

«Информатика»

Научная статья

УДК 004.94

DOI: 10.17072/1993-0550-2024-1-43-52

**Компьютерное моделирование робастных
и адаптивных алгоритмов в технологии
модельно-ориентированного проектирования****Александр Юрьевич Ощепков¹, Александр Николаевич Власов²**^{1,2} Пермский государственный национальный исследовательский университет, Пермь, Россия¹aos57@mail.ru;²anvlasov99@yandex.ru

Аннотация. Автоматические цифровые регуляторы, обеспечивающие работу автономных (бортовых или встраиваемых) устройств, должны использовать робастные или адаптивные алгоритмы. Для составления программы на языке С, реализующей современные эффективные алгоритмы управления, используется технология модельно-ориентированного проектирования (МОП-технология), суть которой заключается в том, что сначала составляется и отлаживается имитационная компьютерная модель системы автоматического управления на языке более высокого уровня, а затем генерируется код работы регулятора для микропроцессора. В статье приведен обзор робастных и адаптивных алгоритмов, которые могут быть применены в бортовых системах автоматического управления. Построены компьютерные модели систем управления с применением этих алгоритмов в пакетах MATLAB и SimInTech, как примеры начальных этапов технологии модельно-ориентированного проектирования.

Ключевые слова: модельно-ориентированное проектирование; системы автоматического управления; робастные алгоритмы; адаптивные алгоритмы

Для цитирования: Ощепков А.Ю., Власов А.Н. Компьютерное моделирование робастных и адаптивных алгоритмов в технологии модельно-ориентированного проектирования // Вестник Пермского университета. Математика. Механика. Информатика. 2024. Вып. 1(64). С. 43-52. DOI: 10.17072/1993-0550-2024-1-43-52.

Статья поступила в редакцию 24.01.2024; одобрена после рецензирования 02.02.2024; принята к публикации 18.03.2024.

«Computer Science»

Research article

**Computer Modeling of Robust and Adaptive Algorithms
in Model-based Design Technology****Alexander Yu. Oshchepkov¹, Alexander N. Vlasov²**^{1,2} Perm State University, Perm, Russia¹aos57@mail.ru;²anvlasov99@yandex.ru

Abstract. Automatic digital controllers that ensure the operation of autonomous (on-board or embedded) devices must use robust or adaptive algorithms. To compile a program in C language that implements modern effective control algorithms, model-based design technology (MOP technology) is used, the essence of which is that first a simulation computer model of the automatic control system is being compiled and



Данная работа © 2024 Ощепков А.Ю., Власов А.Н. распространяется под лицензией CC BY 4.0. Чтобы просмотреть копию этой лицензии, посетите <http://creativecommons.org/licenses/by/4.0/>

debugged in a higher-level language, and then the controller operation code for the microprocessor is generated. The article provides an overview of robust and adaptive algorithms that can be used in on-board automatic control systems. Computer models of control systems were built using these algorithms in MATLAB and SimInTech packages, as examples of the initial stages of model-based design technology.

Keywords: *model-based design; automatic control systems; robust algorithms; adaptive algorithms*

For citation: Oshchepkov, A.Yu., Vlasov, A.N. (2024), "Computer Modeling of Robust and Adaptive Algorithms in Model-based Design Technology", *Bulletin of Perm University. Mathematics. Mechanics. Computer Science*. no. 1(64), pp. 43-52. DOI: 10.17072/1993-0550-2024-1-43-52.

The article was submitted 24.01.2024; approved after reviewing 02.02.2024; accepted for publication 18.03.2024.

Введение

Современные автоматические цифровые регуляторы, обеспечивающие работу автономных устройств, отдельных узлов промышленных установок, мобильных роботов, должны использовать робастные или адаптивные алгоритмы. Если алгоритм с предварительно настроенными параметрами достигает цели управления при изменяющихся условиях, он называется робастным алгоритмом [1]. Алгоритм, в котором заложена корректировка параметров для наиболее точного управления в условиях неопределенностей, считается адаптивным [2].

Для встраиваемых систем алгоритмы прошиваются в микропроцессор бортового контроллера в виде кода, получаемого компиляцией программы, написанной обычно на языке С или его потомков (С++, С# и др.). Пропорционально-интегрально-дифференциальный (ПИД) закон регулирования, представляющий пример робастного алгоритма, предполагает использование трех постоянных коэффициентов и применение арифметических действий умножения и сложения, поэтому он широко применяется на практике. В то же время для управления в условиях неопределенностей разработаны более эффективные алгоритмы, хотя и более сложные в математическом плане, но не предполагающие использования искусственного интеллекта, основанные на применении градиентных методов, методов экстремального регулирования, управления с эталонной моделью, идентификационных адаптивных и других методов.

Для этих алгоритмов непосредственное написание программы на языке С представляет громоздкую процедуру, что ограничивает их практическое применение во встраиваемых системах управления. На помощь приходит методика модельно-ориентированного проектирования (МОП-технологии).

Суть методики заключается в том, что сначала составляется и отлаживается имитационная компьютерная модель САУ на языке более высокого уровня, а затем генерируется код работы регулятора для микропроцессора.

В данной статье рассказывается о содержании основных этапов МОП-технологии, дан обзор адаптивных и робастных алгоритмов, которые могли бы применяться в бортовых системах управления и приведены примеры моделирования этих алгоритмов в пакетах MATLAB+Simulink и SimInTech.

Полученные модели могут быть использованы для генерации С-кода и последующей прошивки в микроконтроллер.

1. Этапы МОП-технологии

Последовательность выполнения этапов модельно-ориентированного проектирования приведена на рис. 1. На каждом этапе применяется свое программное обеспечение. Например, в Центре Инженерных Технологий и Моделирования (ЦИТМ "Экспонента") используются возможности пакета MATLAB. Здесь МОП-технология хорошо зарекомендовала себя при разработке различных сложных систем, в частности встраиваемых систем управления, цифровой обработки сигналов, компьютерного зрения, радиолокации и связи [3]. Российской компанией ООО "ЗВ Сервис" разработана среда динамического моделирования технических систем SimInTech (Simulation In Technic). Среда также предназначена для расчетной проверки работы систем управления сложными техническими объектами, описание динамики которых может быть представлено в виде системы дифференциально-алгебраических уравнений и/или реализовано методами структурного моделирования [4].

Рассмотрим этапы методологии МОП на примере ее применения в ЦИТМ "Экспонента".

Архитектура системы создается в виде модели, описания компонентов, их свойств, интерфейсов между компонентами, а также анализа свойств системы. Для этого используется программное обеспечение **System Composer**, которое позволяет создавать или импортировать модели архитектуры, описывающие систему в виде компонентов и интерфейсов. При помощи пользовательских представлений архитектуры могут быть изучены отдельные ограничения проектирования или анализа.

Архитектурные модели позволяют анализировать требования, собирать свойства с помощью стереотипов, а также создавать спецификации и описания интерфейсов.

Исполнение системной модели осуществляется с помощью программного обеспечения **Simulink**. Simulink – среда динамического междисциплинарного моделирования сложных технических систем и основной инструмент для модельно-ориентированного проектирования. Его основным интерфейсом является графический инструмент для построения диаграмм и настраиваемый набор библиотек блоков. Он предлагает тесную интеграцию с остальной средой MATLAB и может либо использовать MATLAB, либо создавать сценарии из него. Simulink широко используется в автоматическом управлении и цифровой обработке сигналов для многодоменного моделирования и проектирования на основе моделей.



Рис. 1. Этапы МОП-технологии

Проектирование алгоритмов может быть проведено в виде программ М-языка или в виде графических диаграмм. Для анализа и настройки алгоритмов применяется ПО **Simulink Control Design**, которое позволяет проектировать и анализировать системы управления, смоделированные в Simulink. Можно выполнять автоматическую настройку систем с SISO и MIMO архитектурами, включая ПИД-регуляторы. Алгоритм автоматической настройки ПИД-регуляторов может быть

перенесен в программное обеспечение встраиваемых систем для проведения настройки коэффициентов регуляторов в режиме реального времени.

Моделирование физических элементов включает в себя моделирование компонентов и окружения, как составных частей системы. При проектировании или выборе компонента для использования в системе крайне важным является анализ взаимовлияния этих подсистем или компонентов друг на друга.

МОП позволяет в одной модели объединить все подсистемы и провести виртуальные комплексные испытания изделия. Это означает, что при разработке систем управления можно воссоздать физическую модель объекта управления, содержащую механическую, гидравлическую, пневматическую, электрическую и другие подсистемы. А при разработке систем цифровой обработки сигналов можно описать канал передачи данных, радиочастотный тракт, радиолокационную обстановку, антенную решетку или даже задать помехи, наводимые потенциальным противником.

Для создания моделей физических систем в среде Simulink используется специальный блок **Simscape**.

Моделируются такие системы, как электродвигатели, мостовые выпрямители, гидроприводы и системы охлаждения. Системы управления для физической системы разрабатываются в Simulink.

При запуске алгоритмов Simscape поддерживает генерацию С-кода для запуска физических моделей на машинах реального времени и проведения программно-аппаратного тестирования (Hardware-in-the-Loop – HIL) блока управления.

Для генерации кода из моделей имеется широкий набор инструментов, позволяющих автоматически генерировать исходный код для различных микропроцессоров, операционных систем реального времени, программируемых логических интегральных схем (ПЛИС), программируемых логических контроллеров (ПЛК) и встраиваемых графических процессоров (GPU).

Такой подход к разработке встраиваемых алгоритмов исключает ошибки ручного кодирования, гарантирует быстрое внесение изменений в проект, обеспечивает быстрый перенос алгоритмов с одной элементной базы на другую и с проекта на проект. ПО **Embedded Coder** генерирует читаемый, компактный и быстрый C/C++ код для встраиваемых процессоров, используемых в массовом производстве. Embedded Coder проводит оптимизации, которые повышают эффективность кода.

Тестирование кода на моделях осуществляется с помощью программного продукта **Simulink Design Verifier**.

Стенды полунатурного моделирования снижают риски при проведении испытаний и экономят бюджет при проектировании систем

за счет использования HIL-технологии. Используемое ПО – **Simulink Real-Time**.

Модельно-ориентированное проектирование – это процесс постоянной верификации и валидации при разработке встраиваемых систем. Систематическое использование моделей позволяет на разных этапах жизненного цикла проекта проводить и повторять испытания, убеждаясь, что система функционирует в соответствии с заложенными требованиями. Удобные инструменты позволяют хранить все тестовые сценарии в удобном виде, сравнивать и анализировать результаты испытаний, а также формально доказывать, что применение инструментов разработки было корректным и не привело к внесению ошибок в проект. Используется программное обеспечение **Simulink Test**.

Ниже рассмотрим выполнение начальных этапов МОП-технологии (1-3) на примере составления математических моделей адаптивных и робастных алгоритмов, которые могли бы применяться в бортовых системах управления и приведем результаты моделирования этих алгоритмов в пакетах MATLAB+Simulink и SimInTech.

2. Робастные и адаптивные алгоритмы

Для демонстрации возможностей различных алгоритмов структуру системы (этап 1 МОП) автоматического управления выберем в простейшей форме системы с обратной связью, как это показано на рис. 2.



Рис. 2. Структура системы автоматического управления

На вход системы подается требуемое значение регулируемой величины (уставка) $U_{уст}$, выходной сигнал объекта y по цепи обратной связи поступает на блок сравнения (компаратор), значение отклонения выхода от требуемой величины на входе (рассогласование, невязка, ошибка) $e = U_{уст} - y$ передается в устройство управления, в котором и реализован тот или иной алгоритм. Сигнал управления и подается на вход объекта управления.

Примеры робастных и адаптивных алгоритмов приведены в табл. 1. Рассмотрим

математические модели некоторых из этих алгоритмов, работу которых затем промоделируем в компьютере.

Таблица 1. Примеры алгоритмов

Наименование алгоритма	Вид алгоритма	Тип САУ
ПИД-закон нерекуррентный	Робастный	Непрерывная/дискретная
ПИД-закон рекуррентный	Робастный	Дискретная
Градиентный метод [5]	Робастный	Дискретная
Экстремальные методы [2]	Адаптивный	Непрерывная
Управление с эталонной моделью [6]	Адаптивный	Непрерывная
Быстрый идентификационный алгоритм [7]	Адаптивный	Дискретная

2.1. Цифровой ПИД-регулятор

Многолетняя практика автоматизации показала, что подавляющее большинство задач синтеза одноканального управления может быть решено с помощью регуляторов по отклонению, учитывающих текущее значение ошибки, первую производную ошибки и интеграл от ошибки за время регулирования. В настоящее время пропорционально-интегрально-дифференциальный (ПИД) закон и его частные случаи считаются стандартными линейными законами регулирования, наиболее часто используемыми в промышленных регуляторах. ПИД-регулятор представляет собой пример универсального робастного регулятора, используемого в системах управления с обратной связью.

Регулятор имеет три настроечных параметра: K_1 , K_2 , K_3 , управляющее воздействие $u(t)$, которое вырабатывает ПИД-регулятор в непрерывной системе, описывается выражением

$$u(t) = K_1 e(t) + K_2 \int_0^t e(\tau) d\tau + K_3 \frac{de(t)}{dt}, \quad (1)$$

где $e(t) = y_{\text{уст}} - y(t)$ – сигнал ошибки, поступающий на вход регулятора.

После дискретизации системы с периодом квантования (шагом дискретизации) T по схеме Эйлера, в которой интеграл в (1) заменяется суммой дискретных отсчетов, а производная – разностью двух последовательных отсчетов, деленной на шаг T , для управляющего воздействия, рассчитываемого в k -й момент времени и выводимого на объект, получим выражение:

$$u_k = K_P e_k + K_I \sum_{j=0}^k e_j + K_D (e_k - e_{k-1}); \quad (2)$$

$$K_P = K_1, K_I = K_2 T, K_D = K_3 / T.$$

Этот закон представляет собой *нерекуррентный алгоритм работы дискретного ПИД-регулятора*, который требует вычисления больших сумм.

Если записать полученное выражение для $(k+1)$ -го момента времени и из него вычесть выражение (2), то получим *рекуррентный алгоритм дискретного ПИД-регулятора*:

$$u_{k+1} = u_k + K_P (e_{k+1} - e_k) + K_I e_{k+1} + K_D (e_{k+1} - e_k - (e_k - e_{k-1})). \quad (3)$$

Как видим, данный алгоритм по-прежнему предполагает использование трех коэффициентов: K_P, K_I, K_D , – значения которых определяются при настройке регулятора.

При достижении цели регулирования ($e = 0$) управляющее воздействие достигает некоторого постоянного значения, компенсируя внешние воздействия на объект управления.

2.2. Градиентный метод

Математическую модель дискретной системы управления представим в виде

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{y}_k = \mathbf{x}_k, \quad (4)$$

где $k = 0, 1, 2, \dots$ – номер стадии функционирования дискретной системы или номер очередного момента измерения и подачи управления t_k ; $\mathbf{x}_k \in R_n$ – вектор переменных состояния, $\mathbf{u}_k \in R_m$ – вектор управляющих воздействий, соответствующих моменту t_k , $\mathbf{y}_k$ – вектор выхода. Градиентный метод [3] предназначен для построения управления дискретной системой (4), когда цель управления задана при помощи некоторой гладкой неотрицательной *целевой функции* $Q = Q(\mathbf{x})$ в виде

$$Q(\mathbf{x}_{k+1}) \leq \Delta, \quad \text{при } k > k_*, \quad (5)$$

где $\Delta > 0$ – заданное значение порога точности задачи.

В качестве целевой функции, может быть выбрана функция дискретного аргумента

$$Q(x_{k+1}) = \frac{1}{2} \langle x_{k+1} - x_*, x_{k+1} - x_* \rangle \quad (6)$$

(здесь угловые скобки означают скалярное произведение векторов).

Подставив очередное состояние объекта из (5) в (6), получим приведенную целевую функцию, непосредственно зависящую от управления $Q_k(u) = Q(f(x_k, u))$. Градиентный метод основан на изменении вектора u_k в направлении, противоположном направлению градиента (вектора из частных производных) от функции Q_k по управляющим переменным

$$u_{k+1} = u_k - \gamma_k \nabla_u Q_k(u_k), \quad (7)$$

где $\nabla_u = \left\{ \frac{\partial}{\partial u_1}, \dots, \frac{\partial}{\partial u_m} \right\}^T$ – вектор градиента, $\gamma_k \geq 0$ – коэффициент шага по управлению. Идея метода основана на том, что при малом γ_k поправка $\Delta u_k = u_{k+1} - u_k$ приводит к уменьшению значения функции $Q_k(u)$, поскольку на следующем шаге управления целевая функция изменяется в силу динамики системы. Одним из условий, гарантирующих достижение цели управления (5), является выбор коэффициента шага γ_k с зоной нечувствительности,

$$\text{если } Q_k(u_k) \leq \Delta, \text{ то } \gamma_k = 0. \quad (8)$$

Условие (8) означает, что при достижении цели регулирования вектор управления компенсирует внешнее воздействие, не изменяясь по величине, аналогично алгоритму ПИД-регулятора (3).

Для дискретного линейного объекта первого порядка уравнение движения можно представить в виде:

$$x_{k+1} = Ax_k + Bu_k + F, \quad (9)$$

где A , B и F – некоторые константы. При $x_* = 0$, целевая функция (6) принимает простой вид

$$Q(e_{k+1}) = \frac{1}{2} e_{k+1}^2, \quad (10)$$

$$e_{k+1} = y_{\text{уст}} - y_{k+1} = -x_{k+1}.$$

Частная производная функции (10) по переменной u равна

$$\frac{\partial Q_k(u)}{\partial u} = (Ax_k + Bu_k + F) \cdot B = -Be_{k+1}.$$

Подставляя это выражение в формулу (7) получаем закон управления

$$u_{k+1} = u_k + \gamma_k \cdot B \cdot e_{k+1}, \quad (11)$$

с учетом условия (8), имеющего в данном случае вид:

$$\text{если } \frac{1}{2} e_{k+1}^2 \leq \Delta, \text{ то } \gamma_k = 0. \quad (12)$$

Таким образом, применение градиентного метода приводит к алгоритму, аналогичному ПИД-закону (3) с $K_p = K_D = 0$ (интегральный регулятор), однако условие (12) делает этот алгоритм нелинейным.

2.3. Быстрый идентификационный алгоритм

Рассмотрим дискретную систему первого порядка, номинальная модель которой имеет вид (9) с начальным условием $x_0 = \tilde{x}_0 \neq 0$. Цель регулирования выберем в виде (5) с целевой функцией (10). При $\Delta \rightarrow 0$ цель регулирования можно записать в простой форме:

$$x_{k+1} \rightarrow 0, \text{ при } k > k_*. \quad (13)$$

Если параметры номинальной модели соответствуют реальной системе, цель регулирования (13) может быть достигнута за один шаг с помощью управляющего воздействия:

$$u_k = -\frac{1}{B} (Ax_k + F), \quad (14.1)$$

поэтому этот алгоритм является самым быстрым из всех возможных. Подстановка (14.1) в (9) даст $x_{k+1} = 0$, поэтому для дальнейших шагов получим

$$\begin{aligned} u_{k+1} &= -\frac{F}{B}, \\ u_{k+2} &= u_{k+1}, \\ k &= 0, 1, 2, \dots \end{aligned} \quad (14.2)$$

Алгоритм управления (14) назовем *быстрым алгоритмом*. Если возможности управления ограничены и управляющее воздействие, рассчитанное по формуле (14.1), превосходит максимально возможное значение U_{max} , на вход объекта управления должна подаваться эта величина с соответствующим знаком, что обеспечит движение системы в нужном направлении, аналогично градиентному методу.

Если параметры системы (9) неизвестны, их можно найти, применив в течение нескольких шагов любой устойчивый робастный алгоритм (предполагается, что параметры системы в течение этих шагов не изменяются). В результате мы получим *идентификационный адаптивный алгоритм управления на основе быстрого алгоритма*.

Будем использовать П-регулятор на первых трех шагах управления: $u_k = -K_p x_k$, $k = 0, 1, 2$. В результате, в соответствии с моделью (9), получим три выражения:

$$\begin{aligned} x_1 &= Ax_0 + Bu_0 + F, \\ x_2 &= Ax_1 + Bu_1 + F, \\ x_3 &= Ax_2 + Bu_2 + F. \end{aligned} \quad (15)$$

Здесь значения входов объекта u_k и измеряемых выходов x_k , $k = 0, 1, 2, 3$ – являются известными величинами, а три параметра A , B , F составляют вектор неизвестных параметров, подлежащих идентификации.

Введем векторы $|\xi\rangle$, $|Y\rangle$ и матрицу G_3 по правилам:

$$|\xi\rangle = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} \equiv \begin{pmatrix} A \\ B \\ F \end{pmatrix}, \quad |Y\rangle = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},$$

$$G_3 = \begin{pmatrix} x_0 & u_0 & 1 \\ x_1 & u_1 & 1 \\ x_2 & u_2 & 1 \end{pmatrix},$$

тогда систему уравнений (15) можно записать в виде

$$G_3 \cdot |\xi\rangle = |Y\rangle. \quad (16)$$

Вектор неизвестных параметров определится после решения (16):

$$|\xi\rangle = G_3^{-1} \cdot |Y\rangle.$$

Определив параметры A , B , F , можем синтезировать быстрый алгоритм (14.1) при $k = 3$. С целью фильтрации возможных ошибок измерений алгоритм идентификации позволяет использовать n начальных шагов, где $n \geq 3$. В этом случае вместо системы трех уравнений (16) мы получим следующую систему:

$$G_n \cdot |\xi\rangle = |Y\rangle,$$

$$|Y\rangle = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad (17)$$

$$G_n = \begin{pmatrix} x_0 & u_0 & 1 \\ x_1 & u_1 & 1 \\ \dots & \dots & \dots \\ x_{n-1} & u_{n-1} & 1 \end{pmatrix}.$$

Система (17) содержит число уравнений, превосходящее количество неизвестных, т.е. является несовместной. Для решения используется метод наименьших квадратов или другие методы линейного регрессионного анализа.

Заметим, что произведение матрицы G_n из (17) на транспонированную матрицу $\tilde{G}_n$ даст квадратную симметричную матрицу размерности 3×3 , для которой можно найти обратную матрицу.

Умножив обе части системы (17) на $\tilde{G}_n$ и проведя операцию обращения квадратной матрицы, найдем вектор $|\xi\rangle$:

$$|\xi\rangle = (\tilde{G}_n G_n)^{-1} \cdot \tilde{G}_n \cdot |Y\rangle. \quad (18)$$

Следует, однако, иметь в виду, что при достижении цели регулирования (13) первый столбец матрицы G_n будет состоять из нулей, поэтому расчеты по формуле (18) становятся невозможными. С другой стороны, достижение цели регулирования говорит о правильной идентификации параметров объекта, следовательно, их не нужно рассчитывать, а следует брать их предыдущие значения.

Аналогично (8) нужно назначить значение порога точности задачи $\Delta > 0$, а целевую функцию на каждом шаге определять как сумму квадратов переменных состояния на всех предшествующих n шагах этапа идентификации $Q_k = \sum_{j=1}^n x_{k+j}^2$.

Полностью работа идентификационной адаптивной системы управления на основе быстрого алгоритма должна быть описана следующим образом:

$$u_{k+1} = -K \cdot x_{k+1} - V, \quad (19.1)$$

$$K = \frac{A_{k+1}}{B_{k+1}}, \quad V = \frac{F_{k+1}}{B_{k+1}}.$$

Здесь A_{k+1} , B_{k+1} , F_{k+1} – значения параметров системы на $(k+1)$ -м шаге, определенные по алгоритму:

если $Q_k > \Delta$, то

$A_{k+1} = \xi_1$, $B_{k+1} = \xi_2$, $F_{k+1} = \xi_3$, вектор $|\xi\rangle$ определяется по формуле (18);

если $Q_k \leq \Delta$, то

$$A_{k+1} = A_k, \quad B_{k+1} = B_k, \quad F_{k+1} = F_k. \quad (19.2)$$

3. Компьютерное моделирование робастных и адаптивных алгоритмов для встраиваемых систем

Следующий этап МОП-технологии после составления математической модели алгоритмов – создание компьютерных моделей алгоритмов и их запуск [8].

Модель системы управления имела вид, представленный на рис. 2. Объект управления моделировался уравнением движения (9) со следующими параметрами ABCD-модели системы: $A = 0, 1$, $B = 0, 1$, $C = 1$, $D = 0$.

3.1. Рекуррентный цифровой ПИД-регулятор

На рис. 3а представлен блок рекуррентного ПИД-регулятора (3), созданный в SimInTech [9].

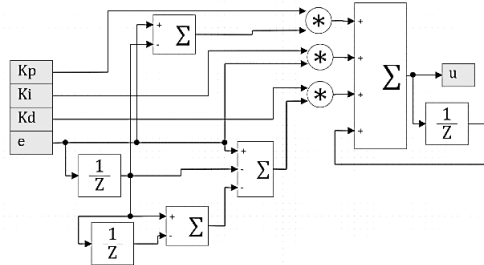


Рис. 3а. Компьютерная модель рекуррентного цифрового ПИД-регулятора в SimInTech

На рис. 3б показана полученная в результате моделирования переходная характеристика системы управления, созданная с использованием этого блока.

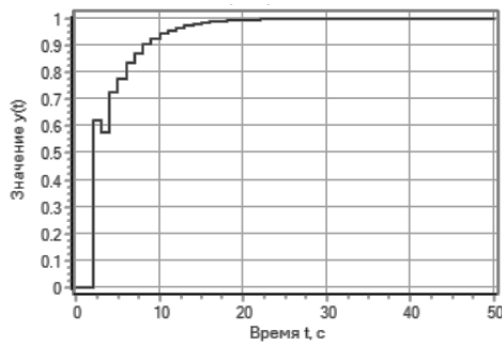


Рис. 3б. Переходная характеристика системы управления с рекуррентным ПИД-регулятором

3.2. Градиентный алгоритм

На рис. 4а представлена модель системы управления на основе градиентного алгоритма (11)–(12).

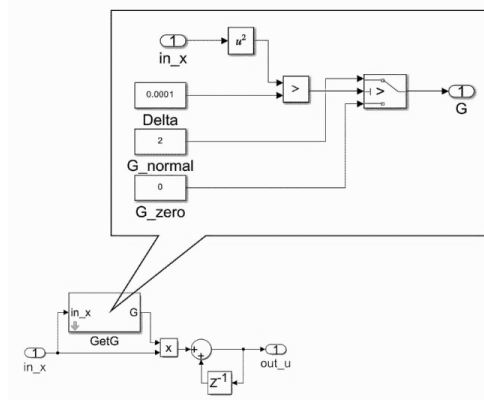


Рис. 4а. S-модель блока управления на основе градиентного алгоритма

На рис. 4б показана переходная характеристика системы, в качестве управляющего блока использующая градиентный алгоритм.

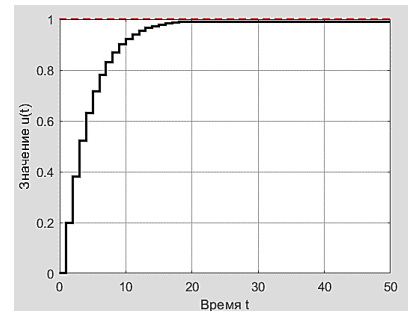


Рис. 4б. Переходная характеристика системы с блоком управления на основе градиентного метода

3.3. Идентификационный быстрый алгоритм

На рис. 5а представлена модель адаптивной системы управления на основе быстрого идентификационного алгоритма.

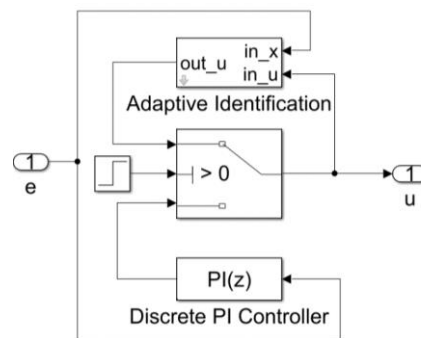


Рис. 5а. S-модель адаптивной системы управления на основе идентификационного быстрого алгоритма

На рис. 5б изображена переходная характеристика системы управления на основе данного алгоритма.

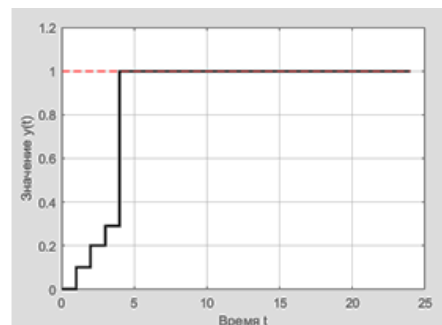


Рис. 5б. Переходная характеристика системы управления на основе идентификационного быстрого алгоритма

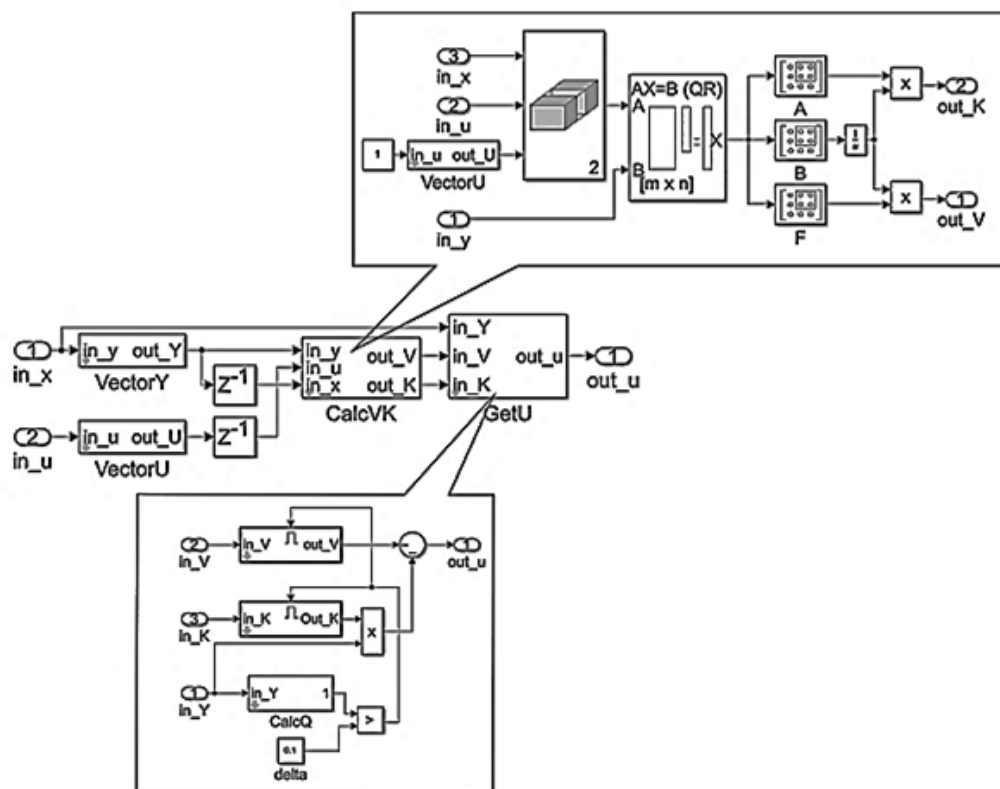


Рис. 6. S-модель блока расчета управляющего воздействия в адаптивной системе на основе идентификационного быстрого алгоритма

На рис. 6 изображена модель блока управления на основе идентификационного быстрого алгоритма.

Заключение

Описана технология модельно-ориентированного проектирования, применяемая при разработке встраиваемых устройств автоматического управления.

Приведены примеры робастных и адаптивных алгоритмов для таких устройств и их моделирования с использованием пакетов MATLAB+Simulink и SimInTech, как реализация первых этапов МОП-технологии.

Полученные результаты позволяют сделать вывод, что разработанные компьютерные модели алгоритмов могут быть применены для управления конкретными объектами с помощью встраиваемых устройств.

Список источников

1. Поляк Б.Т. Математическая теория автоматического управления: учебное пособие / Б.Т. Поляк, М.В. Хлебников, Л.Б. Рапопорт. М.: ЛЕНАНД, 2019. 500 с.
2. Никифоров В.О. Управление в условиях неопределенности: чувствительность, адаптация, робастность / В.О. Никифоров, А.В. Ушаков. СПб.: СПб ГИТМО (ТУ), 2002. 232 с.
3. URL: <https://exponenta.ru/mbd> (дата обращения: 03.01.2024).
4. URL: <https://3v-services.com> (дата обращения: 03.01.2024).
5. Фрадков А.Л. Кибернетическая физика: принципы и примеры. СПб.: Наука, 2003. 208 с.
6. Никифоров В.О. Адаптивное и робастное управление с компенсацией возмущений. СПб.: Наука, 2003. 282 с.
7. Ощепков А.Ю. Системы автоматического управления: теория, применение, моделирование в MATLAB: учеб. пособие для студентов физического факультета. 4-е изд., стер. СПб.: Изд-во "Лань", 2021. 208 с.
8. Деменков Н.П. Модельно-ориентированное проектирование систем управления // Промышленные АСУ и контроллеры. 2008. № 10. С. 66–69.
9. Хабаров С.П., Шилкина М.Л. Основы моделирования технических систем. Среда Simintech: учеб. пособие. СПб.: Лань, 2022. 120 с.

References

1. Polyak, B.T., Hlebnikov, B.T. (2019), "Matematicheskaya teoriya avtomaticheskogo upravleniya: uchebnoe posobie" [Mathematical theory of automatic control: textbook], LE-NAND, Moscow, Russia. 500 p.
2. Nikiforov, V.O., Ushakov, V.O. (2002), "Upravlenie v usloviyah neopredelyonnosti: chuvstvitel'nost', adaptatsiya, robustnost' " [Control under conditions of uncertainty: sensitivity, adaptation, robustness], SPb GITMO (TU), St. Petersburg, Russia. 232 p.
3. CITM Ekspozenta. "Model-based design", available at <https://exponenta.ru/mbd> (accessed 3 January 2024).
4. 3VS Software Development. "3VSERVICES", available at <https://3v-services.com> (accessed 3 January 2024).
5. Fradkov, A.L. (2003), "Kiberneticheskaya fizika: principy i primery" [Cybernetic physics: principles and examples], Nauka, St. Petersburg, Russia. 208 p.
6. Nikiforov, V.O. (2003), "Adaptivnoe i robustnoe upravlenie s kompensatsiey vozmushcheniy" [Adaptive and robust control with disturbance compensation], Nauka, St. Petersburg, Russia. 208 p.
7. Oshchepkov, A.Yu. (2021), "Sistemy avtomaticheskogo upravleniya: teoriya, primeneniye, modelirovaniye v MATLAB": Uchebnoe posobie dlya studentov fizicheskogo fakul'teta [Automatic control systems: theory, application, modeling in MATLAB: A textbook for students of the Faculty of Physics], 4th ed., Lan', St. Petersburg, Russia. 180 p.
8. Demenkov, N.P. (2008), "Modelno-orientirovannoe proektirovaniye sistem upravleniya" [Model-Based Design of Control Systems], Promyshlennyye ASU i kontrolyery. vol. 10. 66-69 pp.
9. Habarov, S.P., Shilkina M.L (2022), "Osnovy modelirovaniya tehnikeskikh sistem. Sreda Simintech: uchebnoe posobie" [Fundamentals of modeling technical systems. Simintech environment: tutorial], Lan', St. Petersburg, Russia. 120 p.

Информация об авторах:

А. Ю. Ощепков – кандидат физико-математических наук, доцент, профессор кафедры радиоэлектроники и защиты информации Пермского государственного национального исследовательского университета (614068, Россия, г. Пермь, ул. Букирева, 15), член международного общества по физике и управлению (International Physics And Control Society – IPACS), AuthorID: 175649;

А. Н. Власов – аспирант физического факультета Пермского государственного национального исследовательского университета (614068, Россия, г. Пермь, ул. Букирева, 15).

Information about the authors:

Alexander Yu. Oshchepkov – Candidate of Physical and Mathematical Sciences, Associate Professor, Professor of the Department of Radio Electronics and Information Security, Perm State University, (Bukireva St., 15, Perm, Russia, 614068), member of IPACS, AuthorID: 175649;

Alexander N. Vlasov – Postgraduate Student of the Department of Radio Electronics and Information Security, Perm State University (Bukireva St., 15, Perm, Russia, 614068).