

Т. А. Голубева; В. М. Дубовой, д. т. н., проф.

МОДЕЛЬ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КАК ДИНАМИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ

Рассмотрен вопрос планирования производственных процессов разработки программного обеспечения с точки зрения неопределенности ряда параметров и в условиях влияния человеческого фактора. В результате этого разработана модель динамики системы управления производственными процессами разработки программного обеспечения.

Ключевые слова: производственный процесс разработки программного обеспечения, планирование, модель динамики.

Введение

За последние годы разработка программного обеспечения превратилась из искусства отдельных личностей в производственный процесс, в котором заняты миллионы специалистов по всему свету. Но производственные процессы разработки программного обеспечения (ППРПО) имеют свою специфику, которая заключается в том, что необходимо планировать умственный и, в какой-то мере, творческий труд. Кроме того, такие процессы на данное время полностью выполняют люди и их автоматизация пока еще маловероятна. Но невзирая на это, планирование и прогнозирование хода ППРПО крайне необходимо для правильного управления ресурсами процесса и является **актуальной научной проблемой**.

Планирование и прогнозирование ППРПО является стандартизированным [1 – 3]. Реализованный ряд систем планирования ППРПО, например [4, 5]. Однако, современные системы планирования ППРПО не учитывают неопределенность параметров и характер отдельных составляющих процессов, которые протекают во время ППРПО. В частности, не учитывается влияние управления ресурсами во время выполнения ППРПО, которое превращает ППРПО в замкнутую динамическую систему. В результате возникает **задача** разработки модели системы управления производственными процессами разработки программного обеспечения как замкнутой динамической системы, в условиях неопределенности.

Решение задачи

Пусть есть множество ресурсов $H = \{H_1, H_2, \dots, H_n\}$ и множество заданий $Z = \{Z_1, Z_2, \dots, Z_m\}$. Ресурсы можно объединить в подмножества множества H в соответствии с заданиями. Для выполнения одного задания могут быть использованы лишь те ресурсы, которые могут реализовывать это задание.

В проанализированной литературе [1 – 5] не найдены ссылки на то, что задания (tasks) являются разными по своим свойствам. Однако у ППРПО все большее внимание уделяют тому факту, что процессы, протекающие в ходе выполнения работ, отличаются от запланированных в силу ряда причин. Как правило, отклонения от запланированного хода выполнения заданий обуславливаются тем, что существуют рискованные события. Если такое рискованное событие наступает, то возникает необходимость выполнения работы по ее устранению. В результате к запланированному множеству заданий добавляется множество работ по устранению последствий рискованных событий (УПРСН) $R = \{R_1, R_2, \dots, R_k\}$. При этом следует заметить, что в этом случае, как правило, количество ресурсов не увеличивается, что приводит к необходимости перепланировки деления ресурсов уже на стадии выполнения работ. Отметим, что работы УПРСН отличаются от заданий тем, что задание обязательно должно быть выполнено, а каждая работа

УПРСН характеризуется определенной вероятностью того, что соответствующее рискованное событие наступит.

Для упрощения охарактеризуем ППРПО следующими утверждениями:

- процесс состоит из фазы планирования и фазы выполнения работ;
- процесс имеет запланированные дату начала (ДНР) $D_{ДНР}$ и конца работ (ДКР) $D_{ДКР}$;
- считается, что к фазе планирования все проблемы решены, потому они не включаются в процесс планирования (наличие проблем и необходимость их решения в ходе процесса значительно усложняет модель);
- задания между собой не связаны;
- количества ресурсов и заданий является фиксированными величинами;
- одно задание выполняет один исполнитель (если задание должно выполнять несколько исполнителей, на стадии планирования это задание может быть разбито на подзадания, каждое из которых будет выполнять один исполнитель);
- работа УПРСН не такая, которая может остановить производственные процессы на длительное время.

В соответствии со сделанными обозначениями, система управления ППРПО будет выглядеть так, как показано на рис. 1. Данная схема построена с учетом приведенных утверждений. На схеме показано, что сначала руководитель работ (РР) определяется с заданиями, работами УПРСН и имеющимися ресурсами. При этом РР распределяет ресурсы на подмножества по заданиям. Далее он оценивает необходимое время для выполнения каждого из заданий и количество имеющихся человеко-часов для решения каждого задания. После этого ресурсы распределяются между заданиями и работами УПРСН. Далее происходит оптимизация разработанного плана с использованием автоматизированных систем управления предприятием и процессами. На этом фаза планирования завершается и РР получает расписание процесса. Потом процесс переходит в фазу выполнения процесса. На этапе выполнения процесса происходит внесение в разработанное расписание уже отработанных часов по каждому из заданий и работ УПРСН. После каждого такого внесения РР получает обновленное расписание и оценивает по нему состояние процесса. Если процесс не завершен и не остановлен, то необходимо повторно оптимизировать расписание. При этом определенные ресурсы могут заново перераспределяться между задачами. После этого происходит опять внесение в разработанное расписание уже отработанных часов по каждому из заданий и работ УПРСН и оптимизация обновленного расписания до тех пор, пока процесс по какой-то из причин не завершится (завершение процесса не обязательно означает полное выполнение заданий).

На рис. 1 пунктиром выделен адаптивный контур системы распределения ресурсов при разработке программного обеспечения. Фактически этот контур является описанной в предыдущем абзаце фазой выполнения работ.

Особенностью процессов разработки программного обеспечения является то, что все операции по разработке выполняют люди. Произвольная операция, где используется человеческий труд характеризуется тем, что время на ее выполнение задается приблизительно руководителем работ и достаточно часто со значительным запасом. Это приводит к неэффективному использованию человеческих ресурсов. Более целесообразным представляется определение промежутка времени за который может быть выполнена данная работа. Такое представление позволит программе указывать какое время будет оптимальным с точки зрения моделирования. Это позволит менеджеру пересмотреть продолжительность некоторых работ для того, чтобы оптимизировать деление ресурсов. На данное время эта работа выполняется самим менеджером, базируясь на опыте и знании каждого отдельного исполнителя. Поскольку промежуток на выполнение конкретно взятого задания определяется экспертно руководителем процесса, то целесообразным является приложение нечеткой логики для описания времени на выполнение задания [6].

Но представление с помощью математического аппарата нечеткой логики является неполным. Такие оценки в некоторых случаях можно уточнить с помощью использования опыта пре-

дыдущих процессов и уже выполненной части работы по данному процессу. Это может помочь в случаях повторяемых заданий на протяжении процесса. К ним принадлежит большинство заданий тестирования программного обеспечения. Особенностью планирования процесса тестирования является то, что необходимо выполнять идентичное задание несколько раз. При этом каждый раз на этот проход будет потрачен одним и тем же человеком разное время. Актуальным является использование базы данных с тем, чтобы использовать предыдущую информацию для последующего планирования. В этом случае мы получаем модель, где одни часовые параметры заданы менеджером, а другие — просчитаны системой и заданы стохастически (рис. 1). При этом можно оценивать как приблизительное необходимое время на выполнение задания, так и оценивать время конкретного исполнителя, затраченное на выполнение определенного задания.

На рис. 1 показано, что в ходе планирования и выполнения процесса происходит как внесение данных в базу знаний, так и использование уже накопленных знаний для планирования.

Задачи могут выполняться параллельно, то есть, если нет ограничения на выполнение задания в определенной последовательности, то несколько исполнителей могут одновременно выполнять несколько заданий, но при этом один исполнитель одновременно выполняет лишь одно задание.

На рис. 1 показана схема системы управления производственным процессом разработки программного обеспечения.

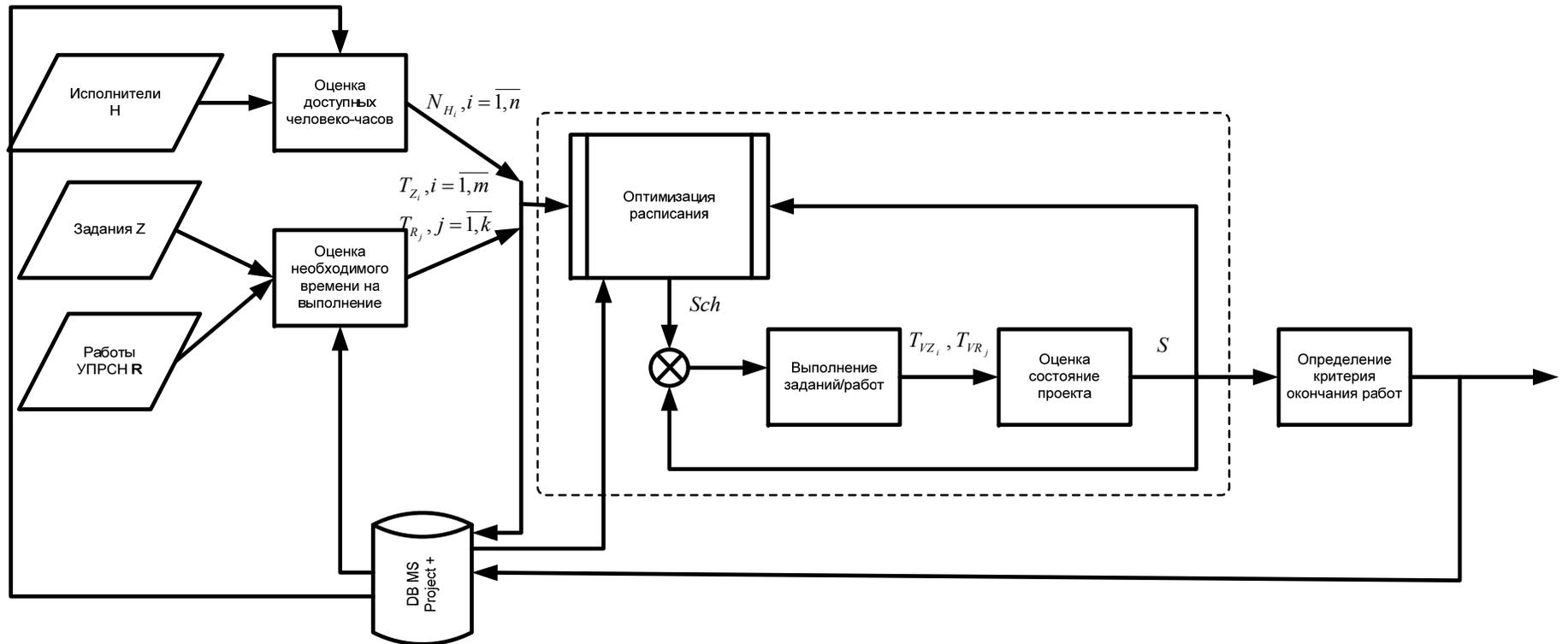


Рис. 1. Схема системы управления производственным процессом разработки программного обеспечения

Прогноз времени выполнения процесса определяется динамикой замкнутого контура, выделенного на рис. 1 пунктирной линией. Рассмотрим эту часть системы детальнее. Детализированная схема изображена на рис. 2.

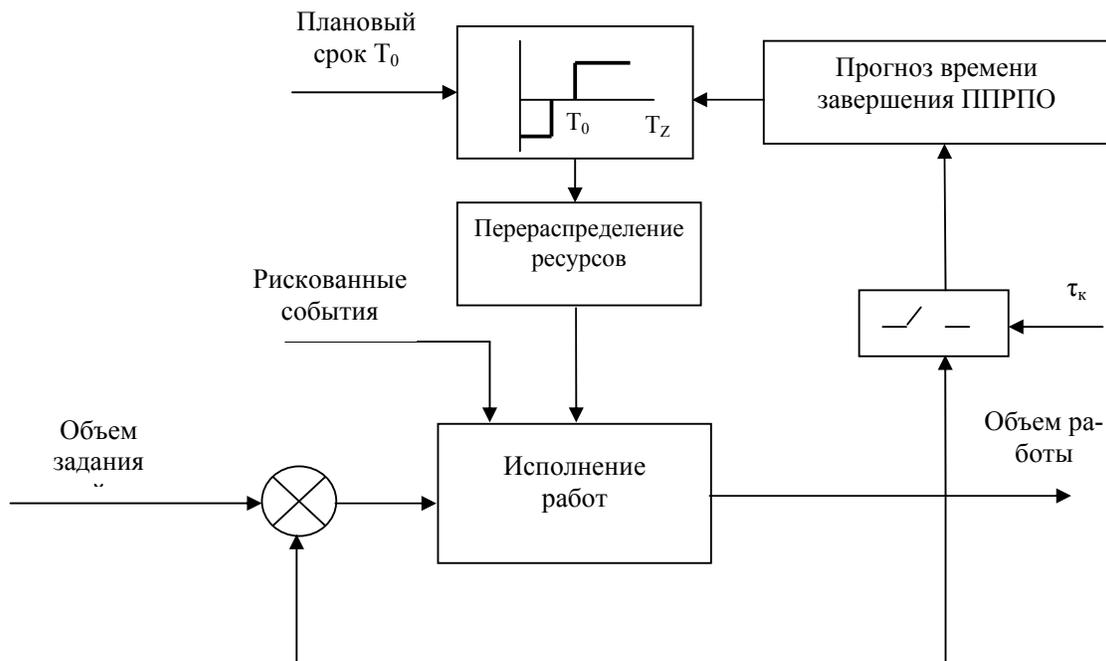


Рис. 2. ППРПО как замкнутая динамическая система

Рассмотрим модели динамики каждого звена системы.

Модель динамики звена «Выполнения заданий» может быть представлена графически фазовым портретом, изображенным на рис.3.

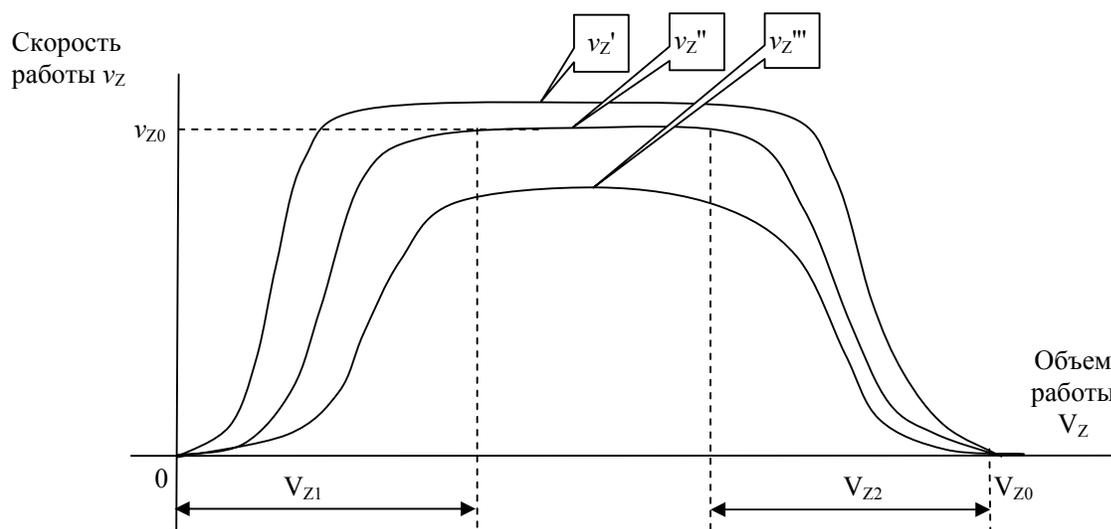


Рис. 3. Фазовый портрет звена выполнения работ

Скорость выполнения задания может быть определена как функция от множества факторов

$$v_{Z_i} = f(V_Z, V_{Z0}, Sk_{Z_i}, Tp_{Z_i}, \{K_{E_j}, Mt_{E_j}, Rs_{E_j}, HV_{E_j}, SO_{E_j}, Hl_{E_j}, PS_{E_j}\} Cm_{Z_i}), \quad (1)$$

где V_Z – выполненный объем задания; V_{Z0} – общий объем задания; K_E – квалификация исполнителя; Sk_{Z_i} – сложность задания, Tp_{Z_i} – вид задания, Mt_{Z_i} – мотивация исполнителя, Rs_{Z_i} – ответственность исполнителя, Cm_{Z_i} – взаимодействие в команде, HV_{Z_i} – рабочее оборудование, SO_{Z_i} – самоорганизация исполнителя, Hl_{Z_i} – общее состояние исполнителя, PS_{Z_i} – способность обучаться. Фазовый портрет построен в координатах $v_Z(V_Z)$. Остальные факторы определяют форму фазовой траектории, которая характеризуется тремя параметрами: длиной начального участка V_{Z1} и соответствующим временем настройки исполнителя на задание, максимальной скоростью выполнения работы v_{Z0} , длиной конечного участка V_{Z2} и соответствующим временем завершения работы, которое в свою очередь зависит от количества ошибок, сделанных во время выполнения задания.

Общее время выполнения работы можно оценить на основе интегрирования обратной фазовой траектории

$$T_Z = \int_0^{V_{Z0}} \frac{dV_Z}{v_Z}. \quad (2)$$

Каждая работа УПРСН характеризуется некоторой вероятностью p_{R_j} того, что ее необходимо будет выполнять. Даже, если рискованное событие наступит, работа УПРСН также может иметь приоритет выполнения $w_{R_j} \in [0;1]$.

В итоге получим взвешенные оценки времени выполнения задания и работ УПРСН:

$$T_{-zv_{Z_i}} = w_{Z_i} T_{Z_i}; \quad (3)$$

$$T_{-zv_{R_j}} = w_{R_j} p_{R_j} T_{R_j}. \quad (4)$$

Второй составляющей модели динамики является модель контроля состояния выполнения работы, которая характеризуется периодичностью τ_k и погрешностью ΔV_Z . Поскольку контроль осуществляется руководителем-экспертом, то оценка погрешности является нечеткой. Периодичность контроля зависит от приоритета работы с учетом ограничений: не чаще, чем раз в день и не реже, чем раз в неделю.

Третьим элементом модели динамики является прогноз срока завершения работы T . Прогноз осуществляется на основе зависимости (1) и формулы (2). Поскольку зависимость (1) является нечеткой, то и интеграл (2) является нечетким [8].

Четвертой составляющей модели динамики является элемент сравнения прогнозируемого срока завершения работы с плановым сроком $T_0 = \{T_{0min}, T_{0max}\}$ и принятие решения относительно необходимости перераспределения ресурсов.

Прогнозы срока завершения работы и принятия решения по перераспределению ресурсов являются формальными процедурами, временем выполнения которых можно пренебречь.

Перераспределение ресурсов является поисково-оптимизационной задачей с ограничениями. Он осуществляется на основе базовой модели (1), (2). Ограничения на перераспределение ресурсов касаются как самих ресурсов, так и времени перераспределения, и зависят от приоритета задачи — чем выше приоритет, тем менее жесткие ограничения. Поскольку ресурсы ППРПО являются дискретной величиной, то перераспределение ресурсов осуществляется на основе многошаговой стратегии, пример которой изображен на рис. 4.

Таким образом общая модель динамики подается системой уравнений:

$$v_{Z_i} = f(V_Z, V_{Z0}, Sk_{Z_i}, Tp_{Z_i}, \{K_{E_j}, Mt_{E_j}, Rs_{E_j}, HV_{E_j}, SO_{E_j}, Hl_{E_j}, PS_{E_j}\} Cm_{Z_i});$$

$$T_Z = \sum_{k=1}^m \int_{V_{Zk\text{ноч}}}^{V_{Zk\text{кин}}} \frac{dV_Z}{v_{Zk}};$$

$$m < (T_Z - t) / \tau_k; \tag{5}$$

$$u = \begin{cases} +1, & T_Z > T_{0\text{max}}; \\ 0, & T_{0\text{min}} \leq T_Z \leq T_{0\text{max}}; \\ -1, & T_Z < T_{0\text{min}}; \end{cases}$$

$$(m, [V_{Zk\text{ноч}}, V_{Zk\text{кин}}], \{E_k\}): \min |T_Z - T_0|.$$

где m – количество шагов стратегии деления ресурсов.

Система уравнений (5) является нечеткой и жесткой, а отдельные ее параметры задаются статистически. Для ее решения с целью нахождения прогноза времени выполнения работ необходимо приложение специальных методов и алгоритмов [8 – 11].

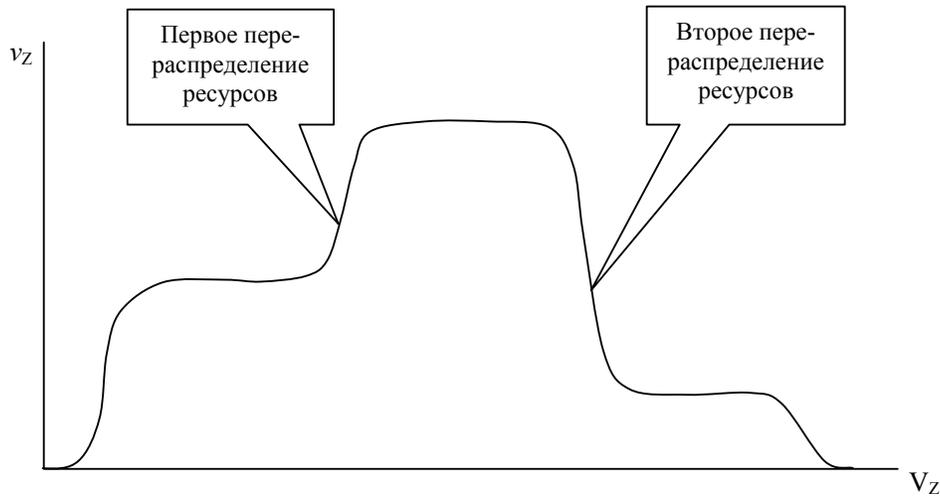


Рис. 4. Пример многошаговой стратегии перераспределения ресурсов

Выводы

В статье разработана модель динамики системы управления производственными процессами разработки программного обеспечения. Данная система учитывает неопределенность ряда параметров исполнителей и работ, вмешательство руководителя и перераспределение ресурсов между заданиями, а также рискованные события, которые могут наступить в ходе выполнения работ, и позволяет повысить точность и достоверность прогнозирования сроков выполнения работ.

СПИСОК ЛИТЕРАТУРЫ

1. ISO – Management standards. – Режим доступа : http://www.iso.org/iso/iso_catalogue/management_standards.htm.
2. Бушуев С. Д. Управление проектами. Основы профессиональных знаний и система оценки компетенции проектных менеджеров (National Competence Baseline, NCB UA Version 3.0) / С. Д. Бушуев, Н. С. Бушуева. – К. : ІРІДІ-УМ, 2006. – 208 с.
3. Типовые нормы времени на программирование задач для ЭВМ. – М. : НИИ труда, 1980. – 28 с.
4. Спайдер Проджект: Управление Проектами|Project Management|консалтинг|обучение|Spider Project – 2008. – Режим доступа : <http://www.spiderproject.ru/>.
5. Project Home Page – Microsoft Office Online – 2008. – Режим доступа : <http://office.microsoft.com/en->

us/project/FX100487771033.aspx.

6. Yousefli A. A. New Heuristic Model for Fully Fuzzy Project Scheduling / A. Yousefli, M. Ghazanfari, K. Shahanaghi, M. A. Heydari. – Режим доступа : <http://www.worldacademicunion.com/journal/jus/jusVol02No1paper07.pdf>.

7. Орловский С. А. Проблемы принятия решений при нечеткой исходной информации / С. А. Орловский. – М. : Радио и связь, 1981. – 286 с.

8. Раскин Л. Г. Нечеткая математика. Основы теории. Приложения / Л. Г. Раскин, О. В. Серая. – Х. : Парус, 2008. – 352 с.

9. Москвіна С. М. Аналіз алгоритмів розв'язання систем жорстких диференціальних рівнянь у пакетах моделювання систем автоматичного керування / С. М. Москвіна, Т. О. Голубева // Вісник Вінницького політехнічного інституту. – 2004. – № 2. – С. 21–25: іл. 5. – ISSN 1997–9266.

10. Москвіна С. М. Підхід до підвищення стійкості чисельних алгоритмів при моделюванні систем автоматичного керування / С. М. Москвіна, Т. О. Голубева // Вісник Вінницького політехнічного інституту. – 2003. – № 6. – С. 92–95: іл. 1. – ISSN 1997–9266.

11. Москвіна С. М. Про підвищення стійкості методів моделювання жорстких систем / С. М. Москвіна, О. М. Москвін, Т. О. Голубева // Вісник Вінницького політехнічного інституту. – 2007. – № 1 – С. 84 – 88. – ISSN 1997–9266.

Голубева Татяна Александровна – аспірантка кафедри комп'ютерних систем управління. gtat@bigmir.net.

Дубовой Володимир Михайлович – д. т. н., професор кафедри комп'ютерних систем управління. Вінницький національний технічний університет.