

ПАРАЛЛЕЛЬНЫЕ КОМПОЗИЦИИ В ЯЗЫКЕ ОПИСАНИЯ ЗАКОНОВ УПРАВЛЕНИЯ

© 2015 А. М. Фрумкин

*ст. науч. сотрудник кафедры математического анализа
и прикладной математики, канд. техн. наук
e-mail: frumkinam@mail.ru*

Курский государственный университет

Предлагается определение параллельной композиции управляющих операций. Усложняется синтаксис языка описания законов управления для описания параллельных композиций. Приводятся примеры описаний законов управления с применением параллельных композиций. Рассматриваются способы синхронизации параллельных операций.

Ключевые слова: управляющая операция, параллельная совместимость операций, параллельная композиция операций, синтаксический ограничитель, процедура регулирования, синхронизация операций.

Введение

В работе [Фрумкин 2015] был описан общий подход к определению языка описания законов управления. На представленном варианте языка удобно описывать прежде всего композиции однородных операций управления, то есть операций, имеющих одинаковые наборы переменных. С точки зрения технической реализации закона управления композиция однородных операций описывает функционирование одного и того же устройства в различных ситуациях.

Для расширения возможностей языка желательно включить в него средства описания композиций, которые определяют взаимодействие нескольких устройств между собой. Таким композициям посвящена данная статья. В ней определяется параллельная композиция операций и приводятся примеры описаний законов управления, которые без применения параллельных композиций описывать неудобно. В процессе построения описаний вводится тип переменной «перечисляемое множество» и несколько новых элементарных операций, не определенных в работе [Фрумкин 2015]. В статье сохранены терминология и обозначения из статей [Фрумкин 2013, Фрумкин 2014, Фрумкин 2015].

1. Определение параллельной композиции операций

Определение 1. Если (Ω, T) – набор переменных, $x \in p(\Omega, T)$ – полный процесс, $\Lambda \subseteq \Omega$, то сужением x на Λ назовем процесс

$$x_\Lambda = \{ ((\alpha, t), z): \alpha \in \Lambda \wedge t \in [0, \infty) \wedge z = x(\alpha, t) \}.$$

Определение 2. Управляющие операции $(\Omega_0, \Omega_1, T, e, f)$, и $(\Lambda_0, \Lambda_1, S, h, g)$ назовем параллельно совместимыми, если $\Omega_1 \cap \Lambda_1 = \emptyset$ и

$$\forall \alpha \in (\Omega_0 \cap \Lambda_1) \cup (\Omega_0 \cap \Lambda_0) \cup (\Lambda_0 \cap \Omega_1) \quad T(\alpha) = S(\alpha).$$

Другими словами, управляющие операции параллельно совместимы, если множества их выходных переменных не пересекаются и любая переменная, участвующая в определении обеих операций, имеет согласно этим определениям один и тот же тип.

Определение 3. Пусть управляющие операции $(\Omega_0, \Omega_1, T, e, f)$, и $(\Lambda_0, \Lambda_1, S, h, g)$ параллельно совместимы. Их параллельной композицией назовем операцию $(\Gamma_0, \Gamma_1, Q, q, r)$, компоненты которой определяются следующим образом.

Множество входных переменных параллельной композиции операций – это множество $\Gamma_0 = (\Omega_0 \setminus \Lambda_1) \cup (\Lambda_0 \setminus \Omega_1)$, множество выходных переменных – это множество $\Gamma_1 = \Omega_1 \cup \Lambda_1$. Функция типов $Q = S \cup T$.

Процесс $x \in r(\Gamma_0, Q)$ и исходный набор $u \in (\Omega_1 \cup \Lambda_1) \wedge Q$ назовем допустимым, если найдутся и единственные процессы $y \in r(\Omega_1, T)$, $z \in r(\Lambda_1, S)$, такие, что выполняются равенства:

$$\begin{aligned} f(u_{\Omega_1}, x_{(\Omega_0 \setminus \Lambda_1)} \cup z_{(\Omega_0 \cap \Lambda_1)}) &= y, \\ g(u_{\Lambda_1}, x_{(\Lambda_0 \setminus \Omega_1)} \cup y_{(\Lambda_0 \cap \Omega_1)}) &= z. \end{aligned} \quad (1)$$

Область определения $Dm(p)$ оператора p – множество допустимых пар $(u, x) \in (\Omega_1 \cup \Lambda_1) \wedge Q \times r(\Gamma_0, Q)$. Для каждой пары $(u, x) \in Dm(p)$ выходной процесс определяется равенством $p(u, x) = y \cup z$, где процессы y, z определяются системой уравнений (1).

Событие выхода определяется формулой

$$\begin{aligned} q = \{ & ((u, x), t) : (u, x) \in Dm(p) \wedge \\ & ((u_{\Omega_1}, x_{(\Omega_0 \setminus \Lambda_1)} \cup p(u, x)_{(\Omega_0 \cap \Lambda_1)}) \dot{\cap} Dm(e) \wedge t = h(u_{\Lambda_1}, x_{(\Lambda_0 \setminus \Omega_1)} \cup p(u, x)_{(\Lambda_0 \cap \Omega_1)})) \\ & \vee ((u_{\Lambda_1}, x_{(\Lambda_0 \setminus \Omega_1)} \cup p(u, x)_{(\Lambda_0 \cap \Omega_1)}) \dot{\cap} Dm(h) \wedge t = e(u_{\Omega_1}, x_{(\Omega_0 \setminus \Lambda_1)} \cup p(u, x)_{(\Omega_0 \cap \Lambda_1)})) \\ & \vee t = \min(e(u_{\Omega_1}, x_{(\Omega_0 \setminus \Lambda_1)} \cup p(u, x)_{(\Omega_0 \cap \Lambda_1)}), h(u_{\Lambda_1}, x_{(\Lambda_0 \setminus \Omega_1)} \cup p(u, x)_{(\Lambda_0 \cap \Omega_1)})) \}. \end{aligned}$$

Формула имеет следующую словесную интерпретацию. Событие выхода из параллельной композиции происходит тогда и только тогда, когда происходит событие выхода хотя бы для одной из параллельных операций. Если происходит выход только из одной параллельной операции, момент выхода из параллельной композиции совпадает с моментом выхода из этой операции. Если же происходят события выхода из обеих параллельных операций, момент выхода из параллельной композиции является минимумом моментов выхода для каждой операции.

Рассмотренное определение параллельной композиции обобщается на произвольный набор попарно совместимых операций. В данном случае удобно изменить способ обозначения компонентов кортежей рассматриваемых операций. Одинаковые по смыслу объекты, находящиеся в одинаковых позициях кортежа, будем обозначать одинаковыми буквами. Индексы будут обозначать номер рассматриваемого кортежа (операции), но не смысл объекта.

Определение 4. Пусть $(\Omega_k, \Lambda_k, T_k, e_k, f_k)$ – конечная последовательность из n управляющих операций. Здесь буквой k обозначены индексы элементов k -й операции, $k \in [1..n]$. Будем называть эту последовательность допустимой для параллельной композиции, если выполняются следующие условия:

а) выходные множества переменных попарно не пересекаются, то есть $\forall k, m \in [1..n] \ k \neq m \Rightarrow \Lambda_m \cap \Lambda_k = \emptyset$;

б) любая переменная, принадлежащая сразу двум из рассматриваемых множеств $\Omega_1, \Omega_2, \dots, \Omega_n, \Lambda_1, \Lambda_2, \dots, \Lambda_n$, имеет единый тип, то есть $\forall k, m \in [1..n] \ \alpha \in \Omega_m \cap \Omega_k \Rightarrow T_m(\alpha) = T_k(\alpha) \wedge \alpha \in \Omega_m \cap \Lambda_k \Rightarrow T_m(\alpha) = T_k(\alpha)$.

Определение 5. Пусть $(\Omega_k, \Lambda_k, T_k, e_k, f_k)$ – последовательность из n управляющих операций, допустимая для параллельной композиции. Параллельной композицией операций из этой последовательности назовем операцию $(\Omega, \Lambda, T, e, f)$, компоненты которой определяются следующим образом.

Множество выходных переменных параллельной композиции – это множество $\Lambda = \bigcup_{k=1}^n \Lambda_k$, множество входных переменных – это множество $\Omega = (\bigcup_{k=1}^n \Omega_k) \setminus \Lambda$. Функция типов $T = \bigcup_{k=1}^n T_k$.

Исходный набор $u \in \Lambda \wedge T$ и процесс $x \in p(\Omega, T)$ назовем допустимыми, если найдется единственный процесс $y \in p(\Lambda, T)$, обладающий свойством:

$$\forall k \in [1..n] \quad f_k(u_{\Lambda_k}, x_{(\Omega_k \setminus \Lambda)} \cup y_{(\Omega_k \cap \Lambda)}) = y_{\Lambda_k}. \quad ($$

Область определения $Dm(f)$ оператора f – множество допустимых пар $2) (u, x) \in \Lambda \wedge T \times p(\Omega, T)$. Для каждой пары $(u, x) \in Dm(p)$ выходной процесс $y = p(u, x)$ определяется системой уравнений (2).

Для определения события e удобно доопределить все события e_k на множестве $Dm(f_k) \setminus Dm(e_k)$ значением ∞ . Тогда событие e определяется формулой

$$e = \{ ((u, x), t) : (u, x) \in Dm(f) \wedge t = \min_{k \in [1..n]} e_k(u_{\Lambda_k}, x_{(\Omega_k \setminus \Lambda)} \cup f(u, x)_{(\Omega_k \cap \Lambda)}) \}.$$

Момент выхода из параллельной композиции равен минимуму моментов выхода для каждой операции композиции. Событие выхода из параллельной композиции происходит (то есть $e(u, x) < \infty$) тогда и только тогда, когда происходит событие выхода хотя бы для одной из операций композиции.

Определение параллельной композиции усложняет понятие алгоритма управления [Фрумкин 2015]. В дереве алгоритма появляются узлы параллельной композиции. Узлу параллельной композиции, как и всякому узлу алгоритма, соответствует собственный набор переменных [Фрумкин 2014, 2015]. Внутри узла параллельной композиции эти переменные делятся на группы (возможно, пересекающиеся), относящиеся к различным параллельным операциям. При этом входная переменная в наборе переменных параллельной композиции остается входной в любой группе. Напротив, выходная переменная, оказывается выходной в единственной группе.

Согласно рассмотренному определению параллельной композиции выходной процесс операции есть решение некоторой системы уравнений. Поэтому каждое описание композиции требует доказательства ее корректности, то есть существования и единственности выходных процессов для заданного класса входных аргументов. Для обеспечения корректности некоторых описаний может быть усложнен способ определения выходного процесса и, соответственно, понятие допустимости входного процесса. В данной статье прежде всего преследуется цель показать возможности построения текстовых описаний законов управления с использованием параллельных композиций. Поэтому рассматриваются простые с логической точки зрения композиции и анализ корректности описаний не проводится.

2. Синтаксические конструкции, описывающие параллельные композиции

Для описания параллельных композиций в грамматике языка изменяется правило, расшифровывающее понятие «композиция». В него включается понятие «параллельная композиция»:

$\langle \text{композиция} \rangle ::= \langle \text{условная композиция} \rangle \mid \langle \text{композиция выбора} \rangle \mid \langle \text{композиция повторения} \rangle \mid \langle \text{последовательная композиция} \rangle \mid \langle \text{параллельная композиция} \rangle \$$

Конструкция параллельной композиции выделяется синтаксическими ограничителями **prc** и **end**. Сокращение **prc** расшифровывается как «параллельная композиция».

В языке без параллельных композиций нет необходимости в специальных словах, выделяющих последовательные композиции, потому что последовательное расположение элементов композиции внутри текста естественно задает последовательную композицию.

Внутри описания параллельной композиции последовательное расположение описаний операций интерпретируется как перечисление элементов параллельной композиции. А так как внутри описания параллельной композиции могут присутствовать несколько последовательных композиций, описывающих параллельно выполняемые операции, в язык необходимо ввести ограничители последовательной композиции, описываемой внутри параллельной. Это символы **sqc** и **end**. Сокращение **sqc** расшифровывается как «последовательная композиция».

В результате в грамматику добавляется такой набор правил.

```

<параллельная композиция> ::= prc <элементы параллельной композиции> end $
<элементы параллельной композиции> ::=
<элемент параллельной композиции> <элемент параллельной композиции> |
<элемент параллельной композиции> <элементы параллельной композиции> $
<элемент параллельной композиции> ::=
sqc <последовательная композиция> end |
<условная композиция> | <композиция выбора> | <композиция повторения>
<присваивание> | <ссылка>$

```

Описания условной композиции, композиции выбора, композиции повторения, присваивания и ссылки внутри описания параллельной композиции непосредственно определяют одну из параллельных операций.

Специальное описание того, какие из переменных являются входными, а какие выходными, для параллельной композиции не требуется по тем же причинам, по которым не требуется описывать вид переменной в заголовке процедуры. Если каждая ссылка и присваивание определяют виды своих переменных, то мы можем классифицировать переменные каждой композиции внутри описания, а также проверить, выполняются ли условия совместимости, рассматриваемые при определении параллельной композиции.

3. Примеры применения параллельных композиций при описании законов управления

3.1. Пропорционально-интегральный регулятор с непрерывным изменением управляющей переменной

Опишем регулятор, задающий коэффициент заполнения импульса некоторого широтно-импульсного исполнительного устройства [Фрумкин 2005]. Его выходной сигнал ограничен и изменяется в промежутке [0,1]. Ограничение задается кусочно-непрерывной функцией

$$L(x) = \begin{cases} 0, & \text{если } x < 0 \\ x, & \text{если } 0 \leq x \leq 1 \\ 1, & \text{если } x > 1 \end{cases}$$

Эта функция может быть описана на алгоритмическом языке [Фрумкин 2015].

ПИ_РЕГУЛЯТОР (real x, real y)

par (real alpha, % коэффициент пропорционального регулирования

real beta, % коэффициент интегратора

real s % уставка регулирования

real y0 % начальное значение интегратора %)

real I := y0; % начальная установка значения интегратора

prc int(beta*(s-x), I); y*=L(I- alpha*(x-s)); **end**

end % конец процедуры

3.2. Генератор пилообразного сигнала с разрывами

Моделируемые в данном и последующем пункте устройства называют также генераторами линейно изменяющегося напряжения (ГЛИН).

ГЛИН(real y)

```

par (real U; % Максимальное значение
      real T; % Время возрастания )
repeat
  y:=0; prc int(U/T,y); delay(T); end
end
end

```

3.3. Генератор непрерывного пилообразного сигнала

```

ГЛИН(real y)
par (real U; % Максимальное значение
      real T; % Время возрастания и убывания)
repeat
  y:= - U;
  prc int(2*U/T,y); delay(T); end
  y:= U;
  prc int(-2*U/T,y); delay(T); end
end
end

```

В описании процедуры можно было бы сделать единственное начальное присваивание переменной y ($y := -U$;) до описания композиции повторения, а внутри композиции присваиваний переменной y не делать. Приведенный вариант может быть более надежно реализован технически.

3.4. Генератор синусоидально изменяющегося сигнала

```

ГСИС(real x)
par (real omega, % частота сигнала
      real U % амплитуда сигнала %)
x:=0; % начальное значение сигнала
real y:= omega*U; % начальное значение производной сигнала
prc int(y,x); int(-omega^2*x,y); end
end

```

3.5. Регулятор частоты

Рассматриваемая процедура может применяться при регулировании частоты напряжения автономных дизель-генераторов [Токарев 2001] с симметричной нагрузкой, не искажающей выходного напряжения. Она строится как параллельная композиция двух процедур. Первая процедура измеряет рассогласование, то есть разность длительности цикла изменения входной переменной (напряжения фазы) и требуемой длительности (уставки периода). Вторая процедура, используя рассогласование как входную переменную, осуществляет частотно-импульсное регулирование.

```

ИЗМЕРЕНИЕ_РАССОГЛАСОВАНИЯ
(real x, % входной сигнал (напряжение) % real y % рассогласование %)
par( real s % уставка периода %)
y:=0; % начальная установка выходной переменной
real T0:=0; % текущая длина полупериода положительных значений
real T1:=0; % текущая длина полупериода отрицательных значений
wait(x<0); % фиксация полупериода отрицательных значений
wait(x>0); % фиксация начала полупериода положительных значений
repeat
  prc timer(T0); wait(x<0); end
  prc timer(T1); wait(x>0); end
  y:=T0+T1-s; T0:=0; T1:=0;
end

```

end

Текст процедуры ЧИМ-регулирования приведен в статье [Фрумкин 2015]. Общая процедура имеет следующий вид.

РЕГУЛИРОВАНИЕ_ЧАСТОТЫ(real x, real y)

prc ИЗМЕРЕНИЕ_РАССОГЛАСОВАНИЯ (real x, real y);

ЧИМ_Регулятор (real y, real x); **end**

end

3.6. Управление нагревательным баком

В нагревательном баке осуществляется накопление воды и одновременный ее подогрев. Такое устройство применяется в технологических линиях пищевых и химических производств. Для нагревательного бака решаются задачи регулирования двух переменных: уровня воды и ее температуры. Если обнаруживается невозможность регулирования уровня, регулирование температуры прекращается. Регулирование уровня рассматриваемой процедурой осуществляется по релейному закону с двумя порогами, регулирование температуры – по пропорционально–интегральному закону.

РЕГУЛИРОВАНИЕ_УРОВНЯ

(real L% уровень %, bit x % выходная переменная)

par (real Lmin, % нижний порог регулирования

real Lmax % верхний порог регулирования %)

x:=0; % начальная установка выходной переменной

repeat

switch x

case 0: wait(L<=Lmin); x:=1;

case 1: wait(x>=Lmax); x:=0;

end

end

end

В описании закона регулирования температуры используется специальный тип переменной – перечислимое множество, которое определяется как множество в математике, с помощью фигурных скобок. Так же как при описании ПИ-регулятора (п.0), будем считать, что выходная переменная регулятора температуры меняется в промежутке [0,1], причем ноль соответствует полному отсутствию подогрева.

РЕГУЛИРОВАНИЕ_ТЕМПЕРАТУРЫ(real T, real L, real y)

par (real Loff, % пороговое значение уровня отключения регулятора

real Lon % пороговое значение уровня включения регулятора %)

{Ожидание, Регулирование} s:= Ожидание; % ситуация регулирования

repeat

switch s

case Ожидание: y:=0; wait(L>=Lon); s:=Регулирование;

case Регулирование: **prc sqc** wait(x<L); s:= Ожидание; **end**

ПИ_РЕГУЛЯТОР (T,y);

end

end

end

end

% -----

УПРАВЛЕНИЕ_НАГРЕВАТЕЛЬНЫМ_БАКОМ

(real T, % температура воды

real L, % уровень воды

bit x, % переменная управления уровнем

```

real y, % переменная управления температурой)
prc РЕГУЛИРОВАНИЕ_УРОВНЯ(L,x);
    РЕГУЛИРОВАНИЕ_ТЕМПЕРАТУРЫ(T,L,y); end
end

```

4. Синхронизация параллельных операций

В процедуре регулирования частоты, описанной в предыдущем пункте, процедура измерения взаимодействует с ЧИМ-регулятором асинхронно. Это означает, что в начале каждого цикла регулирования ЧИМ-регулятор использует то значение регулируемой величины (периода), которое было получено при последнем измерении. Начало и завершение цикла регулирования никак не связаны с циклами изменения (и измерения периода) напряжения.

Иногда возникает задача сделать так, чтобы параллельные процедуры выполнялись в одинаковых промежутках времени, или, другими словами, были синхронизированы между собой. Рассмотрим пример процедуры широтно-импульсного регулирования напряжения автономного синхронного генератора [Фрумкин 2005], в которой циклы регулирования напряжения совпадают с циклами изменения напряжения одной из фаз. Такой упрощенный вариант регулирования допустим, если нагрузка симметрична и не искажает синусоидальной формы выходного напряжения. Определим новую (по отношению к набору элементарных процедур, определенных в статье [Фрумкин 2015]) элементарную операцию.

Продолжение – это неполная (пустая) операция ($\emptyset, \emptyset, \emptyset, \emptyset, \emptyset$) без входных и выходных переменных. Ссылка на операцию продолжения – это слово «continue». Если мы расположим операцию продолжения в терминальном узле дерева алгоритма управления, то согласно общему принципу расширения [Фрумкин 2015] в узле будет определена операция сохранения значений выходных переменных из набора переменных данного узла. Описываемая далее процедура цикла регулирования включается в параллельную композицию с другой операцией, которая и определяет время завершения цикла регулирования.

```

ЦИКЛ_РЕГУЛИРОВАНИЯ
(real T, % длина предшествующего цикла
real I, % интеграл модуля регулируемой переменной в предшествующем цикле
bit y % выходная переменная регулятора %)
par (real sigma, % свободный член закона регулирования
    real alpha % коэффициент пропорционального регулирования %)
real tau:= T*L(sigma-alpha*(I/T-s)); % вычисление длительности импульса
y:=1; delay(tau); y:=0; % формирование импульса
continue; % ожидание конца цикла регулирования
end

```

В описанной процедуре выбран принцип пропорционального регулирования, L – функция ограничения (п. 0). Процедура цикла регулирования используется в общей процедуре, которая параллельно вычисляет длительность цикла регулирования (полупериода изменения напряжения) и интегрирует абсолютную величину напряжения в каждом цикле.

```

РЕГУЛИРОВАНИЕ_НАПРЯЖЕНИЯ
(real x, % напряжение % bit y % управляющая переменная %)
y:=0; % начальная установка управляющей переменной
real T:=0; % текущая длина полупериода
real t:=0; % рабочая переменная для определения длины полупериода
real I:=0; % интеграл за полупериод
real i:=0; % рабочая переменная для интегрирования

```

```

wait(x<0); % первая фиксация отрицательного полупериода
wait(x>0); % первая фиксация начала положительного полупериода
prc wait(x<0); % измерения в первом полупериоде
    sqc t:=0; timer(t); end sqc i:=0; int(abs(x),i); end
end
T:= t; I:=i;
repeat % измерения и регулирование
    prc wait(x<0); % измерения в первом полупериоде
        sqc t:=0; timer(t); end sqc i:=0; int(abs(x),i); end
        ЦИКЛ_РЕГУЛИРОВАНИЯ(T,I,y);
    end
    T:= t; I:=i;
    prc wait(x>0); % измерения в первом полупериоде
        sqc t:=0; timer(t); end sqc i:=0; int(abs(x),i); end
        ЦИКЛ_РЕГУЛИРОВАНИЯ(T,I,y);
    end
    T:= t; I:=i;
end
end

```

Другой способ решения задачи синхронизации обеспечивается введением еще двух специальных элементарных операций.

Импульс. Это неполная операция $(\emptyset, \{\alpha\}, \{(\alpha, \{0,1\})\}, \emptyset, p)$ без входных переменных, имеющая одну выходную переменную α битового (булева) типа: $T(\alpha) = \{0,1\}$. Выходной процесс не зависит от исходного значения выходной переменной. Оператор p можно рассматривать просто как процесс, определяемый формулой

$$p = \{(t,z) \in [0,\infty) \times \{0,1\} : (t=0 \wedge z=1) \vee (t>0 \wedge z=0)\}.$$

Ссылку на операцию импульса будем обозначать $puls(x)$, где x – изменяемая переменная.

Ожидание с задержкой. Это неполная операция $(\{\alpha\}, \emptyset, \{(\alpha, \{0,1\})\}, e, \emptyset)$ без выходных переменных, имеющая одну входную переменную α битового (булева) типа: $T(\alpha) = \{0,1\}$. Событие выражается в том, что входная переменная принимает значение 1 позже, чем в начальный момент:

$$e = \{(u,t) : u \in p(\{\alpha\}, T) \wedge t = \inf\{\tau > 0 : u(\alpha, \tau) = 1\}\}.$$

В отличие от операции ожидания, определенной в [Фрумкин 2015], ссылку на операцию ожидания с задержкой будем обозначать $waitd(x)$, где x – переменная, для которой ожидается значение 1.

С использованием новых понятий описание процедуры регулирования напряжения можно представить с помощью следующей последовательности описаний.

СИНХРОНИЗАЦИЯ

```

(real x, % напряжение %    bit f % признак начала (окончания) цикла
регулирования %)
wait(x<0); % первая фиксация отрицательного полупериода
wait(x>0); % первая фиксация начала положительного полупериода
repeat % измерения и регулирование
    prc pulse(f); wait(x<0); end % измерения в первом полупериоде
    prc pulse(f); wait(x>0); end % измерения во втором полупериоде
end
end
% -----

```

ИЗМЕРЕНИЕ_ПОЛУПЕРИОДА

(bit f, % признак начала (окончания) цикла регулирования

real T % текущая длина цикла регулирования %)

T:=0; real t:=0; wait(f); % ожидание начала первого цикла измерения

repeat

prc waitd(f); % ожидание конца текущего цикла измерения

sqc t:=0; timer(t); **end** % измерение

end

 T:= t; % сохранение результата

end

end

% -----

ИНТЕГРИРОВАНИЕ(bit f, real x, real I % интеграл за полупериод %)

I:=0; real i:=0; wait(f); % ожидание начала первого цикла измерения

repeat

prc waitd(f); % ожидание конца текущего цикла измерения

sqc i:=0; int(abs(x),i); **end** % интегрирование

end

 I:=i; % сохранение результата

end

end

% -----

РЕГУЛИРОВАНИЕ(bit f, real T, real I, bit y % управляющая переменная %)

par (real sigma, % свободный член закона регулирования

 real alpha % коэффициент пропорционального регулирования %)

wait(f); % ожидание начала первого цикла измерения

wait(f); % ожидание начала первого цикла регулирования

repeat

prc waitd(f); % ожидание конца цикла регулирования

sqc real tau:= T*L(sigma-alpha*(I/T-s)); % вычисление длительности импульса

 y:=1; delay(tau); y:=0; % формирование импульса **end**

end

end

end

% -----

РЕГУЛИРОВАНИЕ_НАПРЯЖЕНИЯ

(real x, % напряжение % bit y %управляющая переменная %)

bit f:=0; % признак начала (завершения) цикла регулирования

real T:=0; % текущая длина полупериода

real I:=0; % интеграл за полупериод

prc

 СИНХРОНИЗАЦИЯ(x,f);

 ИЗМЕРЕНИЕ_ПОЛУПЕРИОДА(f,T);

 ИНТЕГРИРОВАНИЕ(f,x,I);

 РЕГУЛИРОВАНИЕ(f,T,I,y);

end

end

С помощью операции «pulse» процедура синхронизации генерирует последовательность «мгновенных» импульсов, обозначающих граничные моменты

циклов регулирования. С помощью операции «waitd» процедуры измерения и регулирования пропускают импульс начала текущего цикла и обнаруживают импульс в момент его окончания. Рассмотренное описание процедуры регулирования напряжения является более громоздким, чем предыдущее, но, с другой стороны, оно лучше структурировано.

Библиографический список

Токарев Л.Н. Системы автоматического регулирования. СПб.: НОТАБЕНЕ, 2001. 188 с.

Фрумкин А.М. Синтез законов управления автономным синхронным генератором в системе автоматизированных испытаний: дис. ... канд. тех. наук. М., 2005. 225 с.

Фрумкин А.М. К определению понятия события при описании процессов в системах управления // Ученые записки: электронный научный журнал Курского государственного университета. 2013. №1. URL: <http://scientific-notes.ru/pdf/029-01.pdf> (дата обращения: 13.11.2015).

Фрумкин А.М. Об объектах, описываемых алгоритмическим языком // AUDITORIUM. Электронный научный журнал Курского государственного университета. 2014. №2. URL: <http://auditorium.kursksu.ru/pdf/002-001.pdf> (дата обращения: 13.11.2015)

Фрумкин А.М. Об объектах языка описания законов управления // AUDITORIUM. Электронный научный журнал Курского государственного университета. 2015. №3. URL: <http://auditorium.kursksu.ru/pdf/007-001.pdf> (дата обращения: 13.11.2015).