

3. Зайчиков И.В. Фильтрация шумов в бинарном растровом изображении с помощью булевых операций // Известия Тульского государственного университета. Технические науки. 2019. Вып 2. С. 89 – 94.

4. Зайчиков И.В. Оконтуривание объектов в бинарном растровом изображении с помощью булевых операций // Известия Тульского государственного университета. Технические науки. 2019. Вып 10. С. 18 – 21.

Зайчиков Игорь Вячеславович, канд. техн. наук, доцент, zigorwm@mail.ru, Россия, Тула, Тульский государственный университет

PROCESSING A BINARY RASTER IMAGE OBJECTS BY THE METHOD OF NONLINEAR LOCAL FILTER.

I.W. Zaychikov

A simulation of processing binary bitmap images of objects using nonlinear local filtering is performed for use in specialized video processors.

Key words: binary, image, anti-aliasing, filtering, address, response, video processor.

Zaychikov Igor Vyacheslavovich, candidate of technical sciences, docent, zigorwm@mail.ru, Russia, Tula, Tula State University

УДК 004.896:621.865; 681.518

**МОДЕЛИРОВАНИЕ ПРОЦЕДУР ДИСПЕТЧЕРИЗАЦИИ
В СИСТЕМАХ УПРАВЛЕНИЯ РОБОТАМИ С ПОМОЩЬЮ СЕТЕЙ
ПЕТРИ-МАРКОВА**

В.В. Котов, Н.А. Котова, Е.В. Ларкин

Предложен модифицированный метод имитационного моделирования процесса диспетчеризации в системе управления робота с использованием сетей Петри-Маркова, основанный на дискретно-событийном подходе. Разработано программное обеспечение имитационного моделирования, реализующее предложенный метод.

Ключевые слова: сеть Петри-Маркова, имитационное моделирование, система управления, диспетчеризация, роботизированная платформа.

Особенностью роботизированных платформ является их многофункциональность. При централизованном управлении различными функциями возникает задача построения общей модели состояний. В современных роботизированных платформах циклограмма реализуется в виде программного кода бортовой ЭВМ. При управлении отдельными устройствами программный продукт, реализующий циклограмму, естественным образом разделяется на ряд программных модулей, каждый

из которых осуществляет свою функцию в общей задаче управления. Наличие подобного разделения позволяет вести автономную промышленную разработку и отработку программных модулей, включаемых в пакет, на основе универсальных инструментальных средств.

Программные модули имеют унифицированную структуру, что позволяет строить диспетчерские программы опроса модулей [1]. Организация очередей задач определяется дисциплиной диспетчеризации, которая определяет их порядок формирования и обслуживания. В роботизированных платформах с однопроцессорной бортовой ЭВМ как программа диспетчеризации, так и прикладные программы интерпретируются одним и тем же процессором, поэтому для повышения эффективности его использования алгоритм диспетчеризации и дисциплина обслуживания очереди должны быть максимально простыми. Именно по этой причине из трех возможных типов диспетчеризации: динамической, циклической и приоритетной – роботизированных платформах рекомендуется использовать две последних.

M-L-параллельные полумарковские процессы с синхронизацией, применяемые, в том числе, в задачах разработки диспетчеров различных типов, могут быть сведены к Петри-Марковским моделям [2]. Однако непосредственное применение ранее разработанной среды моделирования сетей Петри-Маркова оказывается затруднительным, что обуславливает целесообразность введения дополнительных структур данных, соответствующих диспетчерам различных типов.

Разобьем взаимодействующие элементы при решении задачи диспетчеризации на $L+1$ группу: непосредственно диспетчер и L параллельно функционирующих программных модулей, осуществляющих соответствующие функции в общей задаче управления роботизированной платформой. Действия в процессе решения задачи управления каждой группы и их пересечение во времени в виде сети Петри-Маркова могут быть представлены следующим образом (рис. 1).

Основными элементами сети здесь выступают диспетчер (обозначен индексом 9) и L блоков, моделирующих работу соответствующих программных модулей роботизированной платформы (индекс 1, 2 и 3 на рисунке). Каждый из перечисленных блоков относится к собственной подсети. Таким образом, общая сеть Петри-Маркова состоит из $L+1$ подсети, в каждой из которых содержится собственный маркер, позволяющий конкретизировать состояние подсети в произвольный момент времени.

В рассматриваемой задаче диспетчеризации параллельных процессов подсети программных модулей не взаимодействуют между собой, а работают независимо друг от друга, взаимодействуя только с диспетчером. Для реализации подобного взаимодействия каждая подсеть содержит по два непримитивных двухвходовых перехода и по одному дополнительному блоку синхронизаторов, обозначенных на рисунке как синхронизаторы подсети (индексы 4, 5 и 6). В подсеть, к которой относится диспетчер, входят также L дополнительных блоков синхронизаторов диспетчера (индексы 7, 8

и 9). Для повышения наглядности элементы подсети диспетчера выделены на рисунке утолщёнными линиями, а подсети, описывающие программные модули роботизированной платформы – штриховыми линиями.

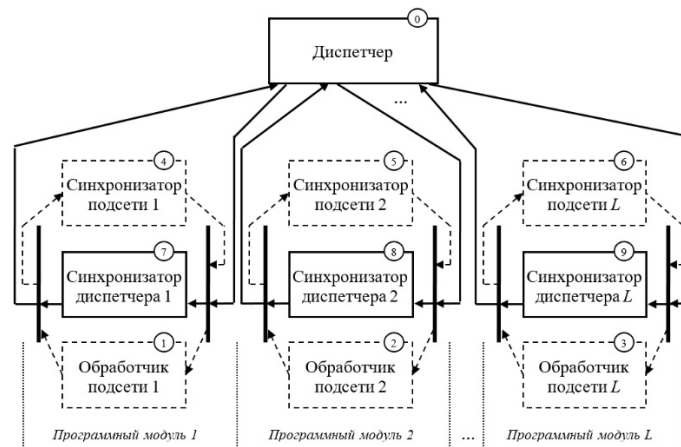


Рис. 1. Пример сети Петри-Маркова в задаче диспетчеризации

Синхронизаторы подсетей являются начальными позициями в своих подсетях, то есть содержат собственный маркер в начальный момент моделирования. Аналогично блок диспетчера является начальной позицией подсети, реализующей механизм диспетчеризации, и также содержит собственный маркер в начале моделирования.

Позиции синхронизаторов диспетчера, служат для соблюдения формальных ограничений, налагаемых аппаратом сетей Петри-Маркова, и настраиваются таким образом, чтобы не вносить дополнительных задержек в перемещение маркера синхронизации. Позиции синхронизаторов подсетей, в зависимости от требований к моделируемой системе, могут иметь как нулевую, так и ненулевую, в том числе случайную, задержку в передаче маркеров соответствующих подсетей.

Общая логика алгоритма работы рассматриваемой сети следующая.

В начальный момент времени моделирования маркеры подсетей программных модулей перемещаются из блоков синхронизаторов подсетей во входные непримитивные переходы (показанные на рисунке справа от блоков обработчиков).

Диспетчер, в зависимости от выбранной дисциплины диспетчеризации, выдаёт свой маркер по одной из исходящих дуг в переход соответствующего программного модуля. В случае циклической диспетчеризации первым модулем выбирается модуль, описываемый подсетью 1. При приоритетной диспетчеризации выбирается модуль, имеющий наибольший приоритет. В случае с квазистохастической диспетчеризацией выбор модуля осуществляется псевдослучайным образом, в соответствии с априорными сведениями о характере распределения вероятностей вызова программных модулей.

Входной переход, в котором собрались оба маркера – собственный маркер программного модуля и маркер диспетчера – срабатывает. Собственный маркер переходит блок обработчика подсети, имитируя процесс работы

соответствующего программного модуля, а маркер диспетчера с нулевой задержкой через позицию соответствующего синхронизатора диспетчера переходит в выходной переход (показанный на рисунке слева от блока обработчика).

Через время, определяемое параметрами блока обработчика подсети, в этот же переход попадает собственный маркер подсети. Это вызывает срабатывание выходного перехода и передачу маркера диспетчера в блок диспетчера, а маркера подсети – вновь в блок синхронизатора подсети.

Далее цикл повторяется для другого программного модуля, в соответствии с выбранной дисциплиной диспетчеризации.

Предложенный метод моделирования процессов диспетчеризации реализован путём модификации алгоритма имитационного моделирования [3], представляющего собой вычисление последовательности времён наступления событий типа «перемещение маркера» с учётом структурно-параметрических свойств сети, описывающей моделируемую систему. Модификация затрагивает логику выбора выходной дуги из позиций особого типа, соответствующих циклическому, квазистохастическому и приоритетному диспетчерам.

Перед началом моделирования для каждого маркера вычисляется время пребывания в позиции, а затем на каждом шаге по всем маркерам вычисляется ближайшее по времени перемещение маркера в позицию, и переключение глобального времени моделирования на соответствующую дискретную величину.

Алгоритм. Моделирование процессов диспетчеризации программного кода роботизированных платформ

1. Обнулить счётчик глобального модельного времени.
2. Для всех подсетей выявить начальные позиции и сформировать соответствующий маркер в каждой из них.
3. Для каждого созданного маркера выполнить шаги 4-6.
4. В зависимости от типа маркера:
 - 4-а. Для подсетей с обычными начальными позициями определить случайным образом дугу, по которой маркер выйдет из позиции в переход, с учётом заданного распределения вероятностей P выхода из позиции;
 - 4-б. Для подсети, содержащей позицию-диспетчер, определить дугу, по которой маркер выйдет из позиции в переход, с учётом дисциплины диспетчеризации, соответствующей типу моделируемого диспетчера:
 - если используется квазистохастический диспетчер – определить дугу случайным образом, с учётом заданного распределения вероятностей P выхода из позиции;
 - если используется циклический диспетчер – выполнить инкремент по модулю «количество исходящих дуг» поля «предыдущая исходящая дуга», и направить маркер по дуге с получившимся индексом;
 - если используется приоритетный диспетчер – осуществить перебор всех исходящих дуг, начиная с нулевого индекса; для каждой дуги просматривать сопряжённый с ней переход; если обнаруживается переход, в котором присутствует один или несколько маркеров из подсетей, описывающих

соответствующие программные модули роботизированной платформы, то прекратить перебор и осуществить передачу маркера диспетчера в этот переход, как имеющий наивысший приоритет; если ни в одном переходе, сопряжённом дугами с позицией-диспетчером, не находится ни одного маркера, маркер-диспетчер передать в переход с нулевым индексом.

5. Для выбранной дуги рассчитать случайное время пребывания dT данного маркера в текущей позиции до его выдачи в соответствующий переход; параметры случайной величины времени пребывания определяются заданным законом распределения с установленными матожиданием и среднеквадратическим отклонением, соответствующими выбранной дуге.

6. Повторять шаги, начиная с шага 4, для оставшихся маркеров; после исчерпания списка маркеров перейти к шагу 7.

7. Из всех маркеров определить тот, который имеет наименьшее время пребывания в позиции dT_{min} – зафиксировать индекс выбранного маркера для последующего выполнения полушага (перемещения из позиции в переход).

8. Скорректировать оставшееся время пребывания в позициях для всех маркеров на величину dT_{min} ; увеличить на эту же величину глобальное время моделирования $GTime$.

9. Если глобальное время $GTime$ превысило заданное максимальное время моделирования, остановить работу алгоритма (выход по превышению предельного времени моделирования).

10. Переместить маркер, выбранный на шаге 7, из соответствующей позиции в переход по ранее выбранной дуге переместить.

11. Проверить выполнение условия срабатывания данного перехода после появления в нём нового маркера.

12. Если условие срабатывания не выполнено, переход к шагу 18.

13. Выбрать все маркеры в рассматриваемом переходе (по умолчанию полагается, что для каждой входной дуги в переходе должен присутствовать один маркер; в сумме они определяют условие срабатывания перехода).

14. Для каждого маркера из анализируемого перехода выполнить второй полушаг в связанную выходной дугой позицию; при выдаче маркеров из перехода в позицию должно выполняться требование соответствия индексов подсетей.

15. Для каждого маркера, задействованного в процедуре срабатывания перехода, выполнить действия, аналогичные описанным в шагах 4-5.

16. Рассчитать статистику (время пребывания маркера в позиции (с точки зрения маркера и с точки зрения позиции), время ожидания маркера в переходе (с точки зрения маркера и с точки зрения перехода), количество маркера попаданий в переход, количество попаданий маркера в позицию и т.п.).

17. Если маркер достиг конечной позиции и условием завершения является достижение хотя бы одним маркером конечной позиции, то выдать сообщение об успешном окончании моделирования и завершить моделирование, иначе – переход к шагу 18.

18. Если существуют маркеры, находящиеся в позициях, отличных от конечной, то вернуться к шагу 7, иначе – переход к шагу 19.

19. Если все маркеры находятся в конечной позиции – выдать сообщение об успешном окончании моделирования и завершить моделирование, иначе – переход к шагу 20.

20. Выдать сообщение о «зависании» системы и завершить моделирование.

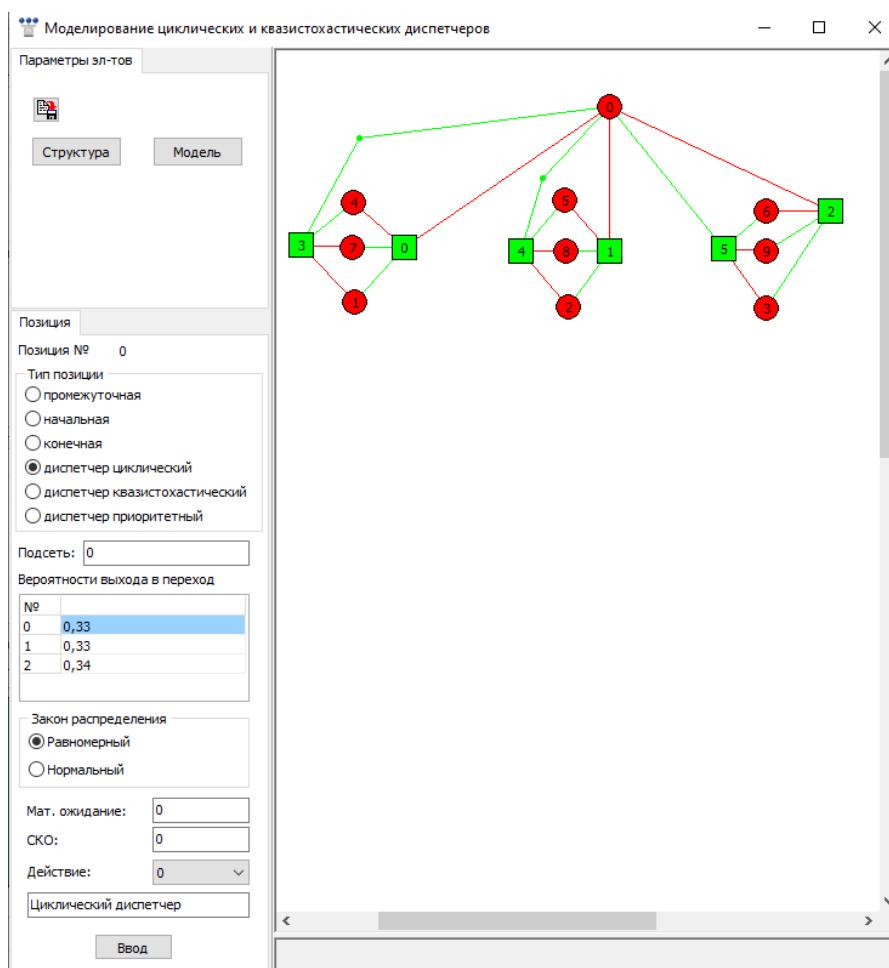


Рис. 2. Внешний вид программного обеспечения с загруженным примером модели циклического диспетчера

Для экспериментальных исследований модели общего вида, приведённой на рис. 1, были синтезированы с использованием модифицированного программного обеспечения ряд тестовых файлов-описаний СПМ, соответствующих диспетчерам различных типов. При загрузке в программу данная сеть, структурно состоящая из подсети диспетчера и трёх подсетей программных модулей роботизированной платформы, имеет вид, показанный на рис. 2.

Таким образом, разработанное алгоритмическое и программное обеспечение позволяет, используя математический аппарат полумарковских процессов с синхронизацией и сетей Петри-Маркова, осуществлять имитационное моделирование циклического и квазистохастического диспетчеров

программного кода роботизированных платформ различного назначения, и решать задачи оценки временных характеристик, которыми будут обладать системы с различными дисциплинами диспетчеризации.

Список литературы

1. Larkin E.V., Kotov V.V., Ivutin A.N., Privalov A.N. Dispatching of Robotic Control Programs // Journal of Computational and Engineering Mathematics. Chelyabinsk, 2016. Vol. 3. No. 2. P. 14 – 24.
2. Larkin E.V., Kotov V.V., Kotova N.A. Estimation of time intervals during mobile robot digital control // 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, 2017.
3. Котов В.В., Котова Н.А., Ларкин Е.В. Метод имитационного моделирования систем с использованием сетей Петри-Маркова // Известия Тульского государственного университета. Технические науки. 2015. Вып. 9. С. 164 – 169.

Котов Владислав Викторович, д-р техн. наук, профессор, vkotov@list.ru, Россия, Тула, Тульский государственный университет,

Котова Наталья Александровна, канд. техн. наук, доцент, nkotova@inbox.ru, Россия, Тула, Тульский государственный университет,

Ларкин Евгений Васильевич, д-р техн. наук, профессор, заведующий кафедрой, elarkin@mail.ru, Россия, Тула, Тульский государственный университет

SIMULATION OF DISPATCHING IN ROBOT CONTROL SYSTEMS USING PETRI-MARKOV NETWORKS

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

A modified method of simulation of dispatching process in a robot control system of a system using Petri-Markov networks based on discrete-event approach is offered. Simulation software that implements the proposed method was developed.

Key words: Petri-Markov network, simulation, control system, dispatching, robotic platform.

Kotov Vladislav Viktorovich, doctor of technical sciences, professor, vkotov@list.ru, Russia, Tula, Tula State University,

Kotova Natalia Aleksandrovna, candidate of technical sciences, docent, nkotova@inbox.ru, Russia, Tula, Tula State University,

Larkin Evgeny Vasilievich, doctor of technical sciences, professor, manager of department, elarkin@mail.ru, Russia, Tula, Tula State University