

## О СИТУАЦИОННО-СОБЫТИЙНОМ ПОДХОДЕ К УПРАВЛЕНИЮ ВЗАИМОДЕЙСТВУЮЩИМИ ДИСКРЕТНО-НЕПРЕРЫВНЫМИ ПРОЦЕССАМИ

В. М. Шпаков<sup>а</sup>, канд. техн. наук, доцент

<sup>а</sup>Санкт-Петербургский институт информатики и автоматизации РАН,  
Санкт-Петербург, РФ

**Постановка проблемы:** процессы управления зависят от логики и динамики развития процессов в объекте управления и должны отвечать требованиям логики и динамики управления. Задача состоит в спецификации и реализации требуемых зависимостей. При этом предъявляются требования к удобству и выразительности спецификаций и эффективности получаемых на их основе программных реализаций процессов управления. Логико-динамические ситуации, используемые для спецификации гибридных процессов, пока не находят применения для реализации процессов управления. **Цель:** экспериментальное исследование технологии реализации процессов управления, основанной на использовании логико-динамических ситуаций. **Результаты:** разработан метод спецификации процессов управления дискретно-непрерывными процессами, основанный на применении трансформационных правил, условными частями которых являются логико-динамические ситуации. Метод обладает хорошей выразительностью, обеспечивает низкую трудоемкость разработки, прозрачность спецификаций и эффективность компьютерной реализации процессов. Возможности спецификации иллюстрируются на примере разработки компьютерной модели системы автоматического управления позиционированием перевернутого маятника. **Практическая значимость:** использование логико-динамических ситуаций для спецификации процессов управления позволит упростить разработку систем управления дискретно-непрерывными процессами.

**Ключевые слова** — автоматическое управление, логико-динамические системы, основанное на правилах программирование.

### Введение

Функционирование многих современных технических систем определяется совокупностями взаимодействующих дискретно-непрерывных процессов. Наглядное представление о таких системах дают большие промышленные автоматизированные установки, такие, например, как воздухоразделительные установки по производству жидкого азота и сжатого кислорода. Технологический процесс в установке зависит от дискретных состояний клапанов, дросселей, вентилей (открыт-закрыт), компрессоров, насосов, детандеров (включен-выключен) и от состояний непрерывных процессов изменения температур, давлений, концентраций и уровней.

Понятие логико-динамической ситуации используется при логико-динамической спецификации гибридных процессов [1–3]. Дискретно-непрерывный процесс с точки зрения динамики является гибридным процессом, т. е. таким, в котором наряду с непрерывными изменениями состояния происходят дискретно-событийные изменения состояния и (или) динамики процесса. Очевидно, чисто непрерывные и чисто дискретно-событийные процессы являются частными случаями гибридных процессов.

Логико-динамическая спецификация процессов ориентирована на их компьютерную реализацию. Текущее состояние гибридных процессов определяется множеством вещественных пере-

менных состояния  $X$ , представляющих непрерывные составляющие, и множеством символьных переменных  $W$ , представляющих дискретные составляющие процессов. Среди этих переменных выделяются подмножества независимых внешних символьных воздействий (команд)  $V$  и непрерывных вещественных воздействий  $X_i$ . В составе множества  $W$  выделяются также подмножество  $Q$ , содержащее переменные для представления дискретных состояний и режимов процессов, и подмножество предикатов от непрерывных состояний  $G$ . Предикаты могут определять состояния дискретных и режимы гибридных процессов. В результате для представления процессов имеем следующие множества переменных:

$$W = V \cup Q \cup G; \quad (1)$$

$$X = X_i \cup X_s, \quad (2)$$

где  $X_s$  — множество переменных состояния непрерывных процессов.

Для спецификации процессов необходимо определить функции транзитивных переходов следующих типов:

$\sigma: W \rightarrow Q$  — функция трансформации состояний дискретно-событийных процессов и режимов гибридных процессов;

$\delta: W \times X \rightarrow X_s$  — функция трансформации непрерывных состояний для возможных режимов гибридных процессов;

$\gamma: X \rightarrow G \rightarrow \{False, True\}$  — зависимость значений предикатов от непрерывных состояний процессов.

Спецификация процессов сводится к заданию конкретных способов определения и вычисления указанных выше функций перехода. Эти способы существенным образом влияют на качество спецификации, которое определяется выразительностью, надежностью и удобством реализации процессов.

Поскольку в общем случае мощности множеств значений дискретных состояний могут быть больше двух, то эти состояния должны представляться символическими переменными. Любую символическую переменную всегда можно представить с помощью формулы нескольких логических переменных. Использование логических переменных и формул для представления дискретных состояний позволяет, на наш взгляд, повысить выразительность спецификаций и эффективность получаемых на их основе исполняющих процедур. Будем все элементы множества  $W$  представлять логическими переменными. В этом случае функцию перехода  $\sigma$  можно представить в следующем виде:

$$\sigma: W \rightarrow Q \times \{False, True\}. \quad (3)$$

Функция  $\gamma$  является логической по определению, ее реализация связана с вычислением неравенств, например, следующего вида:

$$(x_{j_1} \geq (a_k + x_{j_2})) \wedge (x_{j_3} \leq (b_k + x_{j_4})) \rightarrow g_k, \quad (4)$$

где  $x_{j_1}, \dots, x_{j_4} \in X$ ;  $g \in G$ ;  $a_k$  и  $b_k$  — некоторые константы.

Функцию перехода  $\sigma$  задают с помощью совокупности продукционных правил вида «Условие  $\rightarrow$  Действие» [4, 5]. В качестве условия предлагается использовать элементарные конъюнкции логических переменных, а в качестве действия — присвоение требуемых значений определенным дискретным состояниям и режимам процессов. Такие конъюнкции интуитивно понятным образом могут интерпретироваться как динамические ситуации. В общем случае для формализации процесса могут потребоваться произвольные логические формулы. Поэтому правила между собой должны быть соединены логической связкой  $\vee$  (ИЛИ). Это значит, что одно и то же действие может быть произведено при выполнении различных условий, т. е. условия могут образовывать дизъюнкции. Поскольку дизъюнкция элементарных конъюнкций представляет собой дизъюнктивную нормальную форму, то можно утверждать, что таким способом можно в качестве условия действия представить любую логическую формулу. Логико-динамическую

ситуацию можно определить с помощью следующих утверждений:

1) любая логическая переменная (воздействие, состояние, режим, предикат) или ее отрицание является динамической ситуацией;

2) если  $S_i$  и  $S_j$  — динамические ситуации, то  $S_i \wedge S_j$  также является динамической ситуацией;

3) других ситуаций нет.

С учетом этого определение логико-динамической ситуации  $S_j$  можно представить в виде

$$S_j = s_{j_1}, \dots, s_{j_i}, \dots, s_{j_n},$$

где  $s_{j_i} \in W \times \{False, True\}$ ;  $n = 1 \dots N_w$ ,  $N_w = |W|$ .

Таким образом, ситуация представляет собой логическую функцию от логических аргументов. Ситуация имеет значение *True*, если значение каждого из ее аргументов совпадает со значением, указанным в определении ситуации. Если значение хоть одного аргумента отличается от указанного, то ситуация имеет значение *False*.

Представляют интерес две возможные разновидности или два типа правил трансформации ситуаций, различающихся по выразительным возможностям спецификации процессов и по надежности получаемых спецификаций. Условная часть обоих типов правил представляет собой логико-динамическую ситуацию. В исполнительные части правил 1-го типа могут включаться как сами логические состояния, так и их отрицания. Правила этого типа имеют вид

$$S_j \rightarrow q'_{j_1}, \dots, q'_{j_i}, \dots, q'_{j_m}, \quad (5)$$

где  $q'_{j_i} \in Q \times \{False, True\}$ ;  $m = 1, \dots, |Q|$ .

Если ситуация в условной части такого правила имеет значение *True*, то переменным из исполнительной части присваиваются указанные значения (*True* или *False*). В противном случае значения переменных остаются неизменными.

Правила 2-го типа отличаются тем, что в их исполнительную часть помещаются только те переменные, которым необходимо присвоить значения *True*, т. е. эти правила имеют вид

$$S_j \rightarrow q'_{j_1}, \dots, q'_{j_i}, \dots, q'_{j_m}, \quad (6)$$

где  $q'_{j_i} \in Q \times \{True\}$ ;  $m = 1, \dots, |Q|$ .

При использовании этих правил исполняющая процедура их сканирует, запоминая идентификаторы переменных, которым в соответствии с ситуациями должны быть присвоены значения *True*, и в конце каждого шага алгоритма обновления состояния присваивает эти значения соответствующим координатам вектора состояния, а всем остальным координатам присваива-

ется значение *False*. В этом случае исключается возможность появления в базе правил противоречивых правил, однако несколько усложняется присваивание переменным значения *False*. При этом оказывается реализованным принцип, в соответствии с которым «отсутствие» трактуется как «отрицание». Наличие в базе правил нескольких правил, присваивающих одной и той же переменной состояния значения *True*, вполне допустимо и может соответствовать реальному поведению системы.

С каждой ситуацией могут быть связаны два события: одно — с возникновением ситуации, второе — с ее исчезновением. Для реализации событий необходимо запоминать предыдущие значения ситуаций. Логические переменные, представляющие события, могут иметь значение *True* только в течение одного шага цикла обновления состояний. Они могут быть использованы в условных частях трансформационных правил только 1-го типа (5). В этом случае эти правила специфицируют новые события. События могут использоваться в условных частях этих правил совместно с ситуациями. Одни и те же события в разных ситуациях могут специфицировать различные новые события. Использование событий повышает выразительные возможности спецификаций, но усложняет их верификацию.

Для задания функции перехода  $\delta$  необходимо определить алгоритмы вычисления транзитивных отношений между непрерывными состояниями для каждого режима. Предлагается это делать на основе использования передаточных функций элементарных динамических звеньев, функциональных преобразователей и структурных схем их соединения [6]. В этом случае формализация знаний о произвольных непрерывных процессах может быть проведена с помощью алгоритмов вычисления транзитивных отношений для элементарных динамических звеньев. К таким звеньям относятся интеграторы, апериодические, колебательные и дифференцирующие звенья, а также звенья чистого запаздывания [7].

При рассмотрении транзитивных моделей процессов используется соглашение о том, что именем переменной со штрихом обозначается состояние процесса, непосредственно следующее за состоянием, обозначенным этим же именем без штриха. В случае свободного одномерного непрерывного процесса отношение следования, или транзитивное отношение (transitive dependence), представляет собой бинарное отношение на множестве вещественных чисел вида  $\tau(y, y')$  или  $y' = \tau(y)$ , где  $y$  — текущее состояние процесса, а  $y'$  — следующее состояние процесса. Транзитивное отношение  $\tau$  определяется параметрами элементарного динамического звена и длительностью интервала времени  $\Delta t$  между  $y$  и  $y'$ .

В случае вынужденного процесса оно также зависит от внешнего воздействия  $x$ . Для задания зависимости непрерывных состояний от логики развития гибридного процесса алгоритмы вычисления этих отношений включаются в исполнительные части правил, условными частями которых являются требуемые значения соответствующих режимов. В принятых обозначениях эти правила имеют вид

$$S_j \rightarrow x'_k = \tau_k(x_k, x), x_k \in X_s, x \in X, \quad (7)$$

где  $S_j$  — ситуация, определяющая режим;  $\tau_k$  и  $x_k$  — отношение и состояние соответственно;  $x$  — произвольная переменная процесса. В условных частях правил (7) могут также использоваться события, специфицированные правилами (5). Такие правила позволяют специфицировать дискретные изменения состояний гибридных процессов.

Таким образом, знания о развитии процессов могут быть формализованы путем спецификации множеств логических и вещественных состояний (1), (2) и формирования совокупности трансформационных правил (4)–(7).

### О реализации ситуационно-событийного подхода к спецификации взаимодействующих процессов различной динамики

Компьютерная реализация процессов, специфицированных трансформационными правилами (4)–(7), производится путем использования интерпретатора этих правил. Интерпретатор представляет собой программный продукт, который может быть легко реализован с помощью компьютерных средств, использующих алгоритмические языки программирования, например Delphi. Архитектура и алгоритмы функционирования интерпретатора достаточно просты. Множества переменных состояния процессов ( $X$  и  $W$ ) в интерпретаторе представляются массивами записей, каждая из которых содержит имя переменной и ее значение. Все правила реализуются с помощью условных операторов «if...then...». Обе части оператора содержат записи об использованных в данном правиле переменных. Основу интерпретатора составляет исполняющая процедура, которая в цикле сканирует списки правил. Алгоритм обработки правил вычисляет значение условной части правила. В случае правила (5), (6) если это значение равно *True*, то запускается процедура выполнения исполнительной части правила, которая логическим переменным из этой части правила присваивает указанные значения. При этом, естественно, могут измениться значения ситуаций, в определения которых эти пере-

менные входят. В случае правила (7) процедура, указанная в исполнительной части правила, запускается, если значение ситуации или события соответствует значению, указанному в условной части. Эта процедура вычисляет новое значение состояния непрерывного процесса и присваивает его соответствующей переменной.

На каждом шаге цикла обновления состояний процессов производится продвижение времени на интервал  $\Delta t$ . Возможны два режима продвижения времени. В режиме реального времени измеряется длительность каждого цикла, и время увеличивается на измеренное значение. В этом случае быстроедействие соответствует реальным процессам, а точность зависит от производительности компьютера и размерности модели. В режиме модельного времени время продвигается после каждого цикла на заданную пользователем величину. При этом точность определяется величиной заданного приращения времени, а быстроедействие зависит от производительности компьютера и размерности модели. Использование современных персональных компьютеров для реализации моделей, содержащих тысячу правил, позволяет обеспечить величину приращения реального времени, не превышающую нескольких десятков микросекунд, что обеспечивает точность, приемлемую для большинства приложений.

Экспериментальные исследования описываемой технологии проводятся в СПИИРАН с помощью разработанного исследовательского прототипа компьютерной среды EnviCon [8], реализующей описываемый подход. Среда имеет достаточно наглядный интерфейс представления трансформационных правил и средства визуализации процессов. Рассматриваемые ниже примеры спецификации процессов будут представлены в форматах редакторов этой среды. Технологию использования логико-событийного подхода покажем на примере моделирования автоматической системы позиционирования перевернутого маятника.

### Модель позиционирования перевернутого маятника

Маятник представляет собой груз массы  $m_1$ , находящийся на одном конце жесткого невесомого подвеса длины  $L$ . Второй конец подвеса укреплен на шарнире, установленном в центре масс тележки, имеющей массу  $m_2$ . Тележка может двигаться по прямой линии в горизонтальной плоскости. Управление состоянием маятника производится с помощью управляющего воздействия в виде силы, прилагаемой к тележке в том или другом направлении. В исходном состоянии маятник находится в покое. Угол отклонения подвеса от вертикали  $\varphi = 0$ . Задача состоит в том,

чтобы раскачать маятник до амплитуд, близких к  $180^\circ$ , после чего перейти в режим стабилизации перевернутого маятника. Вторая задача управления состоит в перемещении тележки с перевернутым маятником из исходной позиции  $x_0$  в заданную целевую позицию  $x_{ц}$ . Таким образом, необходимо реализовать два режима работы системы: режим раскачивания маятника и режим стабилизации перевернутого маятника и управление положением тележки.

Рассмотрим использование ситуационно-событийного подхода для спецификации комплекса, состоящего из модели маятника на тележке (объекта управления) и модели контроллера, реализующего указанную задачу управления. Логико-динамическая спецификация процессов формируется на основании физической модели процессов. На рис. 1 представлена функциональная схема маятника на тележке.

При отклонении подвеса от вертикали в системе возникают незатухающие колебания. Нас будут интересовать процесс изменения угла отклонения подвеса от вертикали  $\varphi$  и процесс изменения координаты центра тяжести тележки  $x_m$ . Изменения угла происходят вследствие перемещения как груза, так и тележки. Пусть в начальном положении  $\varphi = \varphi_0$  и  $x_m = 0$ . Значения ускорения и скорости груза, и ускорения и скорости тележки, а также значение времени равны нулю. Для спецификации возникающих процессов необходимо на основании начальных значений выполнить следующую последовательность вычислений.

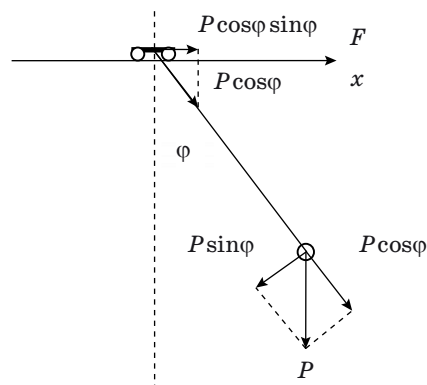
$$\text{Ускорение груза } a_1 = g \sin \varphi;$$

$$\text{скорость груза } V_1' = V_1 + \int_t^{t+\Delta t} a_1 dt;$$

$$\text{угловая скорость от перемещения груза } \omega_1 = V_1/L;$$

$$\text{ускорение тележки } a_2 = (P \cos \varphi \sin \varphi)/m_2;$$

$$\text{скорость тележки } V_2' = V_2 + \int_t^{t+\Delta t} a_2 dt;$$



■ Рис. 1. Маятник на тележке



новое значение координаты тележки  

$$x'_m = x + \int_t^{t+\Delta t} V_2 dt;$$
 угловая скорость от перемещения тележки  

$$\omega_2 = (V_2/L)\cos\varphi;$$
 суммарная угловая скорость  $\omega = \omega_1 + \omega_2;$   
 новое значение угла 
$$\varphi' = \varphi + \int_t^{t+\Delta t} \omega dt.$$

Далее вычисления повторяются для новых значений состояний процессов на каждом обороте цикла обновления состояний процессов. Приведенная совокупность математических зависимостей представляет собой физическую модель данной динамической системы. Поскольку эта совокупность не содержит логических зависимостей, она может быть специфицирована совокупностью только правил (7), в которых в качестве условия использована логическая константа *EverTrue*, равная *True*. Исполнительные части правил практически однозначно соответствуют приведенным вычислениям. На рис. 2 (редактор правил) показана совокупность этих правил, представленных в формате редактора правил среды EnviCon. Выходными переменными модели являются *fi угол* и *X тележки*.

Валидацию этой модели можно произвести путем измерения периода колебаний, возникающих в системе при отклонении груза на небольшой угол. Расчетное значение периода  $T = 2\pi\sqrt{\frac{Lm_2}{g(m_1+m_2)}} = 1,4187$ , а полученное при моделировании  $T = 1,4190$ . При моделировании устанавливались начальное значение  $\varphi_0 = 1^\circ$  и дискрет продвижения времени  $\Delta t = 0,00001$  с.

Рассмотрим модель контроллера. Логическую переменную для представления режима раскачивания маятника назовем *Раскачивание*, а пе-

ременную для стабилизации и управления — *Управление*. В исходном состоянии *Раскачивание = True*, а *Управление = False*. Входными переменными для модели контроллера являются угол  $\varphi$  и координата тележки  $X_m$ . Выходной переменной контроллера является управляющее воздействие *Управ. возд.*

Раскачивание производится путем приложения к тележке постоянной силы  $F$  в положительном или отрицательном направлении. Сила определяет возникновение ускорения и скорости в определенном направлении. Скорость тележки  $V_2$  определяет возникновение дополнительной угловой скорости маятника  $\omega_2 = -(V_2/L)\cos\varphi$ . Для того чтобы происходило увеличение угловой скорости, необходимо, чтобы знак дополнительной угловой скорости совпадал со знаком текущей угловой скорости маятника  $\omega_1$ . С учетом этого и того, что раскачивание наиболее эффективно происходит при малых углах, примем следующий алгоритм раскачивания. В ситуации, когда угловая скорость положительна и угол находится в диапазоне от  $-45^\circ$  до  $+45^\circ$ , прикладываем силу  $F$  в отрицательном направлении (имя ситуации *сила F справа*), а в положительном направлении прикладываем силу (имя ситуации *сила F слева*), когда угол находится в том же диапазоне, но угловая скорость отрицательна. Когда угол выходит за пределы указанного диапазона, имеет смысл тормозить движение тележки, формируя управляющее воздействие в виде силы, пропорциональной скорости и направленной против нее. Логической переменной, представляющей эту ситуацию, присвоим идентификатор *Торможение*. Таким образом, для спецификации процесса раскачивания маятника необходимо определять величину угловой скорости маятника  $\omega$  и величину скорости тележки  $V_m$ . На рис. 3 приведены три правила (6), специфицирующие

Редактор правил трансформации непрерывных состояний, Объект

Исх. ПМ вых. **Логические** Возд. Входы Состояния Предикаты ПМ сост. из модели Объект

№	Переменная	Процедура	Кэф.циент	Аргумент 1	Аргумент 2	Ситуация / событие	имя параметра	значение
1	sin fi	Синус	1.00	fi угол		Ever True		
2	cos fi	Косинус	1.00	fi угол		Ever True		
3	a1 ускор. груза	Умножение	-1.00	g ускор. св. пад.	sin fi	Ever True		
4	V1 скорость груза	Интеграл	1.00	a1 ускор. груза		Ever True		
5	Om1 угл. скор. гр	Деление	57.2958	V1 скорость груза	L длина подвеса	Ever True		
6	P cos fi	Умножение	1.00	P вес	cos fi	Ever True		
7	P cos fi sin fi	Умножение	1.00	P cos fi	sin fi	Ever True		
8	a2 ускор. тел.	Деление	1.00	P cos fi sin fi	m2 масса тел.	Ever True		
9	V2 скорость тел.	Интеграл	1.00	a2 ускор. тел.		Ever True		
10	X тележки	Интеграл	1.00	V2 скорость тел.		Ever True		
11	V2 / L	Деление	1.00	V2 скорость тел.	L длина подвеса	Ever True		
12	Om2 угл. скор. тел	Умножение	-57.2958	V2 / L	cos fi	Ever True		
13	Om угл. скор. сум.	Сумма / разность	1.00	Om1 угл. скор. гр	Om2 угл. скор. тел	Ever True	2-ой коэффициент	1.00
14	fi угол	Интеграл	1.00	Om угл. скор. сум.		Ever True		

■ Рис. 2. Правила спецификации модели маятника на тележке

логику режима раскачивания маятника. В правилах имя переменной, значение которой равно *False*, представлено на черном фоне. При этом необходимо, конечно, с помощью правил (4) специфицировать предикаты  $\Omega > 0$  и  $-45 < \varphi < 45$ .

Для переключения в режим *Управление* используем события, связанные с возникновением ситуаций, определяемых предикатами  $\varphi > 160$  и  $\varphi < -160$ . В данном случае необходимо применять правила (5), допускающие использование событий и непосредственное присваивание ситуациям значения *False*. Эти два правила представлены в редакторе правил (рис. 4).

Входными переменными контроллера в режиме *Управление* являются выходные переменные объекта управления: угол  $\varphi$  и координата тележки  $x_m$ . Выходной переменной контроллера является управляющее воздействие *Упр. возд.* Управляющее воздействие используется как для стабилизации перевернутого маятника, так и для перемещения тележки в целевую позицию.

Исходн.	Сорт. "if"	Сорт. "then"
0. If	Раскачивание	$\Omega > 0$
then	Сила F справа	$-45 < \varphi < 45$
1. If	Раскачивание	<b><math>\Omega &gt; 0</math></b>
then	Сила F слева	$-45 < \varphi < 45$
2. If	Раскачивание	<b><math>-45 &lt; \varphi &lt; 45</math></b>
then	Торможение	

■ Рис. 3. Правила спецификации логики раскачивания маятника

Исходн.	Сорт. "if"	Сорт. "then"
0. If	$\varphi > 160$	
then	Управление	<b>Раскачивание</b>
1. If	$\varphi < -160$	
then	Управление	<b>Раскачивание</b>

■ Рис. 4. Правила переключения режимов управления маятником

Исходными данными для формирования управляющего воздействия являются отклонение угла от 180 ( $-180$ ) градусов  $\Delta \varphi$  и отклонение тележки от целевой координаты  $\Delta x_{\text{тел}}$ . Поскольку к вертикальному положению маятник может подходить как со стороны положительных углов, так и со стороны отрицательных углов, то для вычисления  $\Delta \varphi$  необходимо использовать два правила: одно для  $\varphi > 0$  и другое для  $\varphi < 0$ . В данном случае процесс управления должен иметь две составляющие: одну для стабилизации угла и другую для перемещения тележки в целевую позицию. Для уменьшения негативного влияния этих составляющих друг на друга целесообразно разнести эти процессы по быстродействию. Поскольку управление тележкой допускает меньшее быстродействие, то в алгоритм его формирования введем инерционное (апериодическое) звено с постоянной времени  $T = 0,050$  с. Для формирования обеих составляющих управления будем использовать правила, исполнительные части которых содержат процедуры, реализующие пропорционально интегрально дифференциальные (ПИД) законы управления. В результате для спецификации контроллера получаем совокупность правил (7) (рис. 5).

Первые два правила специфицируют процессы реальных дифференцирующих звеньев. Они на основании входных переменных формируют процессы изменения угловой скорости маятника и скорости движения тележки. Угловая скорость нужна для формирования ситуаций, используемых в режиме *Раскачивание*. Скорость тележки используется в правиле 5 для формирования управляющего воздействия в ситуации *Торможение*. Следующие три правила формируют управляющие воздействия в ситуациях, имеющих место в режиме *Раскачивание*. Правила 6 и 7 вычисляют отклонение угла от вертикали для положительных и отрицательных значений угла. Правила 9 и 12 специфицируют процессы управления, соответствующие ПИД-законам

№	Переменная	Процедура	Коэффициент	Аргумент 1	Аргумент 2	Ситуация / событие	имя параметра	значение	имя параметра	значение
1	$\Omega$ угл. скор.	Производная,Сгл.	1.00	угл $\dot{\varphi}$		Ever True	Пост. времени	0.0001		
2	$\dot{x}$ скор. тел.	Производная,Сгл.	1.00	$\dot{x}$ тележки		Ever True	Пост. времени	0.001		
3	Упр. возд.	Пропорциональная	-1.00	F const		Сила F справа				
4	Упр. возд.	Пропорциональная	1.00	F const		Сила F слева				
5	Упр. возд.	Пропорциональная	-50.00	$\dot{x}$ скор. тел.		Торможение				
6	$\Delta \varphi$	Сумма / разность	1.00	180 const	угл $\varphi$	$\varphi > 0$	2-ой коэффициент	-1.00		
7	$\Delta \varphi$	Сумма / разность	-1.00	180 const	угл $\varphi$	<b><math>\varphi &gt; 0</math></b>	2-ой коэффициент	1.00		
8	$\Delta \varphi$ сгл.	Апериодическая	1.00	$\Delta \varphi$		Управление	Пост. времени	0.0001		
9	Управление $\varphi$	ПИД	5.00	$\Delta \varphi$ сгл		Управление	Интеграл. коэф.	0.00	Дифф. коэф.	2.20
10	$\Delta x$ тел.	Сумма / разность	1.00	$\dot{x}$ тележки	$x$ цели	Управление	2-ой коэффициент	-1.00		
11	$\Delta x$ сглаж.	Апериодическая	1.00	$\Delta x$ тел.		Управление	Пост. времени	0.050		
12	Управление $\dot{x}$ тел.	ПИД	20.00	$\Delta x$ сглаж.		Управление	Интеграл. коэф.	0.00	Дифф. коэф.	40.00
13	Упр. возд.	Сумма / разность	1.00	Управление $\varphi$	Управление $\dot{x}$ тел.	Управление	2-ой коэффициент	1.00		

■ Рис. 5. Правила спецификации процессов управления

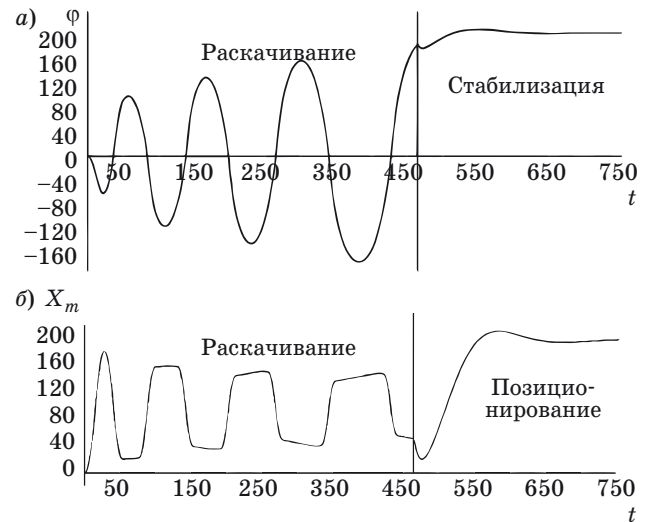
F управления	Насыщение	1.00	Управл. возд.		Ever True
F сумм.	Сумма / разность	1.00	F управления	$P \cos \varphi_i \sin \varphi_i$	Ever True
a2 ускор. тел.	Деление	1.00	F сумм.	m2 масса тел.	Ever True

■ Рис. 6. Правила, специфицирующие взаимодействие объекта с контроллером

управления. Их параметрами являются пропорциональный, интегральный и дифференциальный коэффициенты передачи. Правило 8 сглаживает процесс изменения отклонения по углу. Дело в том, что в правилах, специфицирующих ПИД-управление, использованы идеальные дифференцирующие звенья. Поскольку в компьютере непрерывные процессы представляются ступенчатыми процессами, то эти звенья могут выдавать очень большие пиковые значения, которые даже могут вызывать остановку компьютера. Правила 10 и 11 формируют сглаженное отклонение тележки от целевой позиции. Правило 13 формирует выходное значение управляющего воздействия *Упр. возд.*, представляющего сумму управлений по углу и по положению тележки. Переменная модели контроллера *Упр. возд.* является (соединена с) входной переменной модели маятника *Управл. возд.* Управляющее воздействие контроллера преобразуется исполнительным устройством объекта в силу (переменная *F управления*), приложенную к тележке. Эта сила складывается с силой  $P \cos \varphi \sin \varphi$ , действующей на тележку со стороны груза. Сила, создаваемая исполнительным устройством, пропорциональна управляющему воздействию, но ограничена по величине.

Для спецификации взаимодействия контроллера с объектом модель последнего должна быть расширена. В нее необходимо добавить правило, специфицирующее исполнительное устройство, и правило, суммирующее действующие на тележку силы. Кроме того, необходимо изменить правило 8 (см. рис. 2), вычисляющее ускорение тележки. Дополнительные правила и измененное правило представлены на рис. 6.

Проведенное моделирование подтвердило работоспособность описанных моделей. В ходе моделирования путем изменения параметров алгоритма раскачивания маятника менялась длительность процесса раскачивания. Путем вариации коэффициентов в правилах ПИД-регулирования менялось качество процессов стабилизации перевернутого маятника и позиционирования тележки. На рис. 7, а представлена графическая визуализация процессов среды EnviCon — график изменения угла маятника в режимах раскачивания маятника и стабилизации перевернутого маятника. На рис. 7, б приведен график изменения положения тележки в режиме раскачивания и в режиме позиционирования.



■ Рис. 7. График процесса изменения угла маятника (а) и положения тележки (б)

Графики получены при значениях параметров, представленных в вышеприведенных правилах. Масштабный коэффициент по времени равен 50, масштаб по углу равен 1, а по координате тележки — 100.

### Заключение

Достоинством рассмотренного подхода является возможность достаточно естественно и наглядно специфицировать взаимовлияние дискретных и непрерывных составляющих процессов друг на друга. Это достигается за счет использования в условных частях правил трансформации состояний процессов логико-динамических ситуаций, которые определяются как значениями дискретных состояний, так и предикатами от непрерывных состояний. Наглядность, в частности, обеспечивается использованием логических переменных. Совокупности условных частей правил позволяют формировать дизъюнкции элементарных конъюнкций логических переменных и таким образом специфицировать широкий класс логических функций, описывающих процессы трансформации ситуаций.

Вторым достоинством является то, что трансформационные правила (7) позволяют специфицировать сложные непрерывные процессы в виде схем произвольных соединений элементарных



процессов и различных, в том числе нелинейных, функциональных преобразователей.

Спецификация дискретно-непрерывных процессов с помощью баз трансформационных правил может выполняться непосредственно специ-

алистами в прикладных и проблемных областях, вместе с тем она является пригодной для непосредственной компьютерной реализации процессов.

Работа выполнена в рамках бюджетной темы № 0073-2014-0005 плана НИР СПИИРАН.

## Литература

1. Шпаков В. М. Компьютерная реализация процессов. — Saarbrücken: Palmarium Academic Publishing, 2016. — 252 с.
2. Шпаков В. М. Компьютерная реализация процессов на основе их логико-динамической спецификации // Информационно-управляющие системы. 2016. № 4. С. 68–76. doi:10.15217/issn1684-8853.2016.4.68
3. Alur R., Henzinger T. A., Lafferriere G., Pappas G. J. Discrete Abstractions of Hybrid Systems // Proc. of the IEEE 88. 2000. P. 971–984.
4. Lacoste-Julien S. Hybrid Systems Modeling // Technical Report. McGill University, School of Computer Science. Aug. 2002. <http://msdl.cs.mcgill.ca/people/slacoste/research/report/SummerReport.html> (дата обращения: 30.06.2016).
5. Brownston L., Farrell R., Kant E., Martin N. Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming: Addison-Wesley Series in Artificial Intelligence. — Addison-Wesley, 1985. — 471 p.
6. Howe R. M. Analog Computer Fundamentals // IEEE Contr. Syst. Mag. 2005. Vol. 25. N 3. P. 29–36.
7. Шпаков В. М. Спецификация знаний динамики на основе транзитивной модели непрерывных процессов // Тр. СПИИРАН / под ред. Р. М. Юсупова. СПб.: Наука, 2006. Вып. 3. Т. 1. С. 191–197.
8. Шпаков В. М. Прототип среды моделирования структурированных совокупностей взаимодействующих процессов // Имитационное моделирование. Теория и практика: сб. докл. Второй Всерос. конф., Санкт-Петербург, 19–21 октября 2005 г. Т. II. С. 292–295.

UDC 681.5.017

doi:10.15217/issn1684-8853.2017.2.26

### About Situation-Event Approach to Control of Interacting Discrete-Continuous Processes

Shpakov V. M.<sup>a</sup>, PhD, Tech. Associate Professor, vlad@iias.spb.su

<sup>a</sup>Saint-Petersburg Institute for Informatics and Automation of RAS, 39, 14 Line, V. O., 199178, Saint-Petersburg, Russian Federation

**Introduction:** Control processes depend on the logics and dynamics of the processes in the controlled object, and must meet the requirements of the control logics and dynamics. The challenge is the specification and implementation of the required dependencies. There are certain requirements to the convenience and expressiveness of the specifications and to the efficiency of the control processes obtained on the basis of their software implementations. Logic-dynamic situations used for the specification of hybrid processes are not yet widely used for the implementation of control processes. **Purpose:** Experimental research of technology of control process implementation based on using logic-dynamic situations. **Results:** We have developed a method for the specification of control over discrete-continuous processes based on transformation rules whose conditional parts are logic-dynamic situations. The method has a good expressiveness, providing low cost of the development, transparent specifications and effective computer implementation of the processes. Specification features are illustrated by an example of developing a computer model for a system of automatic control over inverted pendulum positioning. **Practical relevance:** The usage of logic-dynamic situations for control process specification will facilitate the development of control systems for discrete-continuous processes.

**Keywords** — Automatic Control, Logic-dynamic Systems, Rules-based Programming.

## References

1. Shpakov V. M. *Komp'yuternaya realizatsiya protsessov* [Computer Processes Implementation]. Saarbrücken, Palmarium Academic Publishing, 2016. 252 p. (In Russian).
2. Shpakov V. M. Computer Implementation of Processes on the Base of their Logic-Dynamic Specification. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2016, no. 4, pp. 68–76 (In Russian). doi:10.15217/issn1684-8853.2016.4.68
3. Alur R., Henzinger T. A., Lafferriere G., Pappas G. J. Discrete Abstractions of Hybrid Systems. *Proc. of the IEEE* 88, 2000, pp. 971–984.
4. Lacoste-Julien S. *Hybrid Systems Modeling*. Technical Report. McGill University, School of Computer Science, August 2002. Available at: <http://msdl.cs.mcgill.ca/people/slacoste/research/report/SummerReport.html> (accessed 14 December 2016).
5. Brownston L., Farrell R., Kant E., Martin N. *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*. Addison-Wesley Series in Artificial Intelligence. Addison-Wesley, 1985. 471 p.
6. Howe R. M. Analog Computer Fundamentals. *IEEE Contr. Syst. Mag.*, 2005, vol. 25, no. 3, pp. 29–36.
7. Shpakov V. M. Dynamic Knowledge Specification Based on Transitive Model of Continuous Processes. *Trudy SPIIRAN*, 2006, iss. 1, vol. 3, pp. 191–197 (In Russian).
8. Shpakov V. M. An Environment Prototype for Modeling of Structured Sets of Cooperating Processes. *Sbornik dokladov Vtoroi Vserossiiskoi konferentsii "Imitatsionnoe modelirovanie. Teoriia i praktika"* [Proc. 2<sup>nd</sup> All-Russian Conference "Simulation. Theory and Practice"], Saint-Petersburg, 2005, vol. 2, pp. 292–295 (In Russian).