

УДК 004.89: 004.4: 004.421: 519.685: 681.5

**РЕШЕНИЕ ЗАДАЧИ РОБАСТНОЙ СТАБИЛИЗАЦИИ
НЕОПРЕДЕЛЕННОГО ОБЪЕКТА
В ДЕКЛАРАТИВНОЙ ПОСТАНОВКЕ****М.Ф. Степанов, А.М. Степанов, О.М. Степанова¹**Саратовский государственный технический университет имени Гагарина Ю.А.
Россия, 410054, г. Саратов, ул. Политехническая, 77

E-mail: mfstepanov@mail.ru, stepanovam@sstu.ru, stepanovaom@sstu.ru

Аннотация. В задачах синтеза систем автоматического управления (САУ) объектами с параметрической неопределенностью одной из проблем является необходимость учета возможных отклонений неопределенных параметров от их номинальных значений. Проблема выбора метода решения задачи синтеза закона управления осложняется не только их многообразием, но и недостаточной осведомленностью инженера-проектировщика в теории управления. Использование средств автоматического решения непроектно поставленных задач, т. е. без указания метода их решения, не применяется не только из-за необходимости создания и использования модели знаний проблемной области задач теории автоматического управления, но и из-за проблем планирования действий по их решению итерационными методами, т. к. проблема автоматического синтеза циклических программ не имеет универсального подхода к планированию действий для решения декларативно поставленных задач. Для конкретного класса задач может быть построена определенная структура плана действий по решению задач этого класса. Нарастающая потребность в построении новых САУ для новых объектов управления требует решения указанной проблемы. Одним из направлений построения систем управления объектами с неопределенными параметрами являются методы построения грубых (слабо чувствительных к изменениям параметров объекта) или, иначе, робастных систем управления. Методы синтеза робастного управления предусматривают итерационные процессы выполнения ряда подзадач. Статья посвящена исследованию путей решения декларативно поставленных задач робастного управления. Предложена схема представления циклических планов действий, реализующих итерационные методы синтеза законов управления. Рассмотрен пример решения декларативно поставленной задачи робастной стабилизации двухмассовой системы.



© Автор(ы), 2025

¹ Михаил Федорович Степанов, доктор технических наук, доцент, профессор кафедры системотехники и управления в технических системах.

Андрей Михайлович Степанов, кандидат технических наук, старший преподаватель кафедры системотехники и управления в технических системах.

Ольга Михайловна Степанова, аспирант кафедры системотехники и управления в технических системах.

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00488, <https://rscf.ru/project/24-21-00488/>

***Ключевые слова:** представление знаний, планирование действий, декларативно поставленные задачи, синтез итерационных планов решения задач, интеллектуальный решатель, планирующие искусственные нейронные сети.*

Введение

В системах автоматизации проектирования (САПР) все чаще возникает потребность в решении декларативно (непроцедурно, без указания метода решения) поставленных задач. Для области проектирования систем автоматического управления (САУ) характерно наличие неполноты, неопределенности сведений как о внешних возмущениях, так и об объекте управления. Методы теории автоматического управления (ТАУ) проектирования систем автоматического управления в условиях неопределенности являются итерационными, реализующими последовательное приближение к искомому решению [1–3].

В задачах синтеза систем управления объектами с параметрической неопределенностью одной из проблем является необходимость учета возможных отклонений неопределенных параметров от их номинальных значений. Отличительной особенностью неопределенных динамических систем с неопределенными параметрами является стационарность их неизвестных параметров. Таким образом, параметры системы являются неизвестными, однако постоянными, т. е. не изменяются во времени. Однако не существует общего, единого подхода (метода) решения задач управления в условиях неопределенности. Проблема выбора метода решения задачи синтеза закона управления осложняется не только их многообразием, но и недостаточной осведомленностью инженера-проектировщика в бурно развивающейся теории автоматического управления. Использование средств автоматического решения непроцедурно поставленных задач, т. е. без указания метода их решения, не допускается практически ни в одной из современных САПР САУ. Это обусловлено необходимостью создания модели знаний проблемной области задач ТАУ, на которой ставятся и решаются задачи. Также это инициируется принципиальными проблемами планирования действий по их решению. Важнейшей из них является то, что большинство методов ТАУ синтеза и анализа САУ являются итерационными. К сожалению, проблема автоматического синтеза циклических программ не имеет массового (общего) решения в связи с бесконечным разнообразием возможных алгоритмических схем реализации итерационного процесса, реализуемого порождаемой программой [4]. Таким образом, универсального подхода к планированию действий для решения декларативно поставленных задач не существует [5–9]. Для определенного достаточно узкого класса задач может быть определена некая обобщенная структура плана действий по решению задач этого конкретного класса, например с использованием крупноблочного, структурного, концептуального [5–9] подходов. В частности, возможно построение циклических планов решения некоторого класса задач итерационными методами, допускающими представление тела цикла в виде монолитного блока [4]. Нарастающая потребность в построении новых САУ для новых слабо изученных объектов управления в условиях ужесточения требований к затратам как времени, так и ресурсов на разработку и освоение производства требует решения указанной проблемы. Поэтому все более востребованными становятся методы решения задач синтеза и анализа САУ в условиях неопределенности, слабо чувствительных к изменениям параметров объекта, робастных законов стабилизации и управления. Настоящая статья посвящена изложению подхода к автоматизации решения задач синтеза и анализа САУ методами построения робастных законов стабилизации.

Методы синтеза робастного управления предусматривают итерационные процессы выполнения ряда подзадач, которые должны составить основной цикл порождаемой циклической программы. При этом необходим способ выделения среди действий, необходимых для решения исходной декларативно поставленной задачи, действий, выполняемых итерационно в цикле, в отличие от остальных действий, выполняемых лишь однократно. Известные методы представления знаний и планирования действий по решению задач [5–13], как прямой [10], так и обратный методы [12], позволяют выстраивать лишь линейные и древовидные (ветвящиеся) планы действий, включая их представление в виде треугольной таблицы [10]. С другой стороны, двухфазный подход к планированию действий [14] предусматривает разделение процесса поиска решения задачи на *первый* (предварительный, хотя и самый сложный и трудоемкий) этап отбора релевантных задаче действий, которые, скорее всего, и должны быть использованы в плане действий (программе) решения задачи, и *второй* (заключительный), на котором осуществляется уже собственно выполнение отобранных действий в порядке, определяемом наличием соответствующих исходных данных для них, завершая, таким образом, процесс построения плана действий решения задачи. При этом первый этап реализует обратный метод [12] поиска – от искомым результатов к необходимым исходным данным, а на втором этапе используется прямой метод поиска, определяющий готовые к выполнению операции (действия) системы автоматизации решения задач. На роль признака, отделяющего действия, выполняемые однократно, от действий, выполняемых многократно, наиболее подходящей кандидатурой является отношение, описывающее требования к искомому результату решения задачи. Поэтому действия модели знаний проблемной области, которые непосредственно и/или опосредованно влияют на значение такого отношения, можно было бы априорно в модели знаний пометить указанным отношением, например в требованиях к искомому результату действия. Это позволит действия построенного плана действий решения конкретной задачи разместить (локализовать) внутри тела основного цикла порождаемой программы (плана действий) решения поставленной задачи. Однако в разных задачах требования к искомому результату могут быть различными, что зависит от конкретной задачи и не позволяет осуществить такую предварительную разметку модели знаний. Таким образом, разметку действий на однажды выполняемые и циклически выполняемые придется делать непосредственно в процессе построения конкретного плана действий конкретной решаемой задачи. При этом простейший способ разметки заключается во включении всех действий построенного плана в состав циклически выполняемых действий («*повторное выполнение не ухудшает результат*»). Однако это приведет к необоснованным многократным затратам на выполнение действий, которые на пути достижения искомого результата достаточно выполнить только один раз. Если количество итераций окажется весьма большим, например сотни или даже тысячи, то эффективность такого «простого/легкого» способа организации циклических программ окажется недопустимо низкой. Как следствие, необходим подход к корректировке (разметке) модели знаний, включающий как статическую разметку аксиом (описаний действий), подходящую для всех задач, так и динамическую разметку действий для конкретной задачи, т. е. включенных в план решения задачи. Статическую разметку необходимо осуществлять на основе знаний об итерационных методах решения задач, включаемых в модель знаний. Указанная деятельность должна осуществляться непосредственно при пополнении модели знаний новыми знаниями проблемной области, осуществляемом учеными-

исследователями, выступающими в роли экспертов в проблемной области. Как следствие, процесс создания модели знаний проблемной области является также итерационным [4–17]. Кроме того, для создания алгоритмической структуры выполнения построенного плана действий конкретной задачи необходима разработка схемы представления синтезируемых программ, допускающей циклическое их выполнение (или их фрагментов) для реализации формализуемого для включения в модель знаний итерационного метода (например, синтеза законов робастной стабилизации объекта с неопределенными параметрами) проблемной области. Настоящая работа посвящена исследованию способов решения декларативно поставленных задач синтеза и анализа САУ на примере задач робастной стабилизации объекта с параметрической неопределенностью [1–3], предусматривающих итеративное выполнение плана действий до возможного достижения заданных требований к искомому результату решения задачи, т. е. порождение и выполнение циклических программ решения задач.

Модель и метод

Планирование действий по решению задач синтеза и анализа систем автоматического управления не может быть выполнено без использования знаний проблемной области [5–12]. В данном случае используется модель знаний [14–15], развиваемая в исследованиях авторов. Для изложения предлагаемого подхода приведем краткое описание ее структуры для введения обозначений используемых понятий [16–17]:

$$M_o = \langle D, \mathfrak{R}, O \rangle, \quad (1)$$

где $D = \{d_i \mid d_i = \langle \mathfrak{I}(d_i), \wp(d_i), \aleph(d_i), \Xi(d_i), \Psi(d_i) \rangle\}$ – множество формализованных обобщений (математических моделей) компонентов систем управления, называемых «предметами», обладающих: *свойствами* $\wp_j \in \wp(d_i) = \{\wp \mid \wp \in \{true \mid false\}\}$;

характеристиками $\chi_j \in \aleph(d_i) = \{\chi_{k_i} \mid \chi_{k_i} \in C^{n_{k_1}} \times C^{n_{k_2}} \times \dots \times C^{n_{k_l}}\}$, где C^n – n -мерное пространство комплексных чисел; *формами* математических моделей $\mu_j \in \mathfrak{I}(d_i) = \{\mu_1, \dots, \mu_{t_i}\}$, где

$$\mu_j = \langle n_j, m_j \rangle, \quad m_j = \{m_{jk} \mid m_{jk} \in C^{n_{jk_1}} \times C^{n_{jk_2}} \times \dots \times C^{n_{jk_l}}\},$$

$$n_j = \{n_{j1}, n_{j2}, \dots, n_{jl} \mid n_{jl} \in N\} -$$

множество коэффициентов компонентов (матриц m_{jk} соответствующих размерностей) математической модели μ_j ; классификационными *признаками*, представляющими собой выделенные свойства предметов $\xi_j \in \Xi(d_i) \subseteq \wp(d_i)$; *компонентами* (составными частями предметов), в качестве которых могут выступать другие предметы $d_{ij} \in \Psi(d_i) \subseteq D$;

$$\mathfrak{R} = \{r \mid r = \langle c, s, g \rangle: \wp \cup \aleph \cup \mathfrak{I} \cup \mathfrak{R} \cup O \rightarrow \{true \mid false\}\} -$$

множество *отношений* (предикатов) над предметами, их компонентами и атрибутами, действиями (операциями) над ними;

$$O = \{o \mid o = \langle c, s, g, q \rangle: \wp \cup \aleph \cup \mathfrak{I} \cup \mathfrak{R} \rightarrow \wp \cup \aleph \cup \mathfrak{I} \cup \mathfrak{R}\} -$$

множество *действий* (операций) над предметами и их атрибутами.

Действия $o_i = \langle c_i, s_i, g_i, r_i \rangle \in O$ и отношения $r_i = \langle c_i, s_i, g_i \rangle \in \mathfrak{R}$ характеризуются атрибутами: $c_i \in \wp \cup \mathfrak{R}$ – условия применимости; $s_i \in \wp \cup \aleph \cup \mathfrak{I}$ – исходные

данные; $g_i \in \wp \cup \aleph \cup \mathfrak{T} \cup \mathfrak{R}$ – результат выполнения действия (операции) ; $r_i \in \mathfrak{R}$ – требования к результату выполнения действия.

Задача на модели (1) представляется в виде $t_j = \langle s_j, g_j, r_j \rangle$ или в форме продукции $t_j : s_j \rightarrow g_j \leftarrow r_j$. План решения задачи t_j , порождаемый решающей подсистемой интеллектуального решателя задач, построенной на основе планирующих искусственных нейронных сетей (ПИНС) [14–17], представляется в виде подмножества

$$p(t_j) = \{o_{ji} \mid o_{ji} = \langle c_{ji}, s_{ji}, g_{ji}, r_{ji} \rangle, i = 1, \dots, n_j\}, \quad (2)$$

операций (аксиом) модели знаний, которые должны быть выполнены для получения искомого результата решения задачи, где $p(t_j)$ – план решения задачи t_j ; o_{ji} – i -я операция плана $p(t_j)$; n_j – количество операций, входящих в план $p(t_j)$ решения задачи t_j .

Подход к автоматизации решения непроцедурно поставленных задач управления в условиях неполноты, неопределенности сведений как о внешних возмущениях, так и об объекте управления, изложим на примере методов робастной стабилизации объектов управления с параметрической неопределенностью. Рассмотрим в качестве объектов управления системы вида

$$\dot{x} = \mathbf{A}x + \mathbf{B}u, x \in R^n, u \in R^m, \quad (3)$$

где элементы матриц $\mathbf{A}$ и $\mathbf{B}$ зависят неизвестным образом от неопределенных параметров

$$q \in Q \subseteq R^l. \quad (4)$$

Тогда вместо объекта с неопределенными параметрами (3) следует [1–3] представлять семейство динамических систем

$$\dot{x} = \mathbf{A}(q)x + \mathbf{B}(q)u, x \in R^n, u \in R^m, \quad (5)$$

каждая из которых представляет собой возможную реализацию системы (3), определенную конкретной комбинацией возможных значений неопределенных параметров (4).

Итерационные методы решения задач включают действия (операции), приводящие к «улучшению» значения критерия достижения цели задачи. Таким критерием служит заданная цель управления. Например, в задаче робастной стабилизации объекта управления (3) с неопределенными параметрами (4) целью является обеспечение устойчивости замкнутой регулятором системы управления [1–3] конкретным реализовавшимся экземпляром объекта управления из семейства (5). В соответствии с прямым методом Ляпунова система (3) является устойчивой, если для нее можно построить функцию Ляпунова

$$V(x) = x^T S x, S \succ 0, \quad (6)$$

удовлетворяющую

$$\mathbf{A}^T S + S \mathbf{A} \prec 0, \quad (7)$$

на движениях системы (3). Для обеспечения степени устойчивости матрицы $\mathbf{A}$ не меньше, чем заданное значение σ , вместо (7) следует использовать

$$\mathbf{A}^T S + S \mathbf{A} + 2\sigma S \prec 0. \quad (8)$$

Если для объекта (3) с неопределенностью (4) не удастся построить функцию (6), то необходимо решить задачу робастной квадратичной стабилизации [1–3]: для семейства динамических систем

$$\dot{x} = A(q)x + Bu, x \in R^n, u \in R^m \quad (9)$$

построить общий регулятор вида

$$u = Kx \quad (10)$$

такой, чтобы у замкнутых систем

$$\dot{x} = A_c(q)x, A_c(q) = A(q) + BK, \forall q \in Q \subseteq R^l \quad (11)$$

была бы общая функция Ляпунова (6).

Алгоритм решения задачи робастной стабилизации объекта с параметрической неопределенностью

В соответствии с [1–3] для решения задачи робастной стабилизации объекта с параметрической неопределенностью используется алгоритм, предусматривающий итерационное выполнение для всех систем семейства (5) следующих действий:

1) выбрать из семейства (5) i -ю систему, порожденную параметрами $q_i \in Q$;

2) решить матричное неравенство для заданного значения σ желаемой степени устойчивости замкнутой системы [3]

$$A_i S + S A_i^T + B K S + S K^T B^T \prec -2\sigma S \quad (12)$$

или матричное неравенство, введя $Y = KS$:

$$A_i S + S A_i^T + Y S + S Y^T \prec -2\sigma S; \quad (13)$$

3) если решение (12) или (13) не существует, то уменьшить значение σ , и если $\sigma \geq \sigma_0 > 0$, то идти к п. 1, где σ_0 – минимально допустимое значение степени устойчивости замкнутой системы, иначе идти к п. 7;

4) построить регулятор

$$K = YS^{-1}; \quad (14)$$

5) проверить устойчивость замкнутых построенным регулятором (14) систем управления для каждой из систем (5):

$$\dot{x} = A_c(q)x, A_c(q) = A_i + BK, \forall q_i \in Q \subseteq R^l. \quad (15)$$

Если хотя бы одна замкнутая система (15) с регулятором (14) является *неустойчивой*, то увеличить величину σ желаемой степени устойчивости и идти к п. 1, начиная решение задачи построения робастного регулятора (10) для систем семейства (6) сначала;

6) если остались системы семейства (5), для которых не решалась задача синтеза стабилизирующего регулятора, то идти к п. 1; иначе *успешно* завершить работу алгоритма с указанием в качестве результата решения поставленной задачи последний построенный робастный стабилизирующий регулятор (14) для объекта (3) с неопределенностью (4), который обеспечивает единую функцию Ляпунова (11);

7) завершить работу алгоритма *аварийно* в связи с невозможностью построения стабилизирующего регулятора при заданном уровне неопределенности (4) объекта управления.

Для обеспечения возможности автоматического построения планов решения декларативно поставленных задач робастной стабилизации необходимо осуществить корректировку модели знаний проблемной области теории автоматического управления [14, 15].

Подход к коррективке модели знаний проблемной области ТАУ

В статье в развитие крупноблочного подхода, структурного синтеза, концептуального программирования [5–7] предлагается подход к коррективке модели знаний проблемной области ТАУ, базирующийся на следующих допущениях:

– построение предопределенной циклической схемы, реализующей предложенный обобщенный итерационный алгоритм решения задачи;

– введение (дополнение) в структуру предложений (описаний выполняемых операций) генерируемого решателем плана действий (программы) описания дополнительных требований к искомым результатам решения поставленной задачи, представляемых в виде отношений или свойств соответствующего предмета модели знаний (1); значение указанного отношения или свойства должно вычисляться на каждой итерации выполнения плана действий (программы) решения задачи и будет использовано при анализе условия завершения итерационного процесса решения задачи.

Например, в качестве такого отношения (свойства) могут быть приняты:

– отношение r_j^k «запас устойчивости замкнутой системы управления не меньше заданного»:

$$r_j^k = \langle \chi_n, \chi_m \rangle \in R, \chi_n \in \mathbb{N}, \chi_m \in \mathbb{N}, \quad (16)$$

где k – номер итерации; χ_n – n -я характеристика предмета d_i , например σ_0 ; χ_m – m -я характеристика предмета d_j , например $\sigma = \max_i (\operatorname{Re} \lambda_i^k)$, где λ_i^k – собственные значения матрицы замкнутой системы (15), тогда $r_j^k = \langle \geq, \chi_n, \chi_m \rangle$, что означает $\sigma \geq \sigma_0$;

– свойство ρ_{ij}^k «замкнутая система устойчива»:

$$\rho_{ij}^k \in \wp(d_j^k) = \{\rho \mid \rho \in \{true \mid false\}\}. \quad (17)$$

В целом для обеспечения возможности автоматического построения интеллектуальным решателем задач [14] циклически выполняемого плана действий декларативно поставленной задачи синтеза робастного квадратичного стабилизирующего регулятора объектом (3) с неопределенностью (4) предлагается:

1) ввести в структуру описания постановки задачи дополнительные компоненты атрибута «Требования к результату»;

2) ввести в структуру предложений порождаемого плана действий решения поставленной задачи дополнительные компоненты атрибута «Требования к результату», элементами которого как логического выражения могут быть отношения вида (16) и/или свойства вида (17) предметов модели знаний;

3) ввести в модель знаний (1) отношение (свойство), описывающее условия завершения (продолжения) циклического выполнения операций плана действий;

4) ввести в атрибут «Требования к результату» предложений плана решения задачи отношение (свойство), определяющее условия прекращения циклического выполнения плана действий;

5) разработать схему описания плана решения задачи итерационным методом с контролем выполнения требований к искомому результату задачи;

6) дополнить исполнительную подсистему решателя задач алгоритмической структурой реализации циклического выполнения планов решения задач с заданными требованиями к искомому результату.

Реализация предлагаемого подхода осуществляется в рамках проекта модернизации системы автоматического решения задач ИНСТРУМЕНТ-3м-И [18]. Система ИНСТРУМЕНТ-3м-И предназначена для решения декларативно поставленных задач проблемной области автоматического управления на моделях знаний вида (1). Среди пользователей системы ИНСТРУМЕНТ-3м-И выделяются категории пользователей с принципиально различающимися целями деятельности, а следовательно, и ролями: а) ученые-исследователи, осуществляющие построение, корректировку и исследование методов решения задач проблемной области; б) инженеры, занимающиеся решением задач проектирования законов управления конкретных систем автоматического управления с использованием известных методов.

В связи с этим в системе ИНСТРУМЕНТ-3м-И реализованы два основных вида пользовательских интерфейсов: «Среда исследователя» (рис. 1) и «Среда инженера» (рис. 2).

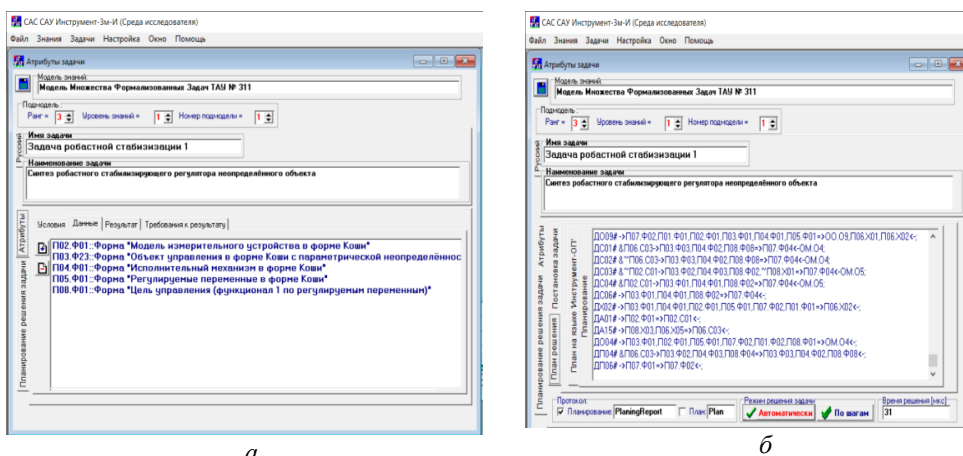


Рис. 1. Примеры представления экранных форм «Среды исследователя»: а – постановка тестовой задачи исследования построенной модели знаний; б – протокол тестирования корректности планирования действий решения задачи робастной стабилизации

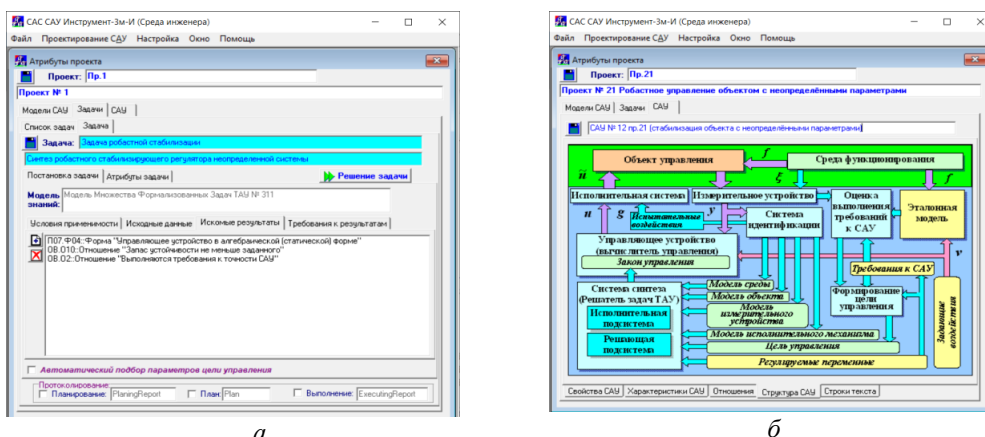


Рис. 2. Примеры представления экранных форм «Среды инженера»: а – пример постановки задачи синтеза робастного стабилизирующего регулятора неопределенной системы; б – пример формирования структуры системы управления

В поле «Планирование» представлен фрагмент списка аксиом модели знаний, описывающих действия, включенные в план решения задачи.

С учетом предложенного подхода к корректировке модели знаний проблемной области обобщенный алгоритм выполнения плана $p(t_j)$ решения задачи t_j итерационным методом можно описать следующим образом:

$$\neg Ex(p(t_j)) \rightarrow Go(p(t_j)) \doteq \{ \neg r_{ij}^k \rightarrow \{ \forall i (\neg Ex(o_{ji}) \rightarrow (Calc(r_{ij}^k) \rightarrow (Calc(c(o_{ji})) \rightarrow (\exists s(o_{ji}) \rightarrow ((Go(o_{ji}) \Rightarrow g_{ji}), (Set(Ex(o_{ji}), 1)) \} \forall i (Set(Ex(o_{ji}), 0))) \} \} \} \quad (18)$$

где $\doteq$ – «по определению»; $\neg$ – символ отрицания; $Ex(x)$ – предикат «объект x выполнен», в качестве x может выступать план действий $p(t_j)$ или отдельная операция $o_{ji}(\cdot)$; $Ex(x)$ принимает значение **1** (истина), если аргумент x выполнен, или **0** (ложь) в противном случае; $Set(x, v)$ – задание (присваивание) значения v для объекта x ; $Go(x)$ – выполнение объекта x (плана $p(t_j)$ или операции $o_{ji}(\cdot)$); $Calc(x)$ – вычисление значения x , т. е. выполнение операции вычисления x .

Формула (18) представляет схему функционирования исполнительской подсистемы решателя задач, осуществляющей итерационное выполнение плана действий $p(t_j)$ решения поставленной задачи t_j итерационным методом.

Для упрощения будущей реализации схема (18) может быть также представлена, например, в виде записи программы по А.А. Ляпунову [19] в форме схемы программы

$$p(t_j) := \downarrow^0 A_1 \uparrow^4 \downarrow^1 A_2 A_3 \uparrow^2 A_4 A_5 \uparrow^3 A_6 \uparrow^0 \downarrow^3 A_7 \uparrow^1 \downarrow^2 A_8 \uparrow^0 \downarrow^4 \quad (19)$$

и спецификации: A_1 – условие r_j завершения выполнения итеративного метода решения задачи выполнено (цель достигнута: регулятор построен, функция Ляпунова найдена); A_2 – выбор i -й динамической системы семейства (5); A_3 – матричное неравенство (13) не решено; A_4 – построить регулятор (14); A_5 – замкнутая система (15) устойчива; A_6 – увеличить желаемую степень устойчивости σ ; A_7 – остались неисследованные системы семейства (5); A_8 – уменьшить желаемую степень устойчивости σ , пока еще выполнено условие $\sigma \geq \sigma_0$.

$\uparrow^i$ – «передающая стрелка i » – выполнение передачи управления (переход) к оператору, отмеченному меткой «приемная стрелка i »;

$\downarrow^i$ – «приемная стрелка i » – метка оператора программы, начиная с которого продолжается ее выполнение после перехода по метке « i ».

Иллюстративный пример

Рассмотрим в качестве объекта управления двухмассовую систему [20] из двух твердых тел с массами m_1 и m_2 , соединенных пружиной с коэффициентом упругости k , скользящих без трения вдоль неподвижного горизонтального стержня (рис. 3). К левому телу приложено управление, призванное стабилизировать колебательную систему.

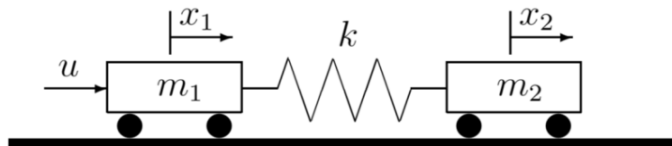


Рис. 3. Двухмассовая система [20]

Введем обозначения: x_1 и v_1 – координата и скорость левого тела; x_2 и v_2 – координата и скорость правого тела; $x = [x_1, x_2, v_1, v_2]^T$ – вектор фазового состояния динамической системы. Непрерывная модель колебаний двухмассовой системы [20] из двух твердых тел (см. рис. 3) описывается уравнениями [20]

$$\dot{x}_1 = v_1, \quad \dot{x}_2 = v_2, \quad \dot{v}_1 = (-k/m_1)x_1 + (k/m_1)x_2 + (1/m_1)u \quad (20)$$

или в матричном виде

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & k/m_1 & 0 & 0 \\ k/m_2 & -k/m_2 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{pmatrix} u, \quad (21)$$

где пара матриц $\mathbf{A}, \mathbf{B}$ полностью управляема, номинальные значения параметров модели (20) $k=1, m_1=1, m_2=1$, начальные условия

$$x(0) = [-1, 1, 1, -1]^T, \quad (22)$$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & k/m_1 & 0 & 0 \\ k/m_2 & -k/m_2 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{pmatrix}. \quad (23)$$

В системе (21) – (24) неопределенными приняты величины k, m_1, m_2 . Их неопределенные значения q_1, q_2, q_3 представляют интервальные числа с номинальными значениями k, m_1, m_2 . Предельное отклонение неопределенных величин от их номинальных значений в рамках решаемой иллюстративной задачи составляет $\pm 40\%$:

$$q_1 = [\underline{k}, \bar{k}], \quad q_2 = [\underline{m_1}, \bar{m_1}], \quad q_3 = [\underline{m_2}, \bar{m_2}], \quad (24)$$

где $\underline{z}$ и $\bar{z}$ – обозначают нижнюю и верхнюю границы значений z соответственно, тогда $\underline{k} = 0,4 \times k = 0,4$, $\bar{k} = 1,4 \times k = 1,4$, $\underline{m_1} = 0,4 \times m_1 = 0,4$, $\bar{m_1} = 1,4 \times m_1 = 1,4$, $\underline{m_2} = 0,4 \times m_2 = 0,4$, $\bar{m_2} = 1,4 \times m_2 = 1,4$.

Комбинации значений неопределенных параметров, определяющих множество динамических систем в семействе (всего 8):

$$\begin{aligned} &1) \{\underline{q_1}, \underline{q_2}, \underline{q_3}\}, 2) \{\underline{q_1}, \underline{q_2}, \bar{q_3}\}, 3) \{\underline{q_1}, \bar{q_2}, \bar{q_3}\}, 4) \{\bar{q_1}, \bar{q_2}, \bar{q_3}\}, \\ &5) \{\bar{q_1}, \underline{q_2}, \underline{q_3}\}, 6) \{\bar{q_1}, \underline{q_2}, \bar{q_3}\}, 7) \{\bar{q_1}, \bar{q_2}, \underline{q_3}\}, 8) \{\bar{q_1}, \bar{q_2}, \bar{q_3}\}. \end{aligned} \quad (25)$$

С учетом введенного обозначения (23) примет вид

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -q_1/q_2 & q_1/q_2 & 0 & 0 \\ q_1/q_3 & -q_1/q_3 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 1/q_2 \\ 0 \end{pmatrix}. \quad (26)$$

Требования к синтезируемому стабилизирующему регулятору: единый регулятор вида (10) для систем семейства (20)–(26); допустимые ограничения на значения переменных динамической системы и допустимое ограничение на величину стабилизирующего управляющего воздействия; максимальное количество итераций изменения критерия оптимальности принято равным 5; итерации останавливаются по достижении нарушения ограничения на величину управляющего воздействия:

$$|x_i| < x_i^* = 0, 1, i = 1, \dots, n, t > 10, u = |Kx| \leq u^* = 40. \quad (27)$$

Решение матричных неравенств (7), (8), (12), (13) полуопределенного программирования (SDP-задачи) [3, 20] требует применения итерационных методов. Вместо SDP-задачи может также итерационно решаться задача линейной квадратичной оптимизации, например с функционалом:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt, \quad Q = \text{diag}(q_{ii}),$$

$$q_{ii} = (1/(x_i^*)^2)^k, \quad i = 1, \dots, n, \quad R = [1],$$

где k – номер итерации выполнения цикла.

На рис. 4 представлены графики переходных процессов семейства динамических систем (20)–(25), замкнутых робастным стабилизирующим регулятором (10).

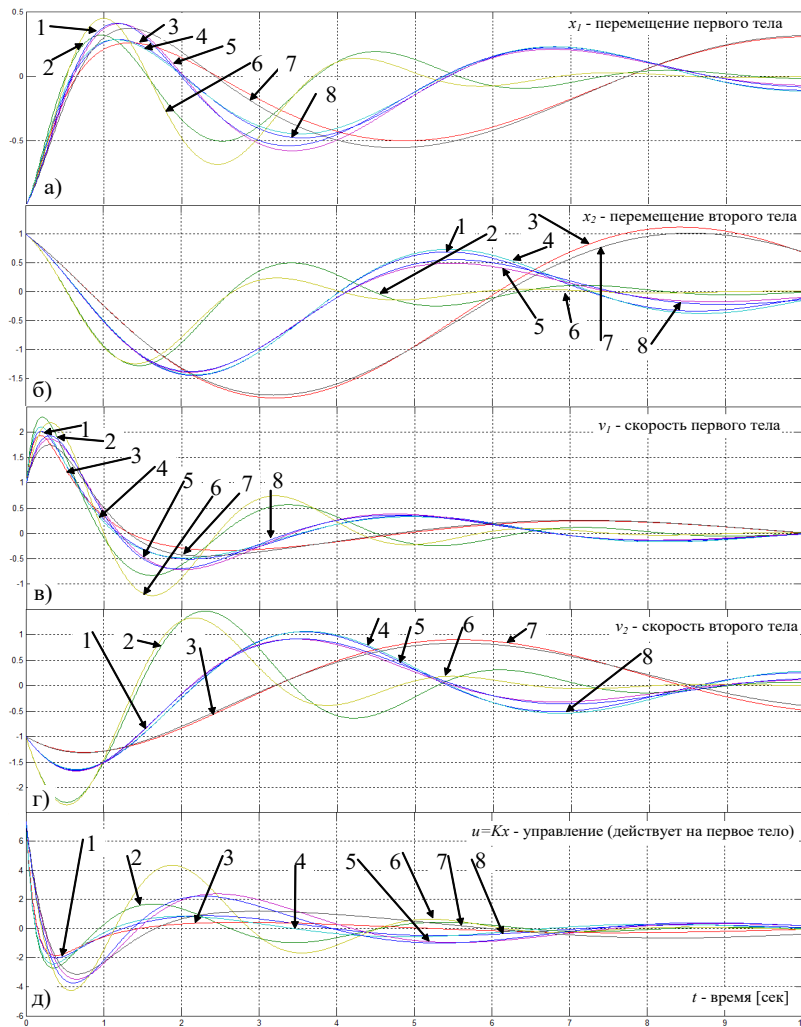


Рис. 4. Графики переходных процессов по переменным состояния а–г и управлению д в системе (21), (24–26), возбужденных начальными условиями (22) по результатам первой итерации цикла итеративного процесса решения задачи

Номера графиков соответствуют комбинациям значений неопределенных параметров q_1, q_2, q_3 (25). При последующих итерациях цикла группы графиков (1, 4, 5, 8), (2, 6), (3, 7) практически сливались, становясь почти неразличимыми. Различия между группами графиков сохранились.

В результате решения задачи синтеза стабилизирующего регулятора получены значения коэффициентов закона управления (матрица $\mathbf{K}$) (10) и значения параметров (матрица $\mathbf{S}$) функции Ляпунова (6):

$$\mathbf{K} = \begin{bmatrix} -25,280 \\ 9,149 \\ -13,068 \\ -9,020 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 319,207 & -132,788 & 35,392 & 130,69 \\ -132,788 & 261,461 & -12,808 & -9,316 \\ 35,392 & -12,808 & 18,295 & 12,628 \\ 130,69 & -9,316 & 12,628 & 160,052 \end{bmatrix}.$$

При этом для замкнутой построенным регулятором системы управления неопределенным объектом достигнута степень устойчивости $\sigma = 0,11975$ семейства динамических систем неопределенной системы (21), (24)–(26). Также удовлетворены требования (27) к динамическим характеристикам замкнутой системы управления регулятором по результатам третьей итерации цикла.

Выводы

Предложен подход к расширению возможностей интеллектуального решателя задач способностью построения (синтеза) циклических планов решения задач управления объектами с неопределенностью итерационными методами теории автоматического управления. Построена схема циклического выполнения плана действий, формируемого интеллектуальным решателем для решения декларативно поставленной задачи робастного управления. Построенная схема представлена в виде записи программы по А.А. Ляпунову, которая упрощает последующую реализацию схемы в исполнительной подсистеме решателя задач. Предложенная схема (19) применена при решении итерационным методом иллюстративной задачи построения робастного квадратичного стабилизирующего регулятора для системы (21), (24), (25) с неопределенными параметрами. Дальнейшие исследования направлены на изучение возможностей автоматизации построения планов решения декларативно поставленных задач управления объектами с другими схемами алгоритмических структур циклически выполняемых планов решения задач с различными видами неопределенности, продолжая развитие направления построения систем автоматического решения задач в декларативной постановке с использованием в качестве решающей (планирующей) подсистемы интеллектуального решателя задач планирующих искусственных нейронных сетей, важнейшим достоинством которых является способность одновременно (параллельно) решать как основную задачу, так и все ее подзадачи [14–17].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Красовский А.А. Справочник по теории автоматического управления. М.: Наука, 1987. 712 с.
2. Пупков К.А., Егунов Н.Д. Методы классической и современной теории автоматического управления: учебник: в 5 т. 2-е изд., перераб. и доп. М.: МГТУ им. Н.Э. Баумана, 2004. Т. 1. 656 с.
3. Поляк Б.Т., Щербаков П.С. Робастная устойчивость и управление. М.: Наука, 2002. 303 с.
4. Новиков Ф.А. Системы представления знаний. СПб: ИТМО, 2007. 119 с.

5. *Ненейвода Н.Н.* Соотношение между правилами естественного вывода и операторами алгоритмических языков высокого уровня // Доклады Академии наук СССР. 1978. Т. 239, № 3. С. 526–529.
6. *Лавров С.С., Залогова Л.А., Петрушина Т.И.* Принципы планирования решения задач в системе автоматического синтеза программ // Программирование. 1982. № 3. С. 35–43.
7. *Тыгуз Э.Х.* Концептуальное программирование. М.: Наука, 1984. 256 с.
8. *Ramirez-Rueda R., Benitez-Guerrero E., Mezura-Godoy C., Barcenas E.A.* A decade of advances in program synthesis from natural language: a systematic literature review // Proceedings of the Institute for System Programming of the RAS. 2024. Vol. 36, No. 6. Pp. 59–82. DOI: 10.15514/ispras-2024-36(6)-4. EDN: RLOHJI.
9. *Марков А.А.* Невозможность некоторых алгорифмов в теории ассоциативных систем // Доклады Академии наук СССР. 1947. Т. 55, № 7. С. 587–590.
10. *Ефимов Е.И.* Решатели интеллектуальных задач. М.: Наука, 1982. 192 с.
11. *Генцен Г.* Исследования логических выводов // Математическая теория логического вывода. М.: Наука, 1967. С. 9–76.
12. *Маслов С.Ю.* Теория дедуктивных систем и ее применения. М.: Радио и связь, 1986. 136 с.
13. *Мелехин В.Б., Хачумов М.В.* Полипеременные условно-зависимые предикаты в модели представления знаний автономных мобильных интеллектуальных агентов // Морские интеллектуальные технологии. 2021. № 2–1 (52). С. 184–190. DOI: 10.37220/MIT.2021.52.2.026. EDN: HVEGEX
14. *Степанов М.Ф.* Автоматическое решение задач теории автоматического управления на основе планирующих искусственных нейронных сетей // Нейрокомпьютеры: разработка, применение. 2003. № 3–4. С. 27–44.
15. *Степанов М.Ф.* Нейронные сети для планирования решения задач теории автоматического управления // Проблемы управления. 2004. № 2. С. 66–71. EDN: HSQSSR.
16. *Stepanov M.F., Stepanov A.M.* Mathematical modelling of intellectual self-organizing automatic control system: action planning research // Procedia Engineering. 2017. Vol. 201. Pp. 617–622. DOI: 10.1016/j.proeng.2017.09.657. EDN: XNKUQG.
17. *Stepanov M.F., Stepanov A.M., Stepanova T.V., Stepanova O.M.* Features of knowledge representation of automatic control systems design methods // Studies in Systems, Decision and Control. 2023. Vol. 457. Pp. 231–243. DOI: 10.1007/978-3-031-22938-1_16.
18. *Степанов М.Ф.* Анализ и синтез систем автоматического управления в программной среде ИНСТРУМЕНТ-3м-И // Известия высших учебных заведений. Приборостроение. 2004. Т. 47, № 6. С. 27–30. EDN: PAQNFF.
19. *Ляпунов А.А.* О логических схемах программ // Проблемы кибернетики. 1958. Вып. 1. С. 46–74.
20. *Поляк Б.Т., Хлебников М.В., Рапопорт Л.Б.* Математическая теория автоматического управления: учеб. пособие. М.: Ленанд, 2019. 500 с.

Статья поступила в редакцию 21.07.2025

SOLVING THE DECLARATIVE SET PROBLEM OF ROBUST STABILIZATION OF AN UNCERTAIN PLANT

M.F. Stepanov, A.M. Stepanov, O.M. Stepanova¹

Yuri Gagarin State Technical University of Saratov
77, Politehnicheskaya St., Saratov, 410054, Russian Federation

E-mail: mfstepanov@mail.ru, stepanovam@sstu.ru, stepanovaom@sstu.ru

Abstract. *In the tasks of synthesizing automatic control systems (ACS) for objects with parametric uncertainty, one of the problems is the need to take into account possible deviations of the values of uncertain parameters from their nominal values. The problem of choosing a method for solving the problem of synthesizing the control law is complicated not only by their diversity, but also by the lack of awareness of the design engineer directly in the theory of control. The use of automatic solutions for non-procedural tasks, i.e. without specifying the method of their solution, it is not used not only because of the need to create and use a knowledge model of the problem domain of the theory of automatic control. But also because of the difficulties of planning actions to solve them using iterative methods. Because the problem of automatic synthesis of cyclic programs does not have a universal approach to planning actions to solve declarative set tasks. However, for a specific class of tasks, a certain generalized structure of an action plan for solving problems of this class can be defined. The growing need to build new self-propelled guns for new control facilities requires solving this problem. One of the directions of building control systems for objects with uncertain parameters is the methods of building robust control systems that involve iterative processes of performing a number of sub-tasks. This article is devoted to the study of ways to solve declarative set tasks of robust management. An approach to adjusting the knowledge model is proposed that allows the planning subsystem, based on planning artificial neural networks, to generate action plans for declarative set tasks of robust stabilization, allowing their cyclical execution until acceptable indicators of the requirements for the desired result of solving the problem are achieved. We were developing the scheme for the presentation of cyclic action plans that implement iterative methods for synthesizing control laws. An example of solving a declaratively set task of robust stabilization of a two-mass system is considered.*

Keywords: *knowledge representation, action planning, declaratively set tasks, synthesis of iterative problem solving plans, intelligent solver, planning artificial neural networks.*

REFERENCES

1. Krasovskiy A.A. Spravochnik po teorii avtomaticheskogo upravleniya [Handbook of the theory of automatic control]. M.: Nauka, 1987. 712 p. (In Russian)



© The Author(s), 2025

¹ Mikhail F. Stepanov (Dr. Sci. (Techn.)), Associate Professor, Dept. of System Engineering and Control in Technical Systems.

Andrei M. Stepanov (Ph.D. (Techn.)), Senior Lecturer, Dept. of System Engineering and Control in Technical Systems.

Olga M. Stepanova, Postgraduate Student, Dept. of System Engineering and Control in Technical Systems.

The study was supported by the Russian Science Foundation grant No. 24-21-00488, <https://rscf.ru/project/24-21-00488/>

2. *Pupkov K.A., Egupov N.D.* Metody klassicheskoy i sovremennoy teorii avtomaticheskogo upravleniya [Methods of classical and modern theory of automatic control]. Vol. 1. M.: MGTU im. N.EH. Bauman, 2004. 656 p. (In Russian)
3. *Polyak B.T., Shcherbakov P.S.* Robastnaya ustojchivost i upravlenie [Robust stability and control]. M.: Nauka, 2002. 303 p. (In Russian)
4. *Novikov F.A.* Sistemy predstavleniya znanii [Knowledge Representations System]. SpB: ITMO, 2007. 119 p. (In Russian)
5. *Nepejvoda N.N.* Sootnoshenie mezhdu pravilami estestvennogo vyvoda i operatorami algoritmicheskikh yazykov vysokogo urovnya [The relationship between the rules of natural inference and the operators of high-level algorithmic languages] // DAN SSSR. Vol. 239, 1978. No. 3. Pp. 526–529. (In Russian)
6. *Lavrov S.S., Zalogova L.A., Petrushina T.I.* Principy planirovaniya resheniya zadach v sisteme avtomaticheskogo sinteza programm [Principles of problem solving planning in the system of automatic program synthesis] // Programmirovaniye. 1982. No. 3. Pp. 35–43. (In Russian)
7. *Tyugu En.H.* Konceptualnoe programmirovaniye [Conceptual programming]. M.: Nauka, 1984. 256 p. (In Russian)
8. *Ramirez-Rueda R., Benitez-Guerrero E., Mezura-Godoy C., Barcenat E.A.* A decade of advancements in program synthesis from natural language: a systematic literature review // Proceedings of the Institute for System Programming of the RAS. 2024. Vol. 36. No. 6. Pp. 59–82. DOI: 10.15514/ispras-2024-36(6)-4. EDN: RLOHJI.
9. *Markov A.A.* Nevozmozhnost nekotorykh algoritmov v teorii asociativnykh sistem [The impossibility of certain algorithms in the theory of associative systems] // DAN. 1947. No. 55. Pp. 587–590. (In Russian)
10. *Efimov E.I.* Reshateli intellektualnykh zadach [Intelligent Problem Solvers]. M.: Nauka, 1982. 192 p. (In Russian)
11. *Gencen G.* Issledovaniya logicheskikh vyvodov [Studies of logical conclusions] // Matematicheskaya teoriya logicheskogo vyvoda. M.: Nauka, 1967. Pp. 9–76. (In Russian)
12. *Maslov S.Yu.* Teoriya deduktivnykh sistem i eyo primeneniya [Theory of deductive systems and its applications]. M.: Radio i svyaz, 1986. 136 p. (In Russian)
13. *Melekhin V.B., Khachumov M.V.* Poliperemennyye uslovno-zavisimyye predikaty v modeli predstavleniya znanij avtonomnykh mobil'nykh intellektual'nykh agentov [Semipermanent conditionally dependent predicates in the knowledge representation model of autonomous mobile intelligent agents] // Morskoye intellektual'noye tekhnologii. 2021. No. 2–1 (52). Pp. 184–190. DOI: 10.37220/MIT.2021.52.2.026. EDN: HVEGEX. (In Russian)
14. *Stepanov M.F.* Avtomaticheskoye reshenie zadach teorii avtomaticheskogo upravleniya na osnove planiruyushchikh iskusstvennykh neyronnykh setey [Automatic problem solving in the theory of automatic control based on planning artificial neural networks] // Nejrokompyutery: Razrabotka i primeneniye. 2003. No. 3, 4. Pp. 27–44. (In Russian)
15. *Stepanov M.F.* Neyronnyye seti dlya planirovaniya resheniya zadach teorii avtomaticheskogo upravleniya [Neural networks for planning solutions to problems in the theory of automatic control] // Problemy upravleniya. 2004. No. 2. Pp. 66–71. (In Russian)
16. *Stepanov M.F., Stepanov A.M.* Mathematical modelling of intellectual self-organizing automatic control system: action planning research // Procedia Engineering. 2017. Vol. 201. Pp. 617–622. DOI: 10.1016/j.proeng.2017.09.657. EDN: XNKUQG.
17. *Stepanov M.F., Stepanov A.M., Stepanova T.V., Stepanova O.M.* Features of knowledge representation of automatic control systems design methods // Studies in Systems, Decision and Control. 2023. Vol. 457. Pp. 231–243. DOI: 10.1007/978-3-031-22938-1_16.
18. *Stepanov M.F.* Analiz i sintez sistem avtomaticheskogo upravleniya v programmnoy srede INSTRUMENT-3m-I [Analysis and synthesis of automatic control systems in the INSTRUMENT-3m-I software environment] // Izvestiya vysshikh uchebnykh zavedeniy. Priborostroeniye. 2004. Vol. 47. No. 6. Pp. 27–30. (In Russian)
19. *Ljapunov A.A.* O logicheskikh shemakh programm [About logical circuits of programs] // Problemy kibernetiki. No. 1. M.: Fizmatgiz, 1958. Pp. 46–74. (In Russian)
20. *Polyak B.T., Hlebnikov M.V., Rapoport L.B.* Matematicheskaya teoriya avtomaticheskogo upravleniya [Mathematical theory of automatic control: a textbook]. M.: Lenand, 2019. 500 p. (In Russian)

Original article submitted 21.07.2025