

## МЕТОД ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СЕТЕЙ ПЕТРИ – МАРКОВА

В.В. Котов, Н.А. Котова, Е.В. Ларкин

*Предложен метод имитационного моделирования процесса функционирования системы с использованием сетей Петри – Маркова, основанный на дискретно-событийном подходе. Разработаны структуры данных для программной реализации метода. Описано программное обеспечение имитационного моделирования с использованием сетей Петри – Маркова.*

*Ключевые слова: сеть Петри – Маркова, имитационное моделирование.*

Сеть Петри-Маркова [1-2] представляет собой структурно-параметрическую модель процесса функционирования некоторой системы. Подобная модель позволяет учитывать как временные и вероятностные свойства процесса перехода системы из одного состояния в другое, так и логику срабатывания подобных переходов. Применение аппарата сетей Петри – Маркова позволяет ответить на вопрос о принципиальной достижимости требуемого состояния анализируемой системы при использовании процесса выбранной структуры, и спрогнозировать время достижения указанного состояния. Хотя во многих случаях возможно получение аналитического решения данных задач, зачастую такое решение требует громоздких расчётов и оказывается весьма трудоёмким. Это обуславливает необходимость разработки процедуры численного имитационного моделирования процесса функционирования некоторой системы с использованием сетей Петри-Маркова, которая, будучи реализованной в программном обеспечении, позволит быстро и эффективно оценивать общие временные характеристики процесса функционирования системы, определять частоту реализации отдельных траекторий при многократной реализации процесса функционирования, оценивать частоту попадания в то или иное состояние, вероятности зависаний системы и т.д.

Рассматриваемые в настоящей работе сети Петри – Маркова состоят из элементов четырёх основных типов: позиция, переход, дуга и фишка. Каждый из этих элементов реализуется в программе в виде самостоятельного класса объектов.

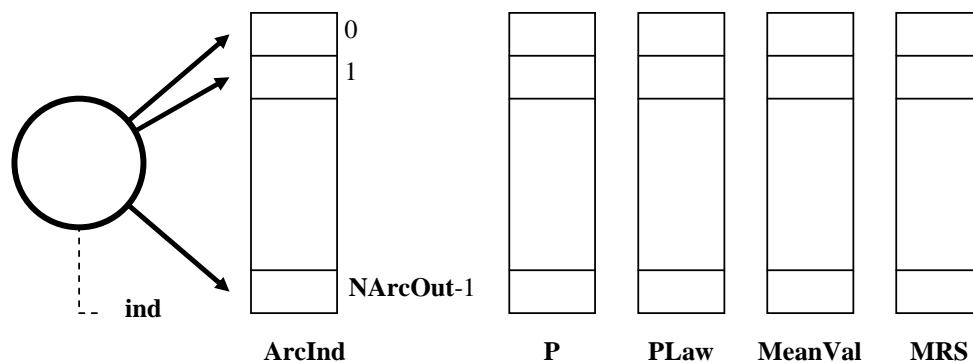
Объект типа «позиция» реализован в виде класса `PMPosition`. Схема данных класса приведена на рис. 1.

Каждый объект класса `PMPosition` имеет собственный уникальный номер, хранящийся в поле `ind` и совпадающий с индексом в глобальном массиве позиций. Основным элементом, описывающим структурные связи

данной позиции с остальной частью сети, является массив `ArcInd`, в котором хранятся индексы переходов, с которыми данная позиция связана выходными дугами.

Количество существующих дуг определяется свойством `NArcOut`.

Вероятности выхода фишки по той или иной дуге в соответствующий переход хранятся в массиве `P`. Его размер совпадает с размером `ArcInd`.



**Рис. 1. Схема данных класса *PMPosition***

В массиве `PLaw` хранится тип закона распределения времени пребывания фишки в позиции, закодированный целым числом. Основные точечные характеристики случайной величины времени пребывания фишки в позиции хранятся в массивах `MeanVal` (математическое ожидание) и `MRS` (среднеквадратическое отклонение). Размеры всех этих массивов также равны `NArcOut`.

Следует обратить внимание на следующие особенности реализации:

1) выходные дуги и связанные с ними переходы не хранятся в позиции, поскольку с точки зрения процедуры моделирования работы сети безразлично, по какой входной дуге в позицию поступила фишка;

2) позиция с индексом «0» рассматривается как начальная позиция, в которой размещаются фишки до начала процедуры имитационного моделирования;

3) позиция, не имеющая выходных дуг, т.е. не связанная по выходу ни с одним переходом, рассматривается как конечная позиция — моделирование останавливается после того, как все фишки собираются в конечной позиции.

Объект типа «переход» реализован в виде класса `PMTransition`. Схема данных класса приведена на рис. 2.

Каждый объект класса `PMTransition` также имеет собственный уникальный номер, хранящийся в поле `ind` и совпадающий с индексом объекта в глобальном массиве позиций.

Структура связей данного перехода с позициями сети описывается парой массивов:

в массиве `ArcInInd`, имеющем размер `NArcIn`, хранятся индексы позиций, которые имеют с переходом общие дуги и связаны с его входами; из этих позиций в переход могут приходить фишки;

в массиве `ArcOutInd`, имеющем размер `NArcOut`, хранятся индексы позиций, которые имеют с переходом общие дуги и связаны с его выходами; в эти позиции из перехода могут уходить фишки.

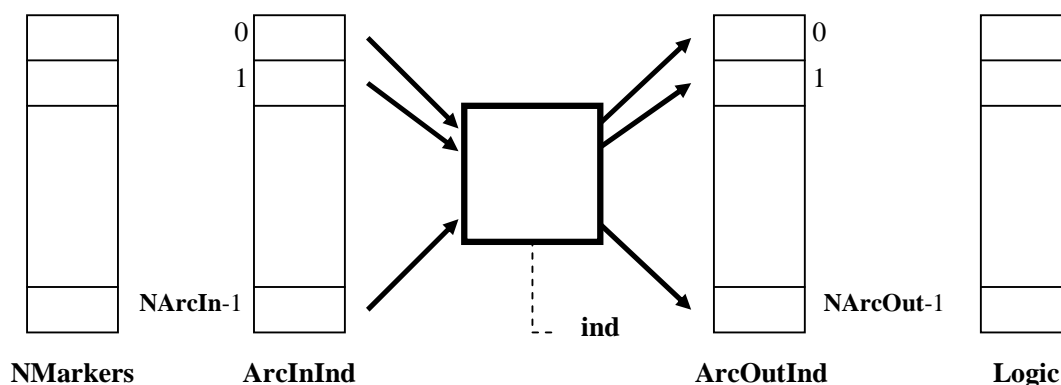


Рис. 2. Схема данных класса *PMTransition*

Массив `Logic`, совпадающий по размеру с массивом `ArcOutInd`, хранит логические условия срабатывания перехода по каждому из имеющихся выходов. Логические условия должны задаваться с дизъюнктивной или (лучше) в совершенной дизъюнктивной нормальной форме.

Вспомогательный массив `NMarkers`, имеющий размер `NArcIn`, совпадающий с массивом входных дуг `ArcInInd`, предназначен для учёта количества фишек, пришедших в переход из соответствующей позиции. Массив используется в процедуре определения условий срабатывания перехода.

Для реализации задачи имитационного моделирования предлагается использовать дискретно-событийный подход, в рамках которого процесс моделирования рассматривается как поток событий, каждое из которых представляет собой перемещение в случайный момент времени одной или нескольких фишек из позиции в переход или из перехода в позицию. Общую последовательность действий имитационного моделирования системы с использованием сетей Петри – Маркова можно представить в виде следующего метода.

1. Обнулить счётчик глобального модельного времени
2. Создать заданное количество фишек в начальной позиции сети
3. Для каждой созданной фишки выполнить шаги 4-6:
4. В соответствии с заданными вероятностями  $P$  выхода из позиции определить случайным образом дугу, по которой фишка выйдет из позиции в переход.

5. Для выбранной дуги в соответствии с заданным законом распределения  $PLaw$  и параметрами математического ожидания  $MeanVal$  и среднеквадратического отклонения  $MRS$  определить случайное время пребывания  $dT$  данной фишки в текущей позиции.

6. Если фишки рассмотрены не все, выбрать следующую и перейти к шагу 4, иначе — к шагу 7.

7. Просмотрев все фишки, определить фишку, имеющую наименьшее время пребывания в позиции  $dT_{min}$  — эта выбранная фишка будет на совершать полушаг из позиции в переход через время  $dT_{min}$ .

8. Уменьшить для всех фишек оставшееся время пребывания в позициях на величину  $dT_{min}$  и увеличить на эту же величину глобальное время моделирования.

9. Вынуть выбранный маркер из соответствующей позиции и по ранее выбранной дуге переместить его в переход.

10. Для перехода, получившего новый маркер, проверить выполнение условий срабатывания перехода.

11. Если ни одно из условий срабатывания не выполнено, переход к шагу 14.

12. Для дуги, для которой обнаружено выполнившееся условие срабатывания перехода, определить связанную позицию.

13. Выполнить второй полушаг, переместив фишки, определившие условие срабатывания, из перехода в связанную позицию.

14. Если существуют фишки, находящиеся в позициях (кроме конечной), вернуться к шагу 7.

15. Если все фишки находятся в конечной позиции — выдать сообщение об успешном окончании моделирования и завершить моделирование.

16. Если все фишки находятся в переходах и конечной позиции — выдать сообщение о «зависании» системы и завершить моделирование.

При практической реализации этой процедуры в программном продукте необходимо предусмотреть сбор статистических сведений (время пребывания маркеров в позициях и переходах, количество полушагов, совершённых каждым маркером и т.п.).

Предлагаемый метод был реализован в программном обеспечении, включающем модуль редактирования сети Петри – Маркова и модуль, имитационного моделирования. Внешний вид основного окна программы приведён на рис. 3.

В процессе имитационного моделирования собирается следующая статистика (рис. 4).

В поле «№» указан номер фишки. Количество строк соответствует количеству фишек, выбранному до начала моделирования.

Столбец  $T\_mod$  показывает общее время пребывания фишки в модели (без учёта времени пребывания в конечной позиции).

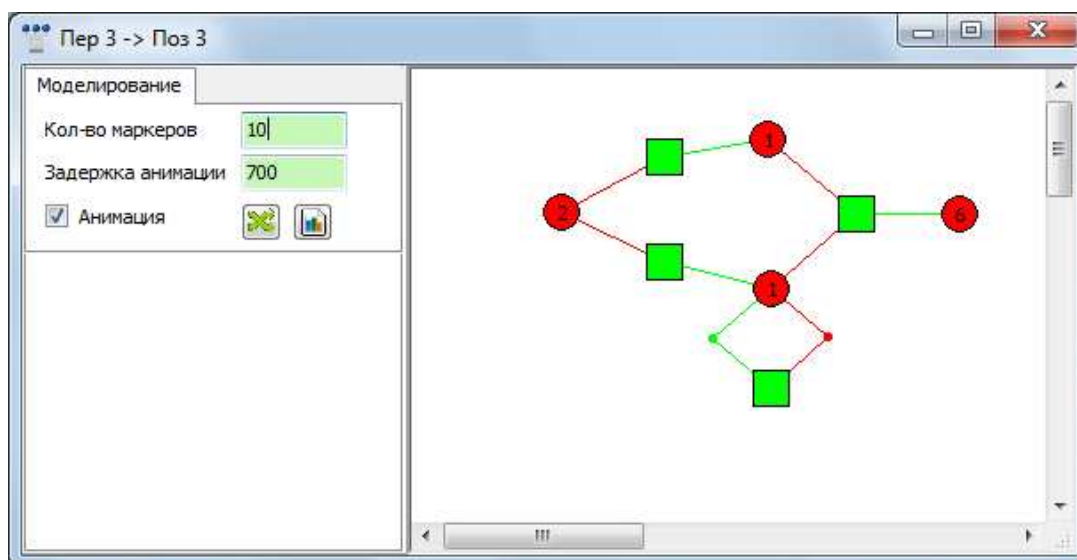
Столбец T\_proc содержит общее время пребывания фишки в позициях (кроме конечной).

Столбец T\_wait показывает общее время пребывания фишки в не-примитивных переходах.

Столбец N\_pos содержит общее число пройденных фишкой позиций.

Столбец T\_per\_pos содержит информацию о среднем времени пребывания фишки в позиции (в пересчёте на одну позицию).

Столбец T\_per\_tra содержит информацию о среднем времени пребывания фишки в переходе.



**Рис. 3. Внешний вид основного окна программы в процессе моделирования**

№	T_mod	T_proc	T_wait	N_pos	T_per_pos	T_per_tra
0	148	148	0	3	49,00	0,00
1	77	77	0	3	25,00	0,00
2	127	127	0	3	42,00	0,00
3	69	69	0	3	23,00	0,00
4	117	117	0	3	39,00	0,00
5	130	130	0	3	43,00	0,00
6	103	103	0	3	34,00	0,00
7	112	112	0	3	37,00	0,00
8	145	145	0	3	48,00	0,00
9	85	85	0	3	28,00	0,00
В среднем:	111,30	111,30	0,00	3,00	37,10	0,00

**Рис. 4. Внешний вид окна статистики моделирования**

По каждой строке можно отследить индивидуальную статистику движения каждой фишки. В последней строке «В среднем» приведена сводная статистика, получающаяся путём усреднения статистик отдельных фишек.

Таким образом, разработанный метод дискретно-событийного имитационного моделирования системы на сети Петри-Маркова, позволяет формализовать описание процесса функционирования анализируемой системы с учётом её структурно-параметрических и вероятностно-временных свойств, оценить достижимость требуемого состояния и время его достижения, а также оценить по формируемой в результате имитационного моделирования статистике траекторий отдельных фишек «узкие» места выбранной структуры и, таким образом, определить направление оптимизации анализируемой системы.

### Список литературы

1. Ларкин Е.В., Котова Н.А. Проектирование информационных систем роботов с использованием сетей Петри-Маркова: учеб. пособие. Тула: Изд-во ТулГУ, 2008. 158 с.

2. К вопросу о моделировании отказоустойчивых систем с помощью сетей Петри-Маркова / Е.В. Ларкин, В.В. Котов, Н.А. Котова, В.А. Соколов // Фундаментальные исследования. №5, 2007. С. 74-78.

*Котов Владислав Викторович, д-р техн. наук, проф., [ykotov@list.ru](mailto:ykotov@list.ru), Россия, Тула, Тульский государственный университет,*

*Котова Наталья Александровна, канд. техн. наук, доцент, [nkotova@inbox.ru](mailto:nkotova@inbox.ru), Россия, Тула, Тульский государственный университет,*

*Ларкин Евгений Васильевич, д-р техн. наук, проф., зав. кафедрой, [elar-kin@mail.ru](mailto:elar-kin@mail.ru), Россия, Тула, Тульский государственный университет*

### *METHOD OF SIMULATION MODELING USING PETRI-MARKOV NETWORKS*

*V.V. Kotov, N.A. Kotova, E.V. Larkin*

*A method of simulation modeling of functioning of a system using Petri-Markov networks based on discrete-event approach is offered. Data structures for software implementation of the method are developed. The developed program for simulation modeling using Petri-Markov network is described.*

*Key words: Petri-Markov network, simulation modeling.*

*Kotov Vladislav Viktorovich, doctor of technical sciences, professor, [ykotov@list.ru](mailto:ykotov@list.ru), Russia, Tula, Tula State University,*

*Kotova Natalia Aleksandrovna, candidate of technical sciences, docent, [nkotova@inbox.ru](mailto:nkotova@inbox.ru), Russia, Tula, Tula State University,*

УДК 681.7

## ПОКООРИНАТНЫЙ ПОИСК МЕСТОПОЛОЖЕНИЯ ТОЧЕЧНОГО ИСТОЧНИКА СИГНАЛА

Е.В. Ларкин, А.А. Аршакян

*На основании сравнительного анализ двух методов поиска экстремума интеграла свертки, показано, что прямой метод поиска дает низкую точность, но обладает повышенным быстродействием. Показано, что совместить быстродействие и точность возможно за счет применения производной от согласованного фильтра при выделении точечного источника. Разработана покоординатная процедура поиска местоположения точечного источника.*

*Ключевые слова: сигнал, свертка, фильтрация, четная функция, нечетная функция, максимум, покоординатный поиск.*

Одной из важных характеристик пеленгатора является точность определения координат точечного источника. Как следует из [1], при применении метода согласованной фильтрации задача оценки местоположения точечного источника сводится к задаче поиска экстремума функции, которая формируется на выходе согласованного фильтра.

Существует множество численных методов поиска экстремума, обладающих различной вычислительной сложностью и точностью, начиная от метода перебора и кончая численными градиентными методами [2]. Указанные методы могут быть разделены на два достаточно обширных класса: методы, связанные с прямым поиском максимума функции  $u(\rho)$ , где  $\rho$  - обобщенная координата, и методы, связанные с поиском нуля производной  $\frac{du(\rho)}{d\rho}$ . Проведем сопоставительный анализ точности, достигаемой при использовании методов данных классов (рис. 1).

Задача решается при условии, что на компьютерную обработку поступает цифровая модель сигнала, т.е. сигнал, прошедший процедуру дискретизации и квантования по уровню [3]. Выбор шага дискретизации по аргументу и квантования по уровню обусловлен в основном максимальной допустимой погрешностью измерения значения сигнала  $u$ . Будем считать, что указанная погрешность определяется интервалом квантования  $\Delta_u$ .