕНИЕ ВЫХОДА НА ТЕКУЩЕМ ТАКТЕ ПЕРЕСЧЕТА.



6) МОДУЛЬ ОБЪЕКТА ПЕРВОГО ПОРЯДКА:

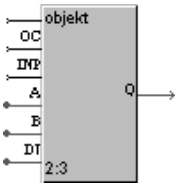
OC (IN) – ВХОД ОБРАТНОЙ СВЯЗИ ОБЪЕКТА;

INP (IN) – ВХОД ОБЪЕКТА;

A (IN)* – КОЭФФИЦИЕНТ *A* МОДЕЛИ ОБЪЕКТА;

B (IN)* – КОЭФФИЦИЕНТ *B* МОДЕЛИ ОБЪЕКТА;

DT (IN)* – ШАГ ДИСКРЕТИЗАЦИИ;



Q (OUT) – ВЫХОД ОБЪЕКТА.

7) МОДУЛЬ ПОДСЧЕТА ФУНКЦИОНАЛА:

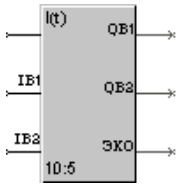
IB1 (IN) – ВХОД ЭНЕРГОЗАТРАТ ПИД-РЕГУЛЯТОРА;

IB2 (IN) – ВХОД ЭНЕРГОЗАТРАТ ОР (НЕ ИСПОЛЬЗУЕТСЯ);

QB1 (OUT) – ЗНАЧЕНИЕ СУММАРНЫХ ЭНЕРГОЗАТРАТ ДЛЯ ПИД-РЕГУЛЯТОРА;

QB2 (OUT) – ЗНАЧЕНИЕ СУММАРНЫХ ЭНЕРГОЗАТРАТ ДЛЯ ОР (НЕ ИСПОЛЬЗУЕТСЯ);

ЭКО (OUT) – ЭКОНОМИЧЕСКИЙ ЭФФЕКТ В ПРОЦЕНТАХ ОТ ИСПОЛЬЗОВАНИЯ ОР В ОТЛИЧИЕ ОТ ПИД-РЕГУЛЯТОРА.



8) МОДУЛЬ ОТОБРАЖЕНИЯ:

IN1 – ОТОБРАЖЕНИЕ ТЕКУЩЕГО ЗНАЧЕНИЯ ФАЗОВОЙ КООРДИНАТЫ ВЫХОДА ОБЪЕКТА (ЦВЕТ ЧЕРНЫЙ);

IN2 – ОТОБРАЖЕНИЕ ТЕКУЩЕГО ЗНАЧЕНИЯ ФАЗОВОЙ КООРДИНАТЫ ВЫХОДА ОБЪЕКТА С НАЛОЖЕНИЕМ ВНЕШНИХ ВОЗМУЩЕНИЙ (ЦВЕТ СИНИЙ);

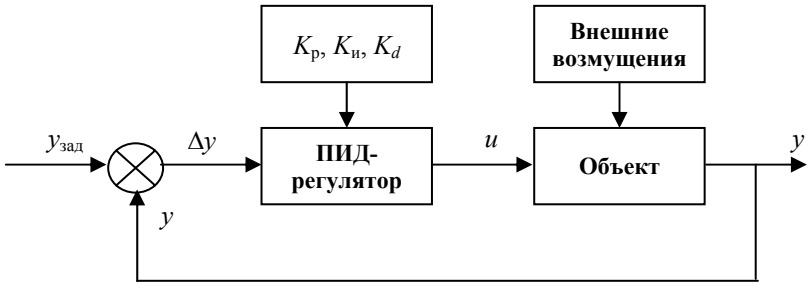
IN3 – ОТОБРАЖЕНИЕ ТЕКУЩЕГО ЗНАЧЕНИЯ УПРАВЛЕНИЯ (ЦВЕТ ЗЕЛЕНЫЙ);

MIN – ГРАНИЦА ОТРАЖЕНИЯ МИНИМУМА;

MAX – ГРАНИЦА ОТРАЖЕНИЯ МАКСИМУМА.

* – разрешается изменять.

Линейка блоков «RND1», «rand», «–», «DZONE», «PID», «objekt», «Sfloat» реализует следующее ПИД-управление:



Редактирование диаграмм блоков является очень важным при реализации определенных алгоритмов каких-либо технологических процессов. Данный вопрос включает в себя следующие пункты: добавление, удаление блоков, язык Техно FBD, редактирование связей, создание новых блоков, язык Техно IL. Рассматриваемая диаграмма реализует законченный процесс и не требует редактирования, поэтому вопрос «Редактирование диаграмм блоков» изложен как дополнительный материал в прил. А.

Лабораторная работа 1

ПРОЕКТИРОВАНИЕ СИСТЕМ РЕГУЛИРОВАНИЯ С ПОМОЩЬЮ ЯЗЫКА ТЕХНО FBD

Цель работы: ознакомление с интерфейсом, с назначениями функциональных блоков, пошаговая эмуляция работы регулятора и получение навыков редактирования связей диаграммы блоков с помощью языка Техно FBD.

Методические указания

Перед выполнением данной работы необходимо на входах модулей диаграммы блоков установить значения, приведенные в табл. 2. Установка параметров выполняется двойным нажатием левой кнопки мыши на соответствующем входе и вводе с клавиатуры требуемых значений.

2 Начальные установки для блоков

Название блока (вход)	Значение	Назначение входа
rand (UST)	0	Значение внешних воздействий

– (AD)	150	Требуемое значение регулирования
DZONE (DLT)	0,1	Зона нечувствительности
PID (KP)	15	Коэффициент при пропорциональной составляющей
PID (KD)	0	Коэффициент при дифференциальной составляющей
PID (KI)	0,9	Коэффициент при интегральной составляющей
PID (MAX)	120	Максимальное значение управления
PID (MIN)	0	Минимальное значение управления
objekt (A)	–0,00037	Параметр *a* модели объекта
objekt (B)	0,00087	Параметр *b* модели объекта
objekt (DT)	60	Шаг дискретизации
Sfloat (MIN)	–20	Минимальная граница отображения
Sfloat (MAX)	200	Максимальная граница отображения

При помощи клавиши «Эмуляция в пошаговом режиме» необходимо активировать диаграмму и нажатием левой кнопки мыши на свободном поле можно выполнять пошаговую работу данной системы. Рекомендуемое число шагов – до стабилизации режима регулятора. В режиме эмуляции можно отслеживать все параметры любого из приведенных в диаграмме блоков, они отражаются на входах/выходах блоков. Более подробную информацию по текущему значению параметров можно получить, наведя на него курсор мыши. В этом случае параметр будет отражен в строке статуса (нижний сервис-бар). На отражающем элементе можно наблюдать переходные процессы регулирования: зеленым цветом определено управление, черным – текущего значения фазовой координаты выхода объекта, синим – текущего значения фазовой координаты выхода объекта с наложением внешних возмущений.

Порядок выполнения

1 Внимательно ознакомьтесь с методическими указаниями и прил. А («Добавление, удаление блоков при помощи языка Техно FBD», «Редактирование связей»).

2 Постройте в редакторе базы каналов ТРЕЙС МОУД диаграмму блоков, реализующую ПИД-регулирование.

3 Выполните пошаговую эмуляцию работы регулятора с регистрацией всех выходных данных.

4 Составьте таблицу результатов на выходах блоков при пошаговой эмуляции.

Форма отчета

Результат выполнения заданий, проверенных и подписанных преподавателем.

Контрольные вопросы

1 Приведите общее описание типового ПИД-регулятора (структурная схема, дифференциальное уравнение или передаточная функция, основные параметры).

2 Перечислите модули и их функции при построении диаграммы ПИД-регулятора.

3 Перечислите основные функции редактора программ Техно FBD.

Лабораторная работа 2

ИМИТАЦИОННОЕ ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ РЕЖИМОВ ПИД-РЕГУЛЯТОРА В СРЕДЕ ТРЕЙС МОУД

Цель работы: получение точностных и энергетических характеристик работы регуляторов в среде ТРЕЙС МОУД при разных значениях параметров модели объекта и внешнего дестабилизирующего фактора.

Методические указания

Необходимо установить на входах функциональных блоков (табл. 3) значения параметров модели, полученные в результате обработки экспериментальных данных (рекомендации по выбору параметров ПИД-регулятора приведены в прил. Б). Остальные параметры (лабораторная работа 1, табл. 2) можно также изменять в зависимости от условий задачи.

Таблица 3

Название параметра	Значение
Требуемое значение регулирования ($y_{зад}$, ед.)	
Параметр *a* модели объекта	
Параметр *b* модели объекта	
K_p ПИД-регулятора	

K_d ПИД регулятора	
$K_и$ ПИД регулятора	
Шаг дискретизации (δt, с)	
Амплитуда внешних факторов (*UST*, ед.)	
Время выхода на заданный режим (t, с)	
Энергозатраты ($I_э$, кВт/ч)	

Число шагов задается индивидуально.

Порядок выполнения

1 Внимательно ознакомьтесь с методическими указаниями и прил. А.

2 Основываясь на методах прил. Б, подберите параметры ПИД-регулятора.

3 Используя пошаговую или непрерывную эмуляцию работы регуляторов, изучите переходные процессы регулирования для разных исходных данных и заполните табл. 3 для каждого эксперимента (число экспериментов 3 – 5).

4 Постройте графики переходных процессов.

Форма отчета

Результат выполнения заданий, проверенных и подписанных преподавателем.

Контрольные вопросы

1 Какие преимущества имеет регулятор при оптимальных настройках в сравнении с регулятором с отклоненными параметрами?

2 Какое влияние оказывает на имитационную модель блок DZONE?

3 Каким образом в диаграмме блоков учитывается тип регулятора?

Литература [1 – 5].

Л а б о р а т о р н а я р а б о т а 3

ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ МОДУЛЕЙ НА ЯЗЫКЕ ТЕХНО IL

Цель работы: изучение языка Техно IL в объеме достаточном для составления программ, описывающих математические функции, получение навыков создания функциональных блоков.

Методические указания

Базовое описание языка Техно IL и основы создания программ приведены в прил. А. Для примера приведем **текст программы** Техно IL с комментариями для функционального блока, реализующего функцию апериодического звена:

$$z(t_{n+1}) = \alpha z(t_n) + \beta u(t_n), \qquad t_{n+1} = t_n + \delta t, \qquad n = 0, 1, 2, ...;$$

$$\alpha = e^{a\delta t}, \qquad \beta = -\frac{b}{a}(1 - e^{a\delta t}).$$

```
/* I0 – текущая координата */
/* I1 – управление */
/* I2 – a*/
/* I3 – b*/
/* I4 – шаг дискреты */
#define I0 OC
#define I1 INP
#define I2 A
#define I3 B
#define I4 DT
#define Q0 Q
/* A * Yo /
F0=I2
F0*I4
EXP F0
```

Комментарии, гласящие, что входы функционального блока I1…I4 содержат численную информацию соответственно о: текущей координате, управлении, коэффициентах *a* и *b* модели апериодического звена и шаге дискретизации.

Буквенные обозначения названий входов и выходов на функциональном блоке. Например, выход Q0 будет обозначен «Q».

Комментарий, гласящий, что ниже приведенный текст программы реализует $\alpha z(t_n)$.

Рабочей переменной F0 присваивается значение входа I2 (параметр *a* модели)

Переменная F0 умножается на значение входа I4 (дискрета δt) и результат присваивается F0.

e^{F0}, т.е. $e^{a\delta t}$.

F2=F0	Сохранение значения F0 в F2 для дальнейшего использования.
F0*I0	$e^{a\delta t} z(t_n)$.
/* B*Uo */	Комментарий, гласящий, что ниже приведенный текст программы реализует $\beta u(t_n)$.
F1=I3	Рабочей переменной F1 присваивается значение входа I3 (параметр *b* модели).
F1/I2	Переменная F1 делится на значение входа I2 (параметр *a* модели) и результат присваивается F1.
F2 1	$-(1 - e^{a\delta t})$.
F1*F2	$-b/a(1 - e^{a\delta t})$.
F1*I1	
/* A * Yo + B*Uo*/	
Q0=F0	Выходу присваивается значение $\alpha z(t_n)$.
Q0+F1	Выходу присваивается значение полной функции.
W3=Q0	Запись выхода в глобальную переменную для дальнейшего использования (например на последующих тактах пересчета, в программах других функциональных блоков, для записи в файл «tracemode5\ASUTP\W3.txt»).

Варианты:

1 $z(t_{n+1}) = x^3 z(t_n) + \delta t$.

2 $z(t_{n+1}) = \sin x^2 z(t_n) + \delta t$.

3 $z(t_{n+1}) = \cos x^2 z(t_n) + \delta t$.

4 $z(t_{n+1}) = \dfrac{x+5}{\cos x} z(t_n) + \delta t$.

5 $z(t_{n+1}) = (x+5)^2 z(t_n) + \delta t$.

6 $z(t_{n+1}) = \dfrac{x+10}{\sin x} z(t_n) + \delta t$.

7 $z(t_{n+1}) = e^{\cos x} z(t_n) + \delta t$.

8 $z(t_{n+1}) = e^{(x+5)} z(t_n) + \delta t$.

9 $z(t_{n+1}) = e^{x^2} z(t_n) + \delta t$.

Значение $x = 3$; $\delta t = 1$.

Порядок выполнения

1 Ознакомьтесь с примером создания программы и прил. А («Описание языка Техно IL», «Создание новых блоков, язык Техно IL»).

2 Напишите программу Техно IL, которая описывает математическую функцию вашего варианта и сохраните текст программы в файл.

3 Выполните трансляцию программы, создайте функциональный блок.

4 В редакторе базы каналов постройте систему управления с вашим функциональным блоком.

5 Выполните пошаговую эмуляцию и запишите в виде таблицы значения выхода блока для 5 шагов.

Форма отчета

Результат выполнения заданий, проверенных и подписанных преподавателем.

Контрольные вопросы

1 Каковы отличия между статическими и динамическими переменными, специфика глобальных переменных?

2 Перечислите типы программ Техно IL и их особенности.

3 Перечислите действия при добавлении в систему блока с новой функцией.

Литература [1].

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ TRACE MODE // ADAASTRA RESEARCH GROUP, LTD. 1998. 770 С.

2 СПРАВОЧНИК ПО ТЕОРИИ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ / ПОД РЕД. А.А. КРАСОВСКОГО. М.: НАУКА, 1987. 712 С.

3 Дворецкий С.И., Лазарева Т.Я. Проектирование автоматизированных систем управления химико-технологическими процессами: Учеб. пособие. Тамбов: ТГТУ, 1993. 206 с.

4 Микропроцессорное управление технологическим оборудованием микроэлектроники: Учеб. пособие. / А.А.Сазонов, Р.В. Корнилов, Н.П. Кохан и др.; Под ред. А.А.Сазонова. М.: Радио и связь, 1988. 264 с.

5 Бесекерский В.А., Попов Е.П. Теория систем автоматического регулирования. М.: Наука, 1975.

П р и л о ж е н и е А

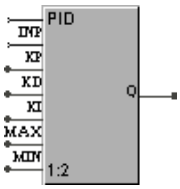
ДОБАВЛЕНИЕ, УДАЛЕНИЕ БЛОКОВ ПРИ ПОМОЩИ ЯЗЫКА ТЕХНО FBD

Для размещения блоков на поле окна используется визуальный язык программирования алгоритмов Техно FBD. Программа, созданная в этом языке, называется FBD-программой.

В рабочем поле окна редактирования FBD-программ выводится диаграмма из функциональных блоков, реализующая программируемые алгоритмы, список переменных программы и диалог управления редактированием.

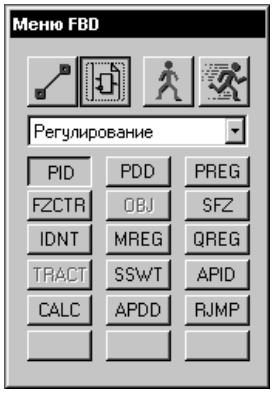
Элементарным звеном разработки программ на языке Техно FBD является функциональный блок.

Блок – это графическое изображение вызова одной из функций, реализованных в языке. Это могут быть либо стандартные, встроенные в систему функции либо разработанные пользователем на языке Техно IL. Пример блока ПИД-регулятора приведен ниже:



Каждый функциональный блок имеет следующие атрибуты: выполняемая функция, номер, входы и выходы. Для данного блока это: функция – PID регулирование, номер на диаграмме – 1 : 2, его входы – INP, KP, KD, KI, MAX, MIN, его выход – Q.

Функция блока выбирается перед размещением блока. Выбор функции блока осуществляется в диалоге Меню FBD, показанном на следующем рисунке.



Здесь сначала указывается функциональный раздел (в данном случае «Регулирование»), а затем в нем выбирается требуемая функция (в данном случае PID).

Номер функционального блока устанавливается системой автоматически при его размещении в рабочем поле. Этот номер недоступен для изменения. Он используется только для индикации очередности выполнения функциональных блоков при вызове программы.

Входы в функциональный блок на его изображении всегда расположены слева, а выходы – справа. При вычислении блока над переменными, связанными с его входами, осуществляются действия, определенные функцией блока. Полученные в результате значения присваиваются переменным, связанным с выходами блока.

Вход блока может быть связан с переменной или константой, а может быть свободным. Последние не учитываются при его пересчете. На диаграмме они выводятся ярко-зеленым цветом.

Назначение всех входов и выходов определяется функцией блока. Исключением является только первый вход. Он для всех блоков управляет их пересчетом. Далее все входы, используемые для формирования значений выходов, будут называться функциональными, а вход, управляющий пересчетом, – входом блокировки. Все функциональные входы и выходы блоков имеют обозначения, включающие в себя до трех символов.

При отладке программы в режиме эмуляции рядом с каждым входом и выходом функционального блока выводится его текущее значение. При этом для входов выводимые значения заменяют их обозначения.

Инверсия может быть применена к любым входам блоков, реализующих логические операции: & (логически умножить), I (логически сложить), ^ (выполнить исключающее логическое сложение), и к битовым входам блоков триггеров: RS, SR, TOFF, TON, TP.

Инверсия на выходах осуществляется для всех функциональных блоков без исключения, но результат при этом превращается в логическую 1 или 0.

Для выполнения инверсии надо в режиме редактирования нажать ПК на изображении входа или выхода. При этом в его основании появится пустой круг – признак инвертирования.

На одном функциональном блоке инвертироваться может только один функциональный вход.

Существуют два вида представления данных для входов и выходов функциональных блоков:

- FLOAT;
- HEX.

Первый из них предназначен для работы с аналоговыми переменными, представленными в виде числа с плавающей точкой одинарной точности. Второй – для работы с целыми 16-битовыми переменными. При связывании входов и выходов с разными видами представления осуществляется преобразование данных из одного вида в другой.

Типы входа и выхода функционального блока могут иметь следующие форматы:

- свободен;
- блок;
- константа;
- аргумент.

Тип входа/выхода определяет переменную, с которой он будет связан. Это может быть либо внутренняя переменная программы, либо константа, либо внешняя переменная, передаваемая в качестве аргумента при вызове программы.

Тип «Свободен» устанавливается автоматически для всех незадействованных входов функциональных блоков. Этот тип недоступен для установки вручную.

Тип «Блок» устанавливается для всех выходов функциональных блоков при их размещении. Вход или выход, имеющий данный тип, связан с внутренней переменной программы. Этот тип автоматически устанавливается для входа, связанного с выходом любого блока (в частном случае – с выходом того же блока).

«Константа» может быть задана только для входа функционального блока. Такой вход имеет фиксированное значение и не может меняться в процессе выполнения программы. Величина входа типа «Константа» задается в специальном поле диалога «Описание переменной».

Вход или выход типа «Аргумент» при вызове программы связывается с внешними переменными. Такими переменными являются атрибуты каналов. При настройке аргументов можно также задать значения констант. Константы не связываются с атрибутами каналов. Им просто устанавливается значение, которое они должны принять при конкретном вызове FBD-программы

Установка на поле чертежа выбранного блока осуществляется по-средством перехода в режим размещения новых блоков клавишей меню FBD и нажатием левой кнопки мыши на поле чертежа, где предусматривается установка данного блока.

Удаление функционального блока с рабочего поля: переход в режим редактирования связей клавишей  меню FBD, выделение нажатием правой клавиши мыши удаляемого блока и нажатие клавиши «Delete» на клавиатуре компьютера.

Чтобы создать новую FBD-программу, надо: нажать кнопку  в верхнем меню, выполнить команду «Создать» из меню Программа диалога FBD-программа. При этом в список данного диалога будет добавлена новая программа. Ее имя формируется следующим образом:

Form <N>, где <N> – номер программы в списке.

После создания программы ее имя может быть отредактировано в соответствующем поле диалога FBD-программа.

Для удаления программы надо сначала указать ее имя в списке диалога FBD-программа. Затем следует выполнить команду «Удалить» из меню Редактирование этого же диалога.

Тип FBD-программ. Существуют два типа программ: локальные и глобальные. Первые создаются и могут быть использованы только на одном узле. Вторые после их создания тиражируются на все узлы проекта. После этого глобальная программа становится локальной. Ее дальнейшее редактирование на любом узле не приводит к тиражированию внесенных изменений.

Установка типа программы осуществляется выбором из списка в соответствующем поле диалога FBD-программа.

По умолчанию создаваемая программа имеет локальный тип. При создании локальной FBD-программы из окна структуры проекта она будет доступна в текущем выделенном узле.

Сохранение и вставка FBD-программ. FBD-программу можно сохранить в файл и затем вставить в другой узел того же или другого проекта. Ее можно сохранить также в виде текста на структурированном языке международного стандарта МЭК 1131-3. Это позволяет использовать данную программу в других приложениях.

Сохранение. Для сохранения FBD-программы в файл надо ее выбрать в списке диалога FBD-программа. Затем следует выполнить команду. Сохранить из меню Программа данного диалога. При этом на экране появляется стандартный диалог выбора файла. Указанная FBD-программа будет сохранена в заданный здесь файл, которому автоматически дается расширение «crm».

Сохранение в виде текста. Для сохранения FBD-программы в текстовом виде надо выполнить команду «Сохранить как текст» меню Программа. Создаваемый файл в этом случае будет иметь текстовый формат и расширение «frm».

Вставка из файла. Любую сохраненную в файл FBD-программу можно вставить в проект. Для этого, находясь в диалоге FBD-программа, надо выполнить команду «Открыть» из меню Программа. В результате на экран будет выведен диалог выбора файла. После указания требуемого файла записанная в нем программа будет добавлена в текущий проект и ее имя появится в списке FBD-программ.

Переход к редактированию FBD-программ. Для перехода к редактированию выбранной FBD-программы следует выполнить команду «Редактировать» из меню Редактирование диалога FBD-программа или нажать левую кнопку мыши на соответствующей кнопке того же диалога. При этом в рабочее поле окна редактирования FBD-программ выводится диаграмма из функциональных блоков, реализующая алгоритм указанной программы.

РЕДАКТИРОВАНИЕ СВЯЗЕЙ

Режим редактирования связей позволяет настроить структуру обработки данных в программе. В этом режиме доступны следующие операции: настройка входов и выходов функциональных блоков; связывание входов и выходов функциональных блоков и удаление связей; удаление блоков; перемещение блоков.

Для настройки входа или выхода функционального блока надо дважды нажать на нем левой клавишей мыши. При этом на экране появится диалог Описание переменной. Этот диалог позволяет задать тип переменной, вид представления и значение. На приведенном ниже рисунке показан вид этого диалога.

Кроме того, в этом диалоге можно ввести комментарий. Этот комментарий выводится в рамке рядом с настраиваемым входом или выходом, так же в бланках диалога «Реквизиты» для переменных типа «Аргумент» и «Константа» при настройке вызова программы.

Связывание входов и выходов функциональных блоков. При разработке алгоритмов, выполняющих несколько последовательных действий, необходимо передать результаты вычислений одного блока другому. Для этого надо связать соответствующие входы и выходы этих блоков.

Для установления такой связи следует нажать левой кнопкой мыши на одном из ее концов и, удерживая кнопку мыши нажатой, перевести курсор в область другого конца связи. При этом соответствующие вход и выход функциональных блоков соединятся линией. Связанным входам и выходам автоматически присваивается тип «Блок».

Связывать можно не только вход с выходом, но и вход со входом. В этом случае линии соединения не проводятся, однако связь устанавливается. При выделении любого из связанных между собой входов цвет всех остальных также меняется на красный.

Чтобы посмотреть все связи выхода функционального блока, следует нажать на нем левой кнопкой мыши. При этом линии связи этого выхода с другими блоками выделяются красным цветом. Нажатие левой кнопкой мыши на входе блока не выделяет его связи. Однако при этом выделяются все входы других блоков, связанных с тем же источником.

Для удаления связи между блоками надо нажать левую кнопку мыши на входе в блок. Нажатие клавиши «DEL» после этого приводит к удалению выделенной связи. После удаления связи входам блоков присваивается тип «Свободен», а выходам – «Блок».

Чтобы удалить функциональный блок из редактируемой программы, его надо сначала выделить. Выделение блока осуществляется нажатием левой кнопкой мыши на его изображении. Нажатие клавиши «DEL» после этого приводит к удалению из программы выделенного блока. Все связи этого блока также удаляются. Входам блоков, связанных с выходами удаляемого блока, присваивается тип «Свободен», а выходам – «Блок».

После удаления блока освобождается номер, под которым он был вставлен в программу, а связанные с ним переменные приобретают статус «Свободен». При добавлении в программу новых блоков им сначала будут присваиваться номера, освободившиеся при удалении, а для их входов и выходов будут использоваться переменные со статусом «Свободен».

Перемещение функциональных блоков. Чтобы сделать программу читаемой и менять последовательность пересчета блоков, предусмотрена возможность перемещения блоков с сохранением всех определенных для них связей.

Для перемещения блока надо сначала выделить его нажатием на нем левой кнопкой мыши. Затем, удерживая нажатой левую кнопку мыши, следует перевести курсор мыши в требуемую область рабочего поля. После того как левая кнопка мыши будет отпущена, выделенный блок переместится в указанное место.

Для перемещения группы блоков необходимо выделить их, при нажатой кнопке «Ctrl», с помощью левой кнопкой мыши. Затем, удерживая нажатой левую кнопку мыши на любом из выделенных блоков, следует перевести курсор мыши в требуемую область рабочего поля. После того как левая кнопка мыши будет отпущена, выделенные блоки переместятся в указанное место. Для снятия выделения с группы блоков можно провести ту же последовательность действий, воспользоваться командой меню «Правка» – «Снять маркировку», или сочетанием клавиш «Ctrl + U». При наложении функциональных блоков и выходе за пределы рабочего окна перемещение не осуществляется.

ОПИСАНИЕ ЯЗЫКА ТЕХНО IL

Текст программы, разработанной на Техно IL, представляет собой последовательность инструкций. Каждая инструкция включает в себя описатель действий и операнды. Максимальное количество инструкций одной программы равно 12000. Описателями действий могут быть функции, операторы и операции. В качестве операндов могут использоваться переменные и константы.

Переменные языка Техно IL. Язык Техно IL позволяет использовать несколько типов переменных. Для всех типов переменных структура их имен остается одинаковой: первый символ в имени определяет тип переменной, далее за символом типа без пробела следует ее номер. Для различных типов переменных Техно IL определены следующие идентификаторы:

- I – входные переменные;
- Q – выходные переменные;

- E – статические переменные;
- W – статические глобальные переменные;
- F – динамические переменные.

Количество доступных в программе переменных разного типа определяется типом IL-программы.

Кроме описанных типов, в Техно IL используются еще две системные переменные:

- result – аккумулятор (X);
- CMP – признак истинности.

Константы в IL-программах. В тексте программ можно использовать константы. Константы могут быть двух типов:

- целые числа в диапазоне от –127 до 127;
- любые другие числа.

Эти константы могут записываться как в десятеричном, так и в шестнадцатеричном виде. Шестнадцатеричная константа должна начинаться с 0*x*.

Ограничений на число констант первого типа не накладывается, поэтому их в программе может быть любое количество.

Количество доступных для использования констант второго типа зависит от типа программы. Для «FB» их можно использовать не более 10, а для «PRG» – не более 40. При этом совпадающие по значению константы считаются за одну. Например, если в тексте программы три раза применяется константа 10,5, то считается, что использована только одна константа.

Операнды языка Техно IL. Каждая инструкция программы содержит оператор и операнды. Оператор задает действие, которое надо осуществить с операндами. В качестве операндов в IL-программе могут использоваться все описанные выше переменные, кроме CMP, а также константы.

Синтаксис записи операций. Язык инструкций предполагает следующую структуру записи операций. В одной строке записывается сначала мнемоническое обозначение операции, затем после символа разделителя (например, пробел) записывается операнд. Техно IL позволяет использовать два операнда и мнемоническое или символьное обозначение операции. Кроме того, обозначение операции может быть помещено между операндами.

Например, операция сложения переменной Q1 с переменной I2 с записью результата в первую из них в соответствии со стандартом записывается следующим образом:

LD Q1
ADD I2
ST Q1

В Техно IL эта операция может быть записана так же, а может – существенно компактнее:

ADD Q1 I2 ИЛИ **+ Q1 I2** ИЛИ **Q1 + I2**.

В одной строке программы можно записать несколько операций. В этом случае их следует разделять символом «;». Полный список функций, меток, операторов, комментариев языка Техно IL приведен в электронной справке ТРЕЙС МОУД.

СОЗДАНИЕ НОВЫХ БЛОКОВ, ЯЗЫК ТЕХНО IL

Язык инструкций (Техно IL) – это текстовый язык ТРЕЙС МОУД для разработки программ, реализующих функции обработки данных и управления. Он является расширением IL-языка международного стандарта IEC 1131-3. Это расширение позволяет использовать более простой и интуитивный синтаксис, дополнительные функции и операторы, а также двухадресный режим.

Вызов IL-программ. Программы, разработанные на Техно IL, могут вызываться двумя способами. Первый способ – из FBD-программ. В этом случае Техно IL используется для программирования функций блоков, добавляемых в систему. Второй способ вызова IL-программ – это запуск их параллельно с пересчетом базы каналов.

Функциональные блоки. Язык Техно FBD имеет около 150 стандартных типов функциональных блоков. Кроме того, он поддерживает использование произвольно программируемых функциональных блоков. Их максимальное количество, которое можно одновременно включить в систему, равно 54. Программирование таких блоков осуществляется на языке Техно IL.

Чтобы добавить в систему функциональный блок с новой функцией, надо в диалоге Техно IL разработать программу, реализующую требуемую функцию и подключить ее к системе. При этом тип программы надо установить «FB».

Вызов IL-программы, реализующей функцию данного блока, будет осуществляться при каждом вычислении FBD-программ, где он используется.

Тип IL-программы определяет способ ее вызова монитором реального времени, а также ограничения, которые накладываются на количество доступных переменных и операторов. Он задается выбором из списка в соответствующем поле диалога Техно IL и может принимать следующие значения:

- FB – программирование блока для Техно FBD;
- PRG – метапрограмма.

Первый из них используется при программировании собственных функциональных блоков для языка Техно FBD. Тип «PGR» устанавливается для метапрограмм, запускаемых параллельно с пересчетом базы каналов.

Внимание! Тексты программ типа «PGR» должны начинаться с ключевого слова «PROGRAM».

Имена IL-программы. Для каждой IL-программы определены два имени: основное и дополнительное. Они вводятся в соответствующих полях показанного выше диалога Техно IL. Текст обоих имен IL-программы может включать в себя 7 любых символов.

Первое из них используется для идентификации программы. Оно выводится во всех списках редактора базы каналов при ссылках на IL-программы.

Назначение дополнительного имени зависит от типа программы. Для типа программы FB оно задает название функции программируемого блока. Это название выводится в верхней части функционального блока при его размещении в рабочем поле редактирования FBD-программ. Если IL-программа имеет тип «PRG», то дополнительное имя используется для задания его номера. Номер программы может быть установлен от 0 до 15.

Эти атрибуты используются для идентификации программы, определения доступа к ней и способа вызова. Они задаются в Диалоге Техно IL. Вход в этот диалог осуществляется по команде «Создать» меню Техно IL.

Диалог Техно IL позволяет выполнять следующие операции с IL-программами:

- создать программу;
- установить тип программы;
- сохранить программу в файл;
- вставить программу из файла;
- ввести или отредактировать основное имя программы;
- ввести или отредактировать дополнительное имя программы;
- редактировать программу;
- транслировать программу;
- просматривать дамп трансляции;
- добавить программу к системе.

Создание IL-программ. Для создания новой IL-программы надо выполнить команду «Создать» меню Техно IL редактора базы каналов. При этом на экран выводится диалог Техно IL. Здесь в соответствующих полях следует ввести тип и имена новой программы (основное и дополнительное). После этого надо набрать текст программы. Для набора текста предусмотрено специальное окно данного диалога.

После этих операций новая программа может быть оттранслирована и добавлена в систему или сохранена в виде файла. В обоих случаях она будет в дальнейшем доступна для редактирования.

Сохранение в файл и вставка IL-программ. Любая IL-программа может быть сохранена в отдельный файл в текстовом формате. Для этого надо, находясь в диалоге Техно IL, выполнить команду «Сохранить как» из его меню «Файл». При этом на экран выводится стандартный диалог выбора файла. Имя файла для сохранения и путь к нему можно указать любые. Однако по умолчанию предлагается использовать файл с именем, образованным из основного имени программы, и поддиректорию «ASM» рабочей директории ТРЕЙС МОУД. Тексты программ на языке Техно IL сохраняются в файлы с расширением «il».

Сохраненная ранее программа может быть снова загружена в редактор. Для этого надо, находясь в диалоге Техно IL, выполнить команду «Открыть» из его меню «Файл». При этом на экран выводится стандартный диалог выбора файла. После выбора требуемого файла сохраненная в нем программа будет загружена. При этом в соответствующих полях диалога Техно IL выводятся ее имена и тип, а в окне редактирования – текст, который становится доступен для изменения.

Трансляция IL-программ. Чтобы вставить IL-программу в систему, надо ее сначала оттранслировать. Для этого следует загрузить требуемую программу в диалог Техно IL и нажать левой клавишей мыши на кнопке «Трансляция».

Если трансляция программы прошла успешно, то в окне вывода сообщений появится строка, содержащая текст «OK», и снимается блокировка с кнопки «Добавить». При обнаружении ошибок в тексте программы в окно сообщений будет выводиться соответствующая информация.

Для облегчения поиска ошибок в тексте программы можно вывести в окно сообщений дамп трансляции. Этот дамп содержит коды трансляции нормально оттранслированных строк программы и тексты этих строк. Если строка содержала ошибку, то вместо кода трансляции выводится соответствующее сообщение.

Добавление IL-программ в систему. Чтобы разработанная IL-программа была доступна для использования при запуске МРВ, ее необходимо добавить в систему. Для этого надо, находясь в диалоге Техно IL, нажать левой клавишей мыши на кнопке «Добавить».

Внимание! Добавление программы в систему доступно только после успешного завершения ее трансляции.

Добавление функционального блока. Если для IL-программы был задан тип «FB», то после выполнения операции добавления в систему будет вставлен новый функциональный блок, реализующий функцию, описанную в программе. Этот блок помещается в один из функциональных разделов: Техно IL_1, Техно IL_2 или Техно IL_3. Раздел выбирается по наличию в нем свободных мест для добавления новых блоков. Названию нового блока присваивается дополнительное имя IL-программы.

Внимание! После добавления в систему нового функционального блока надо выйти из диалога Техно IL. Только после этого можно перейти к добавлению следующего блока. В противном случае второй добавляемый блок затрет первый и займет его место в списке функциональных блоков.

Максимальное количество добавляемых в систему функциональных блоков равно 54. При исчерпании этого количества новых блоков их добавление в систему блокируется. В этом случае, чтобы добавить новый блок, надо удалить любой из добавленных ранее. Эта операция будет описана ниже.

Добавленный в систему функциональный блок становится доступным для использования на любом узле любого редактируемого после этого проекта.

Добавление метапрограммы. Если для IL-программы был установлен тип «PRG», то операция добавления осуществит подключение указанной программы к текущему редактируемому узлу. Для каждого узла можно подключить до 16 таких программ. При этом вместо дополнительного имени программы надо указать ее номер. Этот номер должен лежать в диапазоне от 0 до 15. Если вместо номера в поле дополнительного имени программы был набран текст, то осуществляется добавление программы под номером 0.

Внимание! Если для узла была добавлена метапрограмма с каким-либо номером, то последующее добавление другой программы этому же узлу с тем же номером уничтожит предыдущую программу.

Удаление программы IL-программ из системы. Если в систему были добавлены IL-программы, использование которых больше не требуется, то можно их удалить. Для этого надо выполнить команду «Удалить» из меню Техно IL.

Редактирование добавленных IL-программ. Для коррекции добавленных в систему программ надо выполнить команду «Редактировать» меню Техно IL. При этом на экран выводится диалог выбора IL-программы, который был показан в предыдущем разделе. После выбора в нем имени любой программы осуществляется переход в диалог Техно IL, где она становится доступна для редактирования.

После внесения изменений в программу надо ее снова оттранслировать и дать команду «Добавить».

Если осуществлялось редактирование IL-программы функционального блока, то при выполнении операции добавления производится контроль изменения структуры его входов и выходов. Если количество входов или выходов блока в результате редактирования программы изменилось, то выдается соответствующее предупреждение и обновление программы в системе не осуществляется. Для реализации этого необходимо сначала сохранить отредактированную программу в файл, затем удалить ее из списка добавленных в систему IL-программ. После этого можно загрузить сохраненную программу из файла и повторить ее добавление.

П р и л о ж е н и е Б

РЕКОМЕНДАЦИИ ПО НАСТРОЙКЕ ПИД-РЕГУЛЯТОРА

Выбор периода квантования. Для того, чтобы эффект квантования по времени мало сказывался на динамику системы цифрового регулирования, рекомендуется выбирать период квантования из соотношения:

$$T_{95}/15 < T_{к} < T_{95}/5,$$

где T_{95} – это время достижения выходным сигналом уровня 95 % от установившегося значения при подаче на вход объекта ступенчатого сигнала. Если объект первого порядка, то $T_{95} \approx \tau + 3T$.

Другой подход к выбору величины периода квантования основан на рекомендациях американских ученых Зиглера и Никольса, согласно которым $T_{к} = 0{,}1T_{кр}$, где $T_{кр}$ – период критических колебаний объекта управления.

В реальных условиях при управлении инерционными процессами значение $T_{к}$ берется от 1 секунды до нескольких минут (в газоанализаторах, например, 1 раз в час). При регулировании малоинерционных процессов (например, расхода жидкости) величина $T_{к}$ может составлять десятые доли секунды. Нельзя выбирать большие периоды опроса, особенно для ответственных процессов, так как в этом случае аварийные ситуации будут ликвидироваться слишком медленно. В то же время, при слишком малом периоде опроса повышаются требования к быстродействию ЭВМ и увеличивается влияние шумов.

Упрощенная методика расчета настроек дискретного ПИД-регулятора. С целью упрощения процедуры настройки цифрового ПИД-регулятора рекомендуется (согласно Зиглеру и Никольсу) выбирать следующие значения отношений

$$T_{к} / T_{и} = 0{,}2, \quad T_{d} / T_{к} = 1{,}25$$

при $T_{к} = 0{,}1T_{кр}$, где $T_{кр}$ – период критических колебаний объекта управления.

Таким образом, в алгоритме

$$u_{р}(t_n) = K_{р}\Delta y(t_n) + \frac{K_{р}}{T_{и}}\sum_{t_i=t_0}^{t_i=t_n}\Delta y(t_i) + \frac{K_{р}}{T_{d}}\sum_{t_i=t_0}^{t_i=t_n}\Delta y(t_i), \qquad (1)$$

настраиваемым параметром остается лишь один коэффициент усиления регулятора $K_{р}$, чем и объясняется простота и широкая распространенность этого метода настройки.

После определения периода квантования единственным настраиваемым параметром в алгоритме (1) является коэффициент усиления цифрового регулятора $K_{р}$. Его достаточно просто настроить экспериментально, так чтобы декремент затухания в системе был равен Д = 1/4.

$K_{р} \cdot K$ $K_{р} \cdot K$

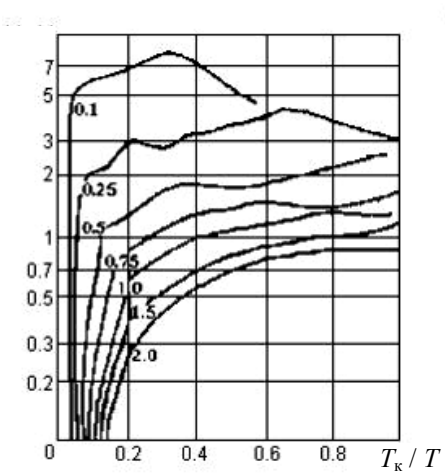


Рис. 1 Номограмма для ПИ

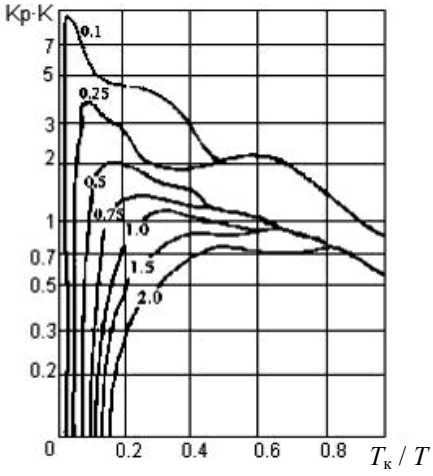


Рис. 2 Номограмма для ПИД

Однако при известных параметрах объекта управления определение величины $K_р$ возможно с помощью номограмм, приведенных на рис. 1 и 2 [5]. Эти номограммы построены с помощью ЭВМ путем минимизации критерия по величине $K_р$.

Пример. Определить настройки цифрового ПИ-регулятора для объекта первого порядка с запаздыванием, с параметрами: $K = –b/a = 2{,}4$; $T = 612$ с; $\tau = 480$ с; $T_к = 120$ с, где K – коэффициент усиления объекта; a, b – параметры модели объекта; T – постоянная времени объекта; τ – запаздывание; $T_к$ – период квантования.

Для определения величины необходимо найти отношения:

$$\tau / T = 480 / 612 = 0{,}78, \quad T_к / T = 120 / 612 = 0{,}196.$$

По номограмме на рис. 1 найдем $KK_р = 0{,}85$, тогда $K_р = 0{,}85 / 2{,}4 = 0{,}354$.

Расчет настроек цифрового регулятора. Здесь, как и ранее, предполагается, что переходная характеристика объекта управления аппроксимирована звеном первого порядка с запаздыванием. Выбрав период квантования $T_к$, рассчитывают параметры настройки дискретного ПИ- или ПИД-регулятора по нижеприведенным формулам [5].

Для ПИ-регулятора:

$$K_р^* = \frac{0{,}9T}{\tau + T_к / 2} - \frac{0{,}135TT_к}{(\tau + T_к / 2)^2};$$

Для ПИД-регулятора:

$$K_р^* = \frac{1{,}2T}{\tau + T_к / 2} - \frac{0{,}3TT_к}{(\tau + T_к / 2)^2}.$$

В этих формулах учтено запаздывание на величину $T_к / 2$, свойственное всем замкнутым цифровым системам регулирования.