УДК 004.8/519.7

**В.В. ХИЛЕНКО, Р. СТРЖЕЛЕЦКИ, И. КОТУЛЯК**

# РЕШЕНИЕ ПРОБЛЕМЫ ДИНАМИЧЕСКОЙ АДАПТИВНОСТИ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА, ОСУЩЕСТВЛЯЮЩИХ УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ ТЕХНИЧЕСКИМИ ОБЪЕКТАМИ

**Аннотация.** Исследованы вопросы повышения быстродействия и устойчивости систем искусственного интеллекта, управляющих динамическими техническими объектами. Рассмотрена проблема вычисления оптимального момента переключения системой искусственного интеллекта с одного класса программного обеспечения на другой по критерию степени жесткости модели объекта управления. Предложено решение данной проблемы для общего случая динамики объекта управления, когда степень жесткости его математической модели может существенно изменяться во время функционирования и необходимо динамическое определение момента перехода от стандартных методов численного интегрирования к специализированным численным методам, предназначенным для расчета жестких моделей.

**Ключевые слова:** математическое моделирование, управление техническими системами, искусственный интеллект.

В настоящее время существенно повышаются требования к качеству систем управления множественными сложными динамическими объектами (например, беспилотными летательными аппаратами, роботизированными наземными транспортными средствами и т.д.) и техническими (экономическими/финансовыми) системами, функционирование которых связано с управлением динамикой технологических или информационных процессов. Сверхширокий диапазон вариативности систем управления, поддержка их работоспособности при различных значениях экзогенных факторов и дополнение системы управления программно-алгоритмическим обеспечением для выбора модели поведения объекта управления (ОУ) в соответствии с изменениями внешних условий (а при необходимости и выполнения корректировки внутренних параметров ОУ) позволяют относить такие системы управления к классу систем с искусственным интеллектом (ИИ). Для большинства систем управления высокотехнологическими сервисами, а также динамическими техническими системами и устройствами понимаемый в указанном смысле ИИ системы управления и обеспечивающий выполнение приведенных требований является необходимым составляющим элементом [1–4]. Используемый далее термин «система ИИ» применяется относительно класса систем управления, способных решать упомянутые задачи.

Одним из основных требований к рассматриваемому в настоящей работе классу систем ИИ является их адаптивность (устойчивость) в режиме реального времени. В общем случае такие системы ИИ должны быть динамически адаптивными к следующим факторам возмущения:

— изменениям экзогенных условий функционирования ОУ;
— изменениям эндогенных условий функционирования ОУ;
— изменениям внутренних параметров ОУ;
— корректировкам решаемых ОУ задач с учетом перечисленных изменений;
— изменениям решаемых ОУ задач.

При наличии этих факторов проблема функционирования системы ИИ, обеспечивающей управление динамическим ОУ, связана с определением момента переключения с одной модели, используемой для расчета управляющих воздействий, на другую, адекватную изменившимся факторам (фактору) возмущения. Предлагаемая в настоящей работе методика обеспечивает решение проблемы выбора момента переключения по такому важному параметру модели, как степень ее жесткости. Зачастую именно степень жесткости является основным параметром, определяющим время моделирования и расчета управляющих воздействий. Целью предлагаемого далее решения является сокращение времени, необходимого для расчета значений управляющих воздействий.

Время вычисления управляющих воздействий, которые обеспечивают реализацию системой ИИ целевой функции ОУ как при наличии всех перечисленных факторов, так и в случае их различных комбинаций, является одним из основных критериев качества этой системы. При проектировании системы ИИ с заданными характеристиками данного критерия качества необходимо учитывать и ряд приведенных далее факторов (в частности, вид математической формализации модели, выбранные решения вопросов, связанных с библиотекой моделей и библиотекой используемого ПО, и т.д). Рассмотрим эти основные факторы подробнее.

Системы ИИ, применяемые для управления сложными техническими объектами (системами), в общем случае используют для описания динамики ОУ системы обыкновенных дифференциальных уравнений (ОДУ). Моделирование динамики ОУ предполагает наличие библиотеки математических моделей, соответствующих различным диапазонам значений и комбинаций факторов возмущения. Это необходимое условие «устойчивости» системы ИИ, позволяющее отработать возмущения в широком диапазоне вариаций параметров факторов возмущения, а также их различных комбинаций, возникающих в процессе динамики ОУ. Увеличение устойчивости системы ИИ связано с формированием широкого спектра математических моделей.

Соответствующие конкретным целям и условиям функционирования ОУ на исследуемом временном интервале модели зачастую относятся к различным классам, например по степени жесткости. В этом случе математические модели, как правило, требуют применения различных численных методов для исследования и расчета, что обуславливает возникновение проблемы динамического контроля степени жесткости используемой модели и выбора момента переключения на соответствующий софт. Формирование библиотеки ПО, содержащей различные программные комплексы, реализующие различные численные методы, является важным элементом при создании систем ИИ. Полнота библиотеки ПО определяется возможностью выполнения численных расчетов для нахождения управляющих воздействий, реализующих целевую функцию в течение заданного временного интервала. В этом случае процесс функционирования системы ИИ можно рассматривать как сопровождающийся необходимым динамическим процессом своевременного и правильного выбора численных методов (программного обеспечения), являющихся эффективными и оптимальными для конкретной модели, используемой в данный момент $t_i$.

Для ряда высокотехнологических промышленных ОУ неприемлем даже кратковременный сбой в работе системы управления вследствие неполучения в оперативном режиме управляющих воздействий. Во избежание подобных ситуаций система ИИ должна быть независимой от обращений к облачным хранилищам данных или системам распределенных вычислений, когда гарантированное время отклика (получения расчетных данных) напрямую связано с работоспособностью коммуникационных систем (зависит от трафика систем передачи данных). Для общего повышения требований к характеристикам систем ИИ, в том числе к сокращению времени реагирования и т.д., целесообразно хранение в памяти системы ИИ библиотеки ПО, которое можно оперативно использовать в процессе работы системы ИИ. В отличие от адаптивных систем, в которых система управления должна обеспечивать коррекцию динамики ОУ, т.е. отрабатывать возникшие возмущения, система ИИ помимо исправления нарушений при необходимости может измененять (корректировать) внутренние параметры ОУ, чтобы, насколько это возможно, оптимизировать достижение функции цели с учетом последующих ожидаемых возмущений или новых условий функционирования. Реализация данного требования предполагает наличие в управляющем блоке определенной базы данных (базы знаний), необходимой для принятия оптимальных решений. Для использования такой базы данных в динамическом режиме, как правило, нужно выполнять большой объем вычислений, связанных с расчетом моделей (в том числе для решения оптимизационных задач).

Актуальным прикладным примером промышленных систем, в которых управляющие системы ИИ должны комплексно реагировать на вариативность параметров математических моделей ОУ, является использование систем ИИ, обеспечивающих управление динамикой автомобилей и аэродинамических объектов. В первом случае для учета вариативности даже по одному фактору внешней обстановки (например, управление движением автомобиля в светлое и темное время суток, при изменении погоды, уровня осадков и т.д.) системе ИИ для принятия адекватных управляющих решений может понадобиться замена (а не только модификация) математической модели динамики ОУ. Во втором случае переход к иной модели ОУ может обуславливаться, например, изменением режима полета или физического состояния (аэродинамической модели) самого ОУ. При этом коррекция лишь параметров используемой модели, адекватной предыдущему режиму, неприемлема.

В общем случае наиболее высокий уровень адаптивности (устойчивости) системы ИИ предполагает следующие ситуации:

— система ИИ имеет библиотеку моделей, содержащую большой спектр моделей различных классов, адекватно описывающих динамику ОУ в различных ситуациях и при этом допускающих широкую вариативность своих параметров;

— система ИИ имеет библиотеку ПО, позволяющего оптимально проводить исследование и расчет используемых математических моделей в течение разрешенного (технологического) временного интервала;

— выполнение приведенных ранее условий обеспечивает системе ИИ в режиме реального времени (допустимого времени реагирования) индифферентность к внешним факторам, находящимся вне зоны влияния системы ИИ.

Актуальной задачей построения систем ИИ является выбор оптимального математического и программно-алгоритмического обеспечения, адекватного конкретной используемой модели для ее эффективного исследования и расчета, и следовательно, определение момента переключения с одного класса ПО на другой, более приемлемый для работы с моделью в данный момент времени.

Как уже отмечалось, в настоящей работе предлагается решение задачи выбора момента переключения с одного класса ПО на другой по одному из основных отличий в моделях, создающих проблему при выборе методов численного интегрирования, а именно по степени жесткости исследуемой математической модели. Вычислительному аспекту данной проблемы посвящено много публикаций и, как отмечено в некоторых из них [5], не существует универсальных численных методов одинаково эффективных для расчета математических моделей различных классов степени жесткости. Необоснованный (случайный) выбор любого метода численного интегрирования для расчета степени жесткости модели может привести к неправильным результатам. Однако использование большинства специализированных методов, ориентированных на расчет степени жесткости моделей, для расчета нежестких моделей существенно неоптимально и связано с большими затратами времени, не оправданными спецификой задачи. Определение момента переключения (т.е. момента замены применяемых вычислительных методов или используемой математической модели ОУ) включает вычисление собственных чисел матриц. Решение проблемы вычисления собственных чисел матриц, изложенное в [6], позволяет в отличие от известных методов избежать экспоненциального роста объема вычислений при увеличении размерности модели.

Пусть модель динамики ОУ представлена в виде

$$\frac{dx}{dt} = f(x, u, t), \tag{1}$$

$$f(x, u, t) \in \Gamma,\ \Gamma \subset I_t \times R^m,\ I_t = \{0 \le t < \infty\},$$

а система ИИ решает задачу Лагранжа для функционала $J(x)$:

$$J(x) = \int_0^T F(x, u)dt$$

с учетом условий $x(0) = X_0$, $u \in U$, где $U$ — некоторое множество. Полагаем, что функционал $J(x)$, определенный в $m$-мерном евклидовом пространстве $R^m$, достигает своего экстремального значения на некотором подмножестве $X^* \subset R^m$.

Для оценки функционирования системы ИИ рассмотрим общий случай:

— совокупные параметры процесса динамики ОУ и параметры самого ОУ могут существенно изменяться и модель ОУ может многократно (на порядки) изменять свою степень жесткости на интервале функционирования, т.е. на отдельных интервалах функционирования ОУ для модели (1) справедливы известные [7] соотношения:

$$\left| \frac{d^l x^{(k)}}{dt^l} \right|_{t \ge t_0 + \tau_{BL}} \le \left( \frac{L}{N} \right)^l_{t \in [t_0, t_0+T]} \max |x^{(k)}(t)|,\ l > 1,$$

где $0 < L \le \rho\left(\frac{\partial f}{\partial x}\right) \le \left\| \frac{\partial f}{\partial x} \right\|$; $(t, x) \in \Gamma$; $N >> 1$; $\rho\left(\frac{\partial f}{\partial x}\right)$ — максимальный модуль собственных чисел матрицы Якоби; $||\cdot||$ — принятая норма матрицы; $\tau_{BL}$ — величина интервала пограничного слоя, удовлетворяющая неравенству $\tau_{BL} << t_D$, здесь $t_D$ — величина интервала исследования динамики ОУ;

— система ИИ вынуждена заменять модель для обеспечения адекватности измененным факторам возмущения.

Численное моделирование динамики сложных технических объектов подтверждает справедливость замечания о том, что на большом временном интервале управления объектом возникновение жесткости в его математической модели больше правило, чем исключение. Следовательно, данное правило обуславливает требование наличия соответствующих специализированных методов в программно-алгоритмическом обеспечении системы ИИ как обязательное к исполнению.

При этом поставлена задача оптимизации функционала $J(x)$, для которого в некоторой точке $x_*$ выполняется соотношение

$$J(X^*) = \underset{x \in R^m}{\operatorname{extr}} J(x)\ \ \forall x_* \in X^*.$$

Далее будем полагать, что функционал $J(x)$ является дважды непрерывно дифференцируемым. Данную задачу оптимизации рассмотрим как задачу определения некоторой точки $\overline{x_*}$, находящейся в окрестности $\varepsilon$ точки $x_*$, удовлетворяющей одному из следующих условий:

$$|| p(\overline{x_*}) - \overline{x_*} || \le \varepsilon,$$

$$| J(X^*) - J(\overline{x_*}) | \le \varepsilon,$$

где $\varepsilon$ — заданная точность решения, а $p(x_*)$ — проекция $x_*$ на множество $X^*$.

Как известно, если функционал $J(x)$ овражный и размерность вектора $x$ не является малой величиной, то решение задачи оптимизации достаточно сложное и требует использования специальных вычислительных методов и соответственно специального ПО.

Решение задачи выбора ПО влияет как на оценки качества системы ИИ, так и на оценку функциональности (устойчивости) системы ИИ в целом. Программно-алгоритмическая реализация вычислительной процедуры, которая обеспечит минимально возможные затраты времени (вычислительного ресурса), — решение данной проблемы. Информацией, необходимой для принятия решения, в данном случае является множество значений скоростей протекания исследуемых процессов, т.е. производных переменных состояния математической модели.

Для решения рассматриваемой задачи необходимы численное определение степени жесткости модели и выбор значений ее параметров, при которых целесообразен переход от стандартных методов расчета к специализированным и наоборот.

Процедуру определения степени жесткости модели рассмотрим на примере системы (1), линеаризованной на некотором интервале

$$\dot{X} = Ax, \ x \in R^m, \ t \in [t_1, t_2], \tag{2}$$

как задачу вычисления собственных чисел $\lambda_i$ матрицы $A$. Полагаем, что $A = [a_{ij}]_1^n$ — матрица размера $n \times n$ с действительными элементами, полученная из функционала $J(x)$.

В настоящее время в известных авторам работах не приведен точный количественный критерий, когда математическую модель объекта или системы можно отнести к классу жестких. Как правило, в публикациях, где исследуются жесткие системы, ограничиваются качественным критерием, согласно которому система ОДУ относится к классу жестких. Один из количественных критериев, удобный в практическом использовании, — критерий оценки на основе спектрального числа обусловленности [7]

$$K(A) >> 1,$$

где

$$K(A) = \frac{\max\limits_j |\operatorname{Re} \lambda_j|}{\min\limits_j |\operatorname{Re} \lambda_j|} >> 1, \ \operatorname{Re} \lambda_j < 0, \ j = \overline{1, n},$$

а значения $\lambda_j$, являющиеся функциями времени, фиксированные для данного момента $t = t^*$.

Далее для определенности будем следовать достаточно условной классификации по количественному критерию:

— слабо жесткие модели (нулевой класс степени жесткости) при $K(A) \le 50$;

— жесткие модели (первый класс степени жесткости) при $50 < K(A) \le 1000$;

— сильно жесткие модели (второй класс степени жесткости) при $K(A) > 1000$;

— сверхжесткие модели (третий класс степени жесткости) при $K(A) \ge 10000$.

Выбор границ классов степени жесткости имеет эмпирический характер.

Решения проблемы выбора используемого ПО требует наличия в арсенале системы ИИ алгоритма (алгоритмов) быстрого определения, к какому классу степени жесткости относится математическая модель.

Собственные числа $\lambda_i$ матрицы $A$ можно найти с помощью известных численных алгоритмов [5]. Характерной особенностью большинства из них есть экспоненциальный рост объема вычислений при увеличении размера матрицы $A$. Альтернативой этим численным методам является реализация численного итерационного алгоритма, основанного на следующей теореме [6].

**Теорема 1** (Khilenko). Для жесткой системы линейных дифференциальных уравнений вида (1), матрица $A$ которой имеет действительные собственные числа $\lambda_i$ $(i=\overline{1,n})$ справедливы соотношения

$$|\lambda_l| = \left|\lim_{g\to\infty} \Delta p^{(g)}(A)\right|,$$

где $l=\text{const}$, $1\le l\le n$, $\Delta\rho^{(g)}(A)=\text{Sp}\,A-\text{Sp}\,A^{(g)}$, $A^{(g)}$ — матрица размера $(n-1)\times(n-1)$, полученная из матрицы $A$ при замене во 2-м, …, $n$-м уравнениях системы (2) переменной $x_1$ выражением, найденным из соотношения

$$\frac{d^g x_1}{dt^g} = f_{1g}(x_1,\ldots,x_n) = 0.$$

Приведенная теорема является основанием для построения алгоритмов определения собственных чисел матрицы $A$. Если собственные числа действительные, то для их определения можно использовать модифицированный алгоритм получения числа избыточных дифференциальных связей в исходной системе уравнений [6, 8].

1. Исключая переменную $x_1$, формируем новую матрицу $A(1)$ размера $(n-1)\times(n-1)$.

2. Определяем $\lambda_k(1)=\text{Sp}\,A-\text{Sp}\,A(1)$.

3. Полагая $x_1^{(2)}=0$, формируем аналогично п. 1 матрицу $A(2)$.

4. Вычисляем $\lambda_k(2)$ аналогично п. 2.

5. Определяем $\varepsilon(1)=\lambda_k(1)-\lambda_k(2)$.

6. Если $\varepsilon(1)\le\varepsilon^*$ (где $\varepsilon^*$ — некоторая заданная величина), то завершаем работу алгоритма, если $\varepsilon(1)>\varepsilon^*$, то выполняем следующий цикл вычислений.

Отметим, что данный алгоритм можно модифицировать и применять для матриц, собственные числа которых являются комплексно-сопряженными.

В общем случае, применяя к исходной системе уравнений метод понижения порядка [8] и устанавливая для переменной $x_1$ алгебраическую связь, можно записать $X_1=(X,a_{ij})$.

Повышая степень вырождаемой производной в соответствии с заданной точностью вычислений, определяем $X_1(S)=(X_k,a_{ij}(S))$, где $a_{ij}(S)$ — элементы матрицы $A(S)$.

Матрица $A(S)$ получается из исходной матрицы формированием алгебраической связи для $s$-й степени вырождаемости производной $X_1$ [8].

Используя приведенную теорему, последовательно определяем приблизительные ($s$-й степени приближения) [5, 7] значения собственных чисел $\lambda_i(S)$ $(i=\overline{1,n})$ матрицы $A$.

Полагая $\lambda_i=\lambda_i(S)$ $(i=\overline{1,n})$, можно вычислить $x_i(t)$ $(i=\overline{1,n})$.

Не выйдя за рамки решаемой задачи, отметим, что, поскольку $x_i=f(c_{ij},\lambda_j)$ $(i,j=\overline{1,n})$, $c_{ij}=\text{const}$, данные процессы вычислений можно выполнять параллельно на различных процессорах.

Таким образом, применение декомпозиционных методов позволяет эффективно использовать многопроцессорные вычислительные комплексы, обеспечивая процедуру распределенных вычислений для решения вычислительных задач, возникающих в процессе функционирования системы ИИ. Отметим, что проведенные численные эксперименты показали работоспособность алгоритмов, основанных на приведенной теореме 1, и для матриц произвольной степени жесткости.

Как известно, метод понижения порядка относится к классу декомпозиционных. Однако в целом данный подход к решению задачи определения момента переключения ПО в системах ИИ может оказаться полезным также и в случае, когда система ИИ использует методы, не относящиеся к классу декомпозиционных.

Задачу вычисления собственных чисел матрицы Якоби, соответствующей функционалу $J(x)$, в этом случае можно рассматривать как элемент решения оптимизационной задачи. В качестве примера приведем метод Гринштадта [7] для решения оптимизационной задачи (1) на основе анализа собственных чисел $\lambda_j(J'')$ и собственных векторов $u_j$, $j=1,\ldots,m$, матрицы Гессе $J''$, когда

$$x_{k+1} = x_k - h_k [\overline{J_k''}]^{-1} J_k',$$

где $\overline{J_k''} = \sum_{j=1}^{m} \bar{\lambda}_j u_j u_j^{\mathrm{T}}$, $\bar{\lambda}_j = \max\{\delta, |\lambda_j(J'')|\}$, $\delta = \mathrm{const} > 0$.

Нетрудно видеть, что для данного алгоритма необходимо вычисление всех собственных чисел матрицы $J(x)$.

Особо отметим случай, когда системе ИИ приходится выполнять в режиме реального времени снятие данных, связанных с функционированием ОУ и, используя их, корректировать значения коэффициентов модели динамики ОУ. При этом аналогичной, но дополнительной задачей, которую необходимо решить системе ИИ, является задача параметрической идентификации. Для систем ИИ, обеспечивающих управление динамикой технического ОУ (например, управление автоматизированным робототехническим комплексом или управление сетью передачи данных [9]), задача параметрической идентификации стандартная и должна решаться в режиме реального времени. Аналогично получение неправильного результата возможно в случае, если исходная математическая модель не была жесткой, но во время функционирования ОУ перешла в класс жестких (сильно жестких). Если задача параметрической идентификации решалась с помощью стандартных (не проблемно-ориентированных) методов без учета возможных обновлений класса степени жесткости модели, то вследствие ошибок при определении малых собственных чисел матриц, описывающих изучаемую систему, управляющие сигналы можно рассчитать неправильно. При этом не изменятся параметры системы или объекта, что необходимо для выполнения ОУ поставленной задачи.

Известным классическим примером является модель простой передаточной функции ОУ $K = \dfrac{b}{p+a}$, $a, b > 0$, где параметры $a$ и $b$ необходимо вычислять на основе данных амплитудно-частотной характеристики:

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $w_i$ | 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 | 2.0 |
| $S_i$ | 71 | 67 | 64 | 61 | 58 | 55 | 53 | 51 | 49 | 47 | 45 |

При решении проблемы параметрической идентификации для данной модели с учетом ее вариабельности легко понять суть рассматриваемой проблемы: подбор метода, оптимального по сложности, эффективности и временным затратам. Методы, которые не являются проблемно-ориентированными, например метод быстрого спуска, не обеспечивают нахождения экстремума с заданной точностью в широким диапазоне параметров модели [7]. В то же время неспециализированные методы экономичнее по затратам машинного времени и проще в реализации. Если в рассматриваемой модели значения параметров $a$ и $b$ близки, то применение специальных методов, ориентированных на работу с овражными функциями, необоснованно.

Таким образом, алгоритмы, используемые системой управления, должны обеспечивать адекватность модели применяемым методам расчета в каждый дискретный момент времени. Для жестких моделей такую адекватность можно обеспечить переходом к декомпозиционным методам и их использованием в последующих расчетах, пока модель является жесткой.

При комплектовании библиотеки специализированных (проблемно-ориентированных) методов системе ИИ следует учитывать преимущества использования декомпозиционных методов для расчета степени жесткости моделей. Необходимость включения их в библиотеку системы ИИ объясняется следующим. Практически всегда процессы приема и передачи информации в любом достаточно сложном кластере «система ИИ–динамический ОУ» связаны с необходимостью проверки ошибок измерений и исправления искажений, которые возникли при передаче сигналов. Например, в процессе передачи информации в сетях связи необходим учет помех, потерь и искажений данных в каналах передачи. Корректировки приводят к изменению элементов матриц соответствующих математических моделей. Можно записать следующее соотношение, когда некоторая матрица возмущений $G$ добавляется к исходной матрице [7]:

$$k_1 \Lambda_A \leq G \leq k_2 \Lambda_A , \tag{3}$$

где $k_1, k_2$ — некоторые коэффициенты, а $G$ — учитывает величины, на которые могут измениться любые собственные числа матрицы $A$.

В силу соотношения (3), если матрица $A$ плохо обусловлена и система уравнений (2) является жесткой, то даже довольно малые искажения элементов матрицы $A$ могут привести к тому, что большие по модулю собственные числа изменятся в процентном соотношении пропорционально внесенным изменениям, т.е. «практически не почувствуют» их. Тогда как изменения малых собственных чисел могут превысить все допустимые пределы, исказив их истинные значения на сотни процентов.

Если применяются методы декомпозиционного типа (например, метод понижения порядка [8]), то в вычислениях используется преобразованная математическая модель, состоящая из нежесткой подсистемы ОДУ и подсистемы алгебраических уравнений. В этом случае минимизируется опасность получения компонентов решения по «медленным» переменным с искажениями в сотни процентов. Если исследователь сталкивается с явлением жесткости в используемой математической модели и переходит к преобразованной (декомпозиционной) модели, где часть дифференциальных уравнений будет заменена алгебраическими соотношениями, то искажения элементов вследствие помех или ошибок в измерениях будут «симметричными» в процентном отношении как для больших, так и для малых по модулю собственных чисел матрицы $A$.

Использование методов декомпозиции для уменьшения объема вычислений и времени отклика системы ИИ требует отдельного рассмотрения и выходит за рамки данной статьи, однако отметим, что предложенное решение задачи динамического контроля степени жесткости модели полностью применимо и для субмоделей, когда они потенциально должны относиться к одному классу и интервал контрольных точек (точек бифуркации) для проверки субмодели по критерию жесткости значительно увеличивается.

СПИСОК ЛИТЕРАТУРЫ

1. Boden M.A. AI. Its nature and future. Qxford: Oxford University Press, 2016. 208 p.
2. Rajan K., Safiotti A. Towards a science of integrated AI and Robotics. *Artificial Intelligence.* 2017. Vol. 247. P. 1–9.
3. Benysek G., Kazmierkowski M.P., Popczyk J., Strzelecki R. Power electronic systems as crucial part of a smart grid infrastructure — a survey. *Bulletin of the Polish Academy of Sciences* (*Technical Sciences*). 2011. Vol. 59, N 4. P. 455–473.
4. Khilenko V.V. Mathematical modeling of the effect of "splashing out" and optimization of management of banking and economic systems under globalization conditions. *Cybernetics and Systems Analysis*. 2018. Vol. 54, N 3. P. 376–384.
5. Computational mathematics: Theory, methods and applications. Chareton P.G. (Ed.). NY: Nova Science Pub. Inc, 2011. 443 p.

6. Khilenko V.V. Convergence of the order decrease method for solution of the rigid systems of linear differential equations. *Reports of the Academy of Sciences of the Ukrainian SSR*. 1987. N 8. P. 76–79.
7. Ракитский Ю.В., Устинов С.М., Черноруцкий И.Г. Численные методы решения жестких систем. Москва: Наука, 1979. 208 с.
8. Khilenko V.V. Methods of system analysis in solving problems of research of adaptive communication and control systems. Kiev: Interlink, 2003. 216 p.
9. Hájovskỳ V., Kotuliaková K., Kotuliak I. HARQ schemes for HSDPA — analysis and simulation. *Proc. Elmar-2008: 50th Int. Symp. ELMAR-2008* (Zadar, Croatia, 10–12 Sept. 2008). Zadar: Croatian Society Electronics in Marine, 2008. Vol. 2. P. 557–560.

*Надійшла до редакції 26.03.2018*

**В.В. Хиленко, Р. Стржелецкі, І. Котуляк**

**РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ДИНАМІЧНОЇ ВАРІАТИВНОСТІ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ, ЩО ЗДІЙСНЮЮТЬ КЕРУВАННЯ ДИНАМІЧНИМИ ТЕХНІЧНИМИ ОБ'ЄКТАМИ**

**Анотація.** Вивчено питання підвищення швидкодії і стійкості систем штучного інтелекту, що керують динамічними технічними об'єктами. Розглянуто задачу обчислення оптимального моменту перемикання системою штучного інтелекту з одного класу програмного забезпечення на інший за критерієм ступеня жорсткості моделі об'єкта керування. Запропоновано розв'язання цієї проблеми для загального випадку динаміки об'єкта керування, коли ступінь жорсткості його математичної моделі може істотно змінюватися під час функціонування і потрібно динамічно визначати момент переходу від стандартних методів чисельного інтегрування до спеціалізованих чисельних методів, призначених для розрахунку жорстких моделей.

**Ключові слова:** математичне моделювання, керування технічними системами, штучний інтелект.

**V.V. Khilenko, R. Strzelecki, I. Kotuliak**

**SOLVING THE PROBLEMS OF DYNAMIC VARIABILITY OF ARTIFICIAL INTELLIGENCE SYSTEMS THAT CONTROL DYNAMIC TECHNICAL OBJECTS**

**Abstract.** The paper investigates increase in performance and stability of artificial intelligence systems that control dynamic technical objects. The problem of calculating the optimal switching time of the artificial intelligence system between the software classes by the criterion of the degree of rigidity of the model of the control object is considered. The solution of this problem is proposed for the general case of the control object dynamics when the rigidity of its mathematical model can significantly change during the operation and it is necessary to dynamically determine the transition time from standard methods of numerical integration to specialized numerical methods intended for calculating rigid models.

**Keywords:** mathematical modeling, control of engineering systems, artificial intelligence.

**Хиленко Владимир Васильевич,**
доктор техн. наук, профессор кафедры Национального университета биоресурсов и природопользования Украины МОН Украины, Киев, e-mail: vkhilenko@ukr.net.

**Стрежелецкий Ришард,**
доктор техн. наук, профессор кафедры Гданьского политехнического института (Республика Польша).

**Котуляк Иван,**
ассистент-профессор Словацкого национального технического университета, Братислава.