

УДК 378.147.88:004.942:681.5
DOI 10.17513/snt.40914



CC BY 4.0

ПРОЕКТНО-ЛАБОРАТОРНАЯ МЕТОДИКА ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ СТУДЕНТОВ С ПРИМЕНЕНИЕМ ЦИФРОВОГО ДВОЙНИКА

Сафаров И. М.

*Федеральное государственное бюджетное образовательное учреждение высшего образования
«Казанский государственный энергетический университет»,
Казань, Российская Федерация, e-mail: ildarsafarov@mail.ru*

Цель исследования состояла в разработке и экспериментальной проверке проектно-лабораторной методики формирования профессиональных компетенций студентов направления 15.03.04 «Автоматизация технологических процессов и производств» с применением учебного цифрового двойника и моделирования программируемого логического контроллера. Исследование проводилось в 2025/2026 учебном году на базе Казанского государственного энергетического университета. В педагогическом квазиэксперименте участвовали 46 студентов третьего курса: по 23 чел. в контрольной и экспериментальной группах. Использовались тестирование, экспертная оценка выполненных работ и статистическая обработка результатов. В контрольной группе студенты выполняли традиционные лабораторные работы, в экспериментальной те же действия были объединены в целостный проект: анализ технологической задачи, разработка алгоритма, программирование, запуск модели, поиск ошибок и отладка. После завершения модуля средний итоговый результат составил $57,8 \pm 10,5$ балла в контрольной и $74,7 \pm 9,2$ балла в экспериментальной группе; различия статистически значимы. Наиболее выраженная разница отмечена по алгоритмическому, программно-практическому и проектно-диагностическому критериям. Полученные результаты показывают потенциальную эффективность комплексной проектно-лабораторной методики. При этом самостоятельный вклад учебного цифрового двойника отдельно не оценивался, поэтому эффект относится к методике в целом.

Ключевые слова: инженерное образование, автоматизация технологических процессов, учебный цифровой двойник, проектно-лабораторная подготовка, профессиональные компетенции, цифровые технологии обучения

PROJECT-LABORATORY METHOD OF DEVELOPING STUDENTS' PROFESSIONAL COMPETENCIES USING A DIGITAL TWIN

Safarov I. M.

*Federal State Budgetary Educational Institution of Higher Education
“Kazan State Power Engineering University”, Kazan, Russian Federation,
e-mail: ildarsafarov@mail.ru*

The aim of the study was to develop and experimentally test a project-based laboratory methodology for developing professional competencies of students majoring in 15.03.04 “Automation of Technological Processes and Production” using an educational digital twin and programmable logic controller simulation. The study was conducted during the 2025/2026 academic year at Kazan State Power Engineering University. The pedagogical quasi-experiment involved 46 third-year students, with 23 students in the control group and 23 in the experimental group. Testing, expert evaluation of completed work, and statistical analysis of the results were used. In the control group, students performed traditional laboratory work; in the experimental group, the same activities were combined into a holistic project involving analysis of the technological task, algorithm development, programming, model launch, error detection, and debugging. After the module, the mean final score was 57.8 ± 10.5 points in the control group and 74.7 ± 9.2 points in the experimental group; the differences were statistically significant. The most pronounced differences were found for the algorithmic, software-practical, and project-diagnostic criteria. The results indicate the potential effectiveness of the comprehensive project-based laboratory methodology. The independent contribution of the educational digital twin was not assessed separately; therefore, the observed effect refers to the methodology as a whole.

Keywords: engineering education, process automation, educational digital twin, project-based laboratory training, professional competencies, digital learning technologies

Введение

В условиях цифровой трансформации промышленности и активного внедрения технологий Индустрии 4.0 возрастают требования к подготовке инженеров в области автоматизации технологических процессов и производств. Современный специалист

должен обладать не только теоретическими знаниями о принципах функционирования автоматизированных систем управления, но и навыками моделирования технологических объектов, разработки алгоритмов управления, программирования промышленных контроллеров и анализа работы создаваемых систем [1–3].

Использование цифровых двойников и виртуальных производственных сред сегодня является одним из направлений развития инженерного образования. Они позволяют моделировать работу оборудования, проверять алгоритмы управления и разбирать производственные ситуации без постоянного доступа к реальным промышленным установкам [4–6]. Студенту это дает возможность увидеть последствия ошибки, изменить алгоритм и снова проверить работу системы.

Отдельно стоит моделирование программируемых логических контроллеров. Просто изучить команды контроллера недостаточно – важно видеть весь процесс целиком: поступление сигнала, его обработку, управляющую команду и реакцию механизма. Поэтому виртуальная производственная среда и моделирование контроллера хорошо дополняют друг друга [7–9].

В исследовании учебный цифровой двойник понимался не как точная копия конкретного промышленного объекта, а как специально подготовленная модель типового технологического процесса. В ней были оборудование, датчики, исполнительные механизмы, входные и выходные сигналы и сама логика переходов между состояниями. То есть студент видел не отдельный элемент, а всю систему целиком. Состояние виртуального объекта менялось – формировался сигнал для контроллера. Контроллер этот сигнал обрабатывал и возвращал уже управляющую команду обратно в модель. После этого объект снова менял состояние. По сути, получалась обычная цепочка управления, только полностью в виртуальной среде.

При этом цифровой двойник специально не делали слишком сложным. Он не был постоянно связан с реальной промышленной установкой и не получал от нее текущие данные. Для учебной задачи это и не требовалось. Какие-то параметры можно было убрать, чтобы не перегружать работу, а аварийные ситуации, наоборот, запускать снова и снова – столько раз, сколько нужно, чтобы студент сам нашел ошибку и понял, почему система отработала именно так.

В научных работах цифровые двойники, виртуальные лаборатории и проектное обучение рассматриваются как перспективные инструменты подготовки инженерных кадров [10–12]. При этом чаще они описываются отдельно. Вопрос о том, как объединить модель технологического объекта, программирование контроллера, поиск ошибок и защиту решения в одну последовательную учебную работу, представлен заметно меньше.

Поэтому в работе рассматривалась методика, где учебный цифровой двойник, моделирование программируемого логического контроллера и проектно-лабораторная работа используются в рамках одной задачи – от анализа объекта до проверки и защиты полученного решения.

Научная новизна исследования состоит в объединении учебного цифрового двойника, моделирования программируемого логического контроллера и проектной работы в одну последовательную методику. Студент проходил весь путь: анализировал технологическую задачу, определял входные и выходные сигналы, составлял алгоритм, программировал контроллер, запускал систему, находил ошибки и защищал полученное решение.

Под проектно-лабораторной методикой в данной работе понимается такая организация практической подготовки, при которой отдельные лабораторные действия не выполняются сами по себе, а соединяются в один проект. Студент разбирает технологический объект, составляет алгоритм управления, программирует контроллер, запускает систему, исправляет ошибки и объясняет принятое решение на защите.

Цель исследования – разработка и экспериментальная проверка проектно-лабораторной методики формирования профессиональных компетенций студентов направления 15.03.04 «Автоматизация технологических процессов и производств» с применением учебного цифрового двойника и моделирования ПЛК.

Материалы и методы исследования

Исследование проводилось в течение 2025/2026 учебного года на кафедре «Автоматизация технологических процессов и производств» ФГБОУ ВО «Казанский государственный энергетический университет» в рамках дисциплины «Программное обеспечение систем управления». Использовались педагогический квазиэксперимент, тестирование, экспертная оценка и статистическая обработка данных. Сначала по входной диагностике проверяли, насколько группы сопоставимы между собой, а потом уже сравнивали результаты до и после формирующего этапа.

Всего участвовали 46 студентов третьего курса направления 15.03.04 «Автоматизация технологических процессов и производств». В контрольной группе было 23 чел. и в экспериментальной тоже 23.

Полной рандомизации не было, потому что работали с уже сформированными учебными группами. При этом условия старались оставить одинаковыми: занятия

вел один преподаватель, использовалась одна рабочая программа, объем аудиторной и самостоятельной работы не отличался. То есть основное различие между группами было именно в организации практической работы, а не в количестве занятий или содержании дисциплины.

Исходную сопоставимость групп оценивали по результатам входной диагностики. Формирующий этап продолжался 12 недель: 36 ч аудиторной и 18 ч самостоятельной работы. В контрольной группе студенты выполнили шесть традиционных лабораторных работ в CODESYS Control Win V3 по готовым схемам и инструкциям. Задания включали работу с входными и выходными сигналами, команды запуска и остановки, таймеры и счетчики, последовательный алгоритм, аварийный останов и поиск ошибок.

В экспериментальной группе те же по содержанию и сложности действия соединялись в один проект. Здесь использовались Factory I/O [13], CODESYS Control Win V3 [14] и обмен данными по протоколу Modbus TCP [15]. Студент работал уже со всей системой: анализировал объект, составлял алгоритм, программировал контроллер, запускал модель и разбирался с ошибками.

В качестве виртуального технологического объекта использовался конвейерно-сортировочный участок *Sorting by Height (Advanced)* в Factory I/O. Коробки поступали на линию и распределялись в зависимости от высоты. Для работы участка требовалось согласованное управление датчиками, конвейерами и поворотным столом.

Управляющая программа разрабатывалась в CODESYS Control Win V3. Связь между CODESYS и Factory I/O выполнялась через Modbus TCP: виртуальный датчик передавал сигнал программному контроллеру, программа его обрабатывала и возвращала управляющую команду в модель. Если алгоритм был написан неправильно, ошибка проявлялась в поведении участка, а не только в коде.

К входным сигналам относились запуск, остановка, сброс и аварийный останов, а также сигналы датчиков наличия и высоты коробки, положения поворотного стола и состояния выходных участков. Такой набор позволял студенту видеть текущее состояние объекта и учитывать его в алгоритме управления.

Выходные сигналы управляли подающим и входным конвейерами, загрузкой, поворотом и разгрузкой стола, а также левым и правым выходными конвейерами. За счет этого можно было проследить всю

цепочку: входной сигнал – условие в программе – команда контроллера – действие исполнительного механизма.

Кроме обычного режима сортировки задавались ситуации, в которых студенту нужно было разбираться с нарушением работы системы: занятисть поворотного стола или выходного конвейера, аварийный останов, сбой датчика и блокирование коробки. Коробки поступали на линию и дальше распределялись по высоте. На первый взгляд задача простая, но для нормальной работы участка нужно было согласовать датчики, конвейеры и поворотный стол. Если хотя бы один элемент отработывал не вовремя, вся последовательность уже нарушалась. Управляющая программа создавалась в CODESYS Control Win V3, а связь с Factory I/O выполнялась через Modbus TCP. Работало это так: виртуальный датчик передавал сигнал в программный контроллер, программа его обрабатывала и возвращала команду обратно в модель. Если алгоритм был написан неправильно, ошибка сразу была видна уже на самом участке. Например, коробка могла уйти не туда, стол не повернуться или конвейер включиться не в тот момент. То есть проблема проявлялась не только в коде, а в поведении всей системы. К входным сигналам относились запуск, остановка, сброс, аварийный останов, а также сигналы датчиков наличия и высоты коробки, положения поворотного стола и состояния выходных участков. По этим сигналам студент понимал, что сейчас происходит с объектом и какое действие должно выполняться дальше.

Выходные сигналы уже управляли подающим и входным конвейерами, загрузкой, поворотом и разгрузкой стола, левым и правым выходными конвейерами. Получалась понятная цепочка: пришел сигнал – программа проверила условие – контроллер выдал команду – механизм сработал. Причем проверяли не только обычную сортировку. Специально задавались и проблемные ситуации: занят поворотный стол, занят выходной конвейер, сработал аварийный останов, отказал датчик или коробка заблокировалась. Здесь уже студенту нужно было не просто запустить программу, а понять, почему система остановилась и что именно нужно исправить. После обнаружения причины студент исправлял программу и повторно проверял работу участка.

Условия реализации и масштабирования методики. Для работы нужны компьютерные рабочие места с Factory I/O и CODESYS Control Win V3 и возможностью настроить обмен по Modbus TCP. Подключение к ре-

альной промышленной установке не требуется. Учебная работа строится последовательно: анализ задачи, определение сигналов, алгоритм, программирование, запуск модели, диагностика ошибок и защита проекта. Занятия может проводить преподаватель профильной дисциплины; итоговые проекты в исследовании дополнительно оценивали три преподавателя профильной кафедры. При переносе методики можно использовать другой виртуальный технологический объект, если сохраняется указанная последовательность работы.

Оценку результатов проводили до начала формирующего этапа и после его завершения с использованием параллельных вариантов заданий. Содержание вопросов отличалось, но структура, сложность и количество баллов оставались одинаковыми.

Диагностика включала четыре части по 25 баллов: теоретическую, алгоритмическую, программно-практическую и проектно-диагностическую. Максимальный итоговый результат составлял 100 баллов.

Теоретический блок включал вопросы по устройству программируемого логического контроллера, входным и выходным сигналам, последовательным алгоритмам, Modbus TCP и требованиям безопасности.

В алгоритмическом блоке студент получал уже конкретную практическую ситуацию. Нужно было определить необходимые сигналы, составить карту входов и выходов и понять, в какой последовательности система должна переходить из одного состояния в другое. Отдельно задавалась неисправность датчика. Здесь важно было не просто написать, что система остановится, а объяснить, почему выбрана именно такая реакция.

Программно-практический блок был связан уже непосредственно с CODESYS. Студенты разрабатывали фрагмент управляющей программы, запускали его, смотрели сигналы и проводили отладку. Если алгоритм работал неправильно, нужно было найти причину. Причем сама компиляция программы еще ничего не означала. Код мог собраться без ошибок, а виртуальный участок при этом работать неправильно, и это тоже учитывалось.

Проектно-диагностическая часть уже объединяла все предыдущие действия. Студент разрабатывал управление сортировочным участком целиком: определял высоту коробки, выбирал нужный конвейер, управлял поворотным столом, задавал аварийный останов и последующий сброс. То есть здесь уже было видно, может ли он связать отдельные команды и состояния в одну нормально работающую систему.

Во время индивидуальной защиты студент объяснял свой алгоритм по шагам. Потом преподаватель специально задавал неисправность или показывал неправильную работу участка. Нужно было найти причину и предложить исправление. Получалось, что проверяли не только сам факт работающей программы, но и понимает ли студент вообще, почему она работает именно так.

Каждый из четырех критериев оценивался максимум в 25 баллов. 0–6 баллов – задание фактически не выполнено или имеются критические ошибки. 7–12 – выполнено частично. 13–18 – основная логика правильная, но остаются ошибки или слабое объяснение. 19–25 баллов – решение полное, и студент может его самостоятельно обосновать.

Критическими считались уже не мелкие ошибки в записи программы, а те, которые влияли на безопасную работу системы. Например, нет приоритета аварийного останова, одновременно включаются несовместимые механизмы или после аварии система сама запускается повторно. Ошибка в имени переменной сама по себе к таким не относилась.

По общей сумме результаты делили на три уровня: 0–49 баллов – низкий, 50–74 – средний, 75–100 – высокий.

Тест проверяли по ключу, а практические задания – уже по заранее установленной рубрике. Итоговые проекты смотрели три преподавателя профильной кафедры независимо друг от друга. Потом их оценки усреднялись, чтобы итог не зависел только от мнения одного эксперта.

Статистику проводили в несколько этапов. Сначала проверяли сами данные: нормальность распределения – критерием Шапиро – Уилка и по Q-Q графикам, однородность дисперсий – критерием Левена. Затем сравнивали группы. Итоговый балл между контрольной и экспериментальной группами оценивали t-критерием Уэлча и дополнительно U-критерием Манна – Уитни. Динамику внутри каждой группы до и после обучения – парным t-критерием.

Для четырех отдельных сравнений по критериям использовали поправку Холма – Бонферрони, чтобы не получить значимость просто из-за большого количества проверок. Значимым считали $p < 0,05$. Дополнительно рассчитывали d Коэна, то есть смотрели не только на то, есть ли статистическая разница, но и на то, насколько она вообще большая.

Этические аспекты исследования

Участие студентов именно в исследовательской части было добровольным. До на-

чала сбора данных им объясняли цель работы и получали информированное согласие. Если студент не хотел, чтобы его результаты использовались в исследовании, это никак не влияло на оценку по дисциплине.

При этом сами лабораторные работы и защита проекта входили в обычную программу и выполнялись всеми студентами. То есть добровольность касалась именно использования данных для исследования, а не учебных заданий. Критерии оценивания для всех оставались одинаковыми.

Перед статистической обработкой фамилии и имена заменялись кодами. В статье приводятся только общие результаты по группам, без раскрытия персональных данных.

Результаты исследования и их обсуждение

Сначала посмотрели входные результаты. Контрольная и экспериментальная группы на старте были практически одинаковыми по уровню подготовки. Распределение студентов по низкому, среднему и высокому уровням статистически значимо не различалось: $\chi^2 = 0,10$; $df = 2$; $p = 0,951$. То есть явного преимущества у одной из групп изначально не было. После завершения экспериментального модуля результаты выросли в обеих группах, но уже по-разному. В контрольной группе средний балл увеличился с $42,8 \pm 11,2$ до $57,8 \pm 10,5$. Плюс 15 баллов. В экспериментальной было $43,7 \pm 11,1$, а стало $74,7 \pm 9,2$ балла. Здесь уже плюс 31 балл – практически в два раза больший прирост. Наиболее заметная разница получилась по алгоритмическому, программно-практическому и проектно-диагностическому критериям, то есть там, где студенту нужно было не просто ответить на вопрос, а самому составить алгоритм, написать программу, запустить систему и разобраться с ошибкой. Эти результаты представлены в табл. 1.

Полученные результаты показывают, что рост был в обеих группах, но в экспериментальной он получился заметнее, особенно там, где студенту нужно было самому составить алгоритм, написать программу, запустить участок и потом еще разобраться, почему система сработала неправильно.

Работа с целой моделью здесь сыграла важную роль. Студент сразу видел связь между программой, сигналами контроллера и тем, что происходит с виртуальным оборудованием. Если в алгоритме допущена ошибка, она проявлялась уже в работе участка. То есть приходилось не только искать проблему в коде, но и смотреть всю логику системы целиком.

Изменилось и распределение студентов по уровням подготовки – здесь разница между группами тоже стала заметной. В экспериментальной группе доля студентов с высоким уровнем выросла с 13,0 до 52,2 %, а с низким уровнем – снизилась с 39,1 до 8,7 %. В контрольной группе высокий уровень изменился с 13,0 до 17,4 %.

Эта динамика показывает, что изменение затронуло не только средний балл. В экспериментальной группе стало заметно меньше студентов, которые не справлялись с основными алгоритмическими и программными действиями, и одновременно увеличилась доля тех, кто выполнял задачу в основном самостоятельно.

При этом связывать полученный результат только с Factory I/O или только с моделированием программируемого логического контроллера было бы неправильно. В экспериментальной группе менялся весь способ организации работы: проект, программирование, самостоятельная отладка, поиск ошибок и защита решения. Поэтому эффект относится к комплексной проектно-лабораторной методике.

Для проверки устойчивости различий отдельно сравнивали итоговые результаты контрольной и экспериментальной групп и динамику внутри каждой группы. Результаты представлены в табл. 2.

Таблица 1

Динамика критериальных результатов студентов, $M \pm SD$, баллы

Показатель	КГ до	КГ после	ЭГ до	ЭГ после
Теоретический критерий	$11,4 \pm 3,2$	$14,8 \pm 3,1$	$11,6 \pm 3,1$	$18,2 \pm 2,8$
Алгоритмический критерий	$11,0 \pm 3,6$	$15,0 \pm 3,5$	$11,2 \pm 3,7$	$19,1 \pm 3,1$
Программно-практический критерий	$10,6 \pm 3,8$	$14,6 \pm 3,4$	$10,9 \pm 3,7$	$19,4 \pm 3,0$
Проектно-диагностический критерий	$9,8 \pm 3,5$	$13,4 \pm 3,6$	$10,0 \pm 3,6$	$18,0 \pm 3,2$
Итоговый балл	$42,8 \pm 11,2$	$57,8 \pm 10,5$	$43,7 \pm 11,1$	$74,7 \pm 9,2$

Примечание: составлена автором на основе полученных данных в ходе исследования.

Таблица 2

Статистическая проверка результатов педагогического эксперимента

Показатель	Метод	Значение	Интерпретация
Исходное распределение уровней	χ^2 Пирсона	$\chi^2 = 0,10$; $df = 2$; $p = 0,951$	Группы сопоставимы до эксперимента
Итоговое распределение уровней	χ^2 Пирсона	$\chi^2 = 6,73$; $df = 2$; $p = 0,035$; $V = 0,38$	Средний эффект в пользу ЭГ
Итоговый балл ЭГ–КГ	t-критерий Уэлча	$t = 5,81$; $df \approx 43$; $p < 0,001$; 95 % ДИ [11,0; 22,8]	Статистически значимое преимущество ЭГ
Непараметрическая проверка	U-критерий Манна – Уитни	$U = 103,0$; $p < 0,001$	Результат устойчив к типу критерия
Размер эффекта	d Коэна	$d = 1,71$	Крупный эффект
Динамика внутри КГ	Парный t-критерий	$t(22) = 6,10$; $p < 0,001$; $d = 0,90$	Значимый рост при традиционной подготовке
Динамика внутри ЭГ	Парный t-критерий	$t(22) = 12,40$; $p < 0,001$; $d = 1,83$	Более выраженный рост при проектно-лабораторной методике

Примечание: составлена автором на основе полученных данных в ходе исследования.

Применение поправки Холма – Бонферрони к четырем дополнительным сравнениям не изменило выводов исследования: различия между группами по каждому критерию сохранили статистическую значимость (скорректированное $p < 0,001$). Результаты t-критерия Уэлча и U-критерия Манна – Уитни также совпали по направлению вывода.

Размер эффекта составил

$$d \text{ Коэна} = 1,71.$$

Различие получилось крупным, но здесь важно учитывать условия самого исследования. Всего участвовали 46 студентов, а полной рандомизации не было. Поэтому результат выглядит убедительно, но переносить его без дополнительной проверки на другие группы было бы слишком прямолинейно.

При выполнении и защите проектов встречались и типичные ошибки. Например, неправильный приоритет аварийного останова, путаница во входных и выходных сигналах, пропуск состояния ожидания или сброса. Иногда студент находил ошибку в программе, но не мог аргументированно объяснить, почему она вообще появилась.

Отдельно статистически такие ошибки не сравнивали. До и после обучения не велся специальный стандартизированный протокол, где фиксировалось бы точное количество каждой ошибки. Поэтому использовать эти наблюдения как самостоятельное доказательство эффективности методики было бы неправильно. Они рассматривались только как дополнение к основным количественным результатам.

Ограничения исследования. Исследование проводилось в одном вузе, со студентами-третьекурсниками одного направления подготовки. Выборка небольшая – 46 чел. Поэтому полученные результаты нельзя сразу переносить на другие инженерные направления и университеты. Сначала методику нужно проверить на других группах и уже на более широкой выборке.

Еще одно ограничение связано с формированием групп. Полной случайной рандомизации не было: использовались обычные учебные группы, уже сформированные по расписанию. Поэтому полностью исключить влияние особенностей конкретной группы нельзя.

Кроме того, в экспериментальной группе одновременно менялось несколько элементов работы: использовались цифровой двойник, программирование контроллера, проектная работа, самостоятельная отладка, поиск ошибок и защита. Из-за этого полученный эффект относится к методике в целом, а вклад каждого отдельного компонента в данное исследование не разделялся.

Примененный учебный цифровой двойник представлял собой функциональную модель типового технологического участка и не был синхронизирован с действующей физической производственной установкой. Поэтому результаты относятся именно к учебно-моделирующему применению цифрового двойника.

Перспективы исследования. Дальнейшая работа предполагает кросс-вузовскую валидацию методики с участием студентов нескольких образовательных организаций.

Также методику необходимо проверить на других технологических объектах, чтобы оценить ее применимость к разным задачам автоматизации.

Заключение

Результаты проведенного квазиэксперимента показывают потенциальную эффективность комплексной проектно-лабораторной методики, объединяющей учебный цифровой двойник, моделирование программируемого логического контроллера, разработку алгоритма управления, программирование, отладку и защиту проекта. В экспериментальной группе были получены более высокие результаты по алгоритмическому, программно-практическому и проектно-диагностическому критериям. Вместе с тем выявленный эффект относится к методике в целом и не доказывает самостоятельного влияния учебного цифрового двойника.

Практическая ценность работы состоит в том, что предложенная методика может использоваться при преподавании дисциплин, связанных с автоматизацией технологических процессов, программированием контроллеров и цифровым моделированием производственных объектов. При необходимости виртуальный объект можно заменить, сохранив общую последовательность проектно-лабораторной работы.

Список литературы

1. Мартынов В. Г., Шейнбаум В. С. Инженерная педагогика в контексте инженерной деятельности // Высшее образование в России. 2022. Т. 31. № 6. С. 152–168. URL: <https://www.elibrary.ru/item.asp?id=48967422> (дата обращения: 12.08.2026). DOI: 10.31992/0869-3617-2022-31-6-152-168. EDN: PDCQIA.
2. Бобровский А. В., Бажутин М. М., Зотов А. В., Чижаткина Е. Д. Ресурсы, мотивация и проблемные зоны модели проектного обучения в подготовке инженеров // Высшее образование в России. 2025. Т. 34. № 4. С. 144–168. URL: <https://www.elibrary.ru/item.asp?id=82309945> (дата обращения: 12.08.2026). DOI: 10.31992/0869-3617-2025-34-4-144-168. EDN: ZCIGCZ.
3. Вихман В. В., Ромм М. В. «Цифровые двойники» в образовании: перспективы и реальность // Высшее образование в России. 2021. Т. 30. № 2. С. 22–32. URL: <https://www.elibrary.ru/item.asp?id=44744729> (дата обращения: 12.08.2026). DOI: 10.31992/0869-3617-2021-30-2-22-32. EDN: FODZSG.
4. Гурьева Т. Н., Шарабаева Л. Ю., Атаян А. М. О возможности использования цифровых двойников в высшем образовании // Современные проблемы науки и образования. 2023. № 5. URL: <https://science-education.ru/article/view?id=32986> (дата обращения: 12.08.2026). DOI: 10.17513/spno.32986.
5. Грачев В. В., Мышляев Л. П., Макаров Г. В., Свицов М. М., Коровин Д. Е. Использование цифровых двойников при обучении оперативно-диспетчерского персонала автоматизированных промышленных комплексов // Информационно-коммуникационные технологии в педагогическом образовании. 2021. № 3 (72). С. 52–57. URL: <https://elibrary.ru/item.asp?id=44868982> (дата обращения: 30.06.2026). EDN: TTKNZF.
6. Намиот Д. Е., Покусаев О. Н., Куприяновский В. П., Жабицкий М. Г. Цифровые двойники и системы дискретно-событийного моделирования // International Journal of Open Information Technologies. 2021. Т. 9. № 2. С. 70–75. URL: <https://elibrary.ru/item.asp?id=44683449> (дата обращения: 30.06.2026). EDN: VWEGHR.
7. Мельник А. Д., Меренков А. В., Сандлер Д. Г. Проектное обучение в передовых инженерных школах: опережающее образование личности // Интеграция образования. 2025. Т. 29. № 2(119). С. 282–299. URL: <https://edumag.mrsu.ru/index.php/ru/articles/140-25-2/1242-10-15507-1991-9468-029-202502-5> (дата обращения: 12.08.2026). DOI: 10.15507/1991-9468.029.202502.282-299. EDN: FJAZPW.
8. Karanam S. A. K., Hartman N. W. A systematic review of Digital Twin (DT) and virtual learning environments (VLE) for smart manufacturing education // Manufacturing Letters. 2025. Vol. 44. P. 1597–1608. URL: <https://www.sciencedirect.com/science/article/pii/S2213846325002159> (дата обращения: 12.08.2026). DOI: 10.1016/j.mfglet.2025.06.179.
9. Chen J., Kolmos A., Du X. Forms of implementation and challenges of PBL in engineering education: A review of literature // European Journal of Engineering Education. 2021. Vol. 46. Is. 1. P. 90–115. DOI: 10.1080/03043797.2020.1718615.
10. Li J., Liang W. Effectiveness of virtual laboratory in engineering education: A meta-analysis // PLOS ONE. 2024. Vol. 19. Is. 12. Art. e0316269. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0316269> (дата обращения: 12.08.2026). DOI: 10.1371/journal.pone.0316269.
11. Vargas H., Heradio R., Donoso M., Farias G. Teaching automation with Factory I/O under a competency-based curriculum // Multimedia Tools and Applications. 2023. Vol. 82. Is. 13. P. 19221–19246. URL: <https://link.springer.com/article/10.1007/s11042-022-14047-9> (дата обращения: 12.08.2026). DOI: 10.1007/s11042-022-14047-9.
12. Vargas H., Heradio R., Farias G., Lei Z., de la Torre L. A pragmatic framework for assessing learning outcomes in competency-based courses // IEEE Transactions on Education. 2024. Vol. 67. Is. 2. P. 224–233. DOI: 10.1109/TE.2023.3347273.
13. Riera B., Vigário B. HOME I/O and FACTORY I/O: a virtual house and a virtual plant for control education // IFAC-PapersOnLine. 2017. Vol. 50. Is. 1. P. 9144–9149. URL: <https://www.sciencedirect.com/science/article/pii/S2405896317323261> (дата обращения: 12.08.2026). DOI: 10.1016/j.ifacol.2017.08.1719.
14. Kučera E., Haffner O., Drahoš P., Čigánek J. Educational Case Studies for Pilot Engineer 4.0 Programme: Monitoring and Control of Discrete-Event Systems Using OPC UA and Cloud Applications // Applied Sciences. 2022. Vol. 12. Is. 17. Art. 8802. URL: <https://www.mdpi.com/2076-3417/12/17/8802> (дата обращения: 12.08.2026). DOI: 10.3390/app12178802.
15. Domínguez M., González-Herbón R., Rodríguez-Osorio J. R., Fuertes J. J., Prada M. A., Morán A. Development of a Remote Industrial Laboratory for Automatic Control based on Node-RED // IFAC-PapersOnLine. 2020. Vol. 53. Is. 2. P. 17210–17215. URL: <https://www.sciencedirect.com/science/article/pii/S2405896320323478> (дата обращения: 12.08.2026). DOI: 10.1016/j.ifacol.2020.12.1741.

Конфликт интересов: Автор заявляет об отсутствии конфликта интересов.

Conflict of interest: The author declares that there is no conflict of interest.

Финансирование: Автор заявляет об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.