

# Архитектура иерархических цифровых двойников на основе PBA-PINN моделей

Т. В. Лазовская, Г. Ф. Малыгина, Д. А. Тархов

*Санкт-Петербургский политехнический университет Петра Великого*

lazovskaya\_tv@spbstu.ru

**Аннотация.** Рассматривается задача построения иерархических систем цифровых двойников для интеллектуального управления сложными организационно-техническими объектами. В классической постановке нижние уровни такой иерархии могут описываться локальными математическими моделями, например системами дифференциальных уравнений, однако их многократное численное решение и последующая передача результатов на верхние уровни управления приводят к высокой вычислительной сложности и затрудняют оперативную адаптацию системы. Предлагается архитектура, в которой на нижних уровнях вместо прямого использования полных моделей формируются PBA-PINN модели, то есть физически-информированные нейросетевые модели с архитектурой, построенной на основе численной или аналитико-численной структуры локальной задачи, включающие обучаемые блоки, уточняемые по данным наблюдений и управляющим сигналам верхних уровней. Нейросетевые агенты нижних уровней передают наверх не полное состояние объекта и не результат многократного решения исходных уравнений, а компактные адаптивные модели или агрегированные зависимости между ресурсами, сценариями, ограничениями и целевыми показателями. Работа носит концептуально-методологический характер и формулирует архитектуру, математическую постановку и направления дальнейшей валидации предложенного подхода.

**Ключевые слова:** цифровой двойник; PBA-PINN; нейросетевая архитектура на основе численной схемы; иерархическое управление; low-fidelity модель; нейросетевой агент

## I. ВВЕДЕНИЕ

Развитие интеллектуальных производственных и организационно-технических систем связано с необходимостью построения моделей, которые не только описывают текущее состояние объекта, но и позволяют прогнозировать его поведение, оценивать последствия управленческих воздействий и адаптироваться к новым данным. Одним из распространённых подходов к решению таких задач является построение цифровых двойников, связывающих физический или организационный объект, данные наблюдений, математическую модель и вычислительные алгоритмы [1].

Для сложных систем недостаточно рассматривать цифровой двойник как единую монолитную модель. Такие системы обычно имеют иерархическую структуру [2]: отдельные технические объекты, подразделения, производственные процессы и уровни управления обладают собственными целями, ограничениями и информационными потоками. Управление в такой системе становится многокритериальным и распределённым: решения на нижних уровнях влияют на

показатели верхних уровней, а цели верхних уровней задают ограничения и приоритеты для нижних.

Классическая иерархическая схема может предполагать, что на каждом нижнем уровне существует локальная модель объекта, например система дифференциальных уравнений, которая используется для оценки поведения подсистемы при различных управляющих воздействиях. Однако в сложных системах такой подход сталкивается с несколькими трудностями. Во-первых, полное описание динамики всех подсистем может быть недоступно или чрезмерно вычислительно затратно. Во-вторых, верхнему уровню обычно не требуется вся внутренняя динамика нижнего уровня; ему важнее получить управленчески значимую зависимость между ресурсами, сценариями, рисками и достижимыми показателями. В-третьих, условия функционирования объекта могут изменяться, что требует регулярного уточнения моделей. В-четвёртых, при многокритериальной постановке часто требуется не одна оптимальная точка, а множество компромиссных решений.

В этих условиях перспективным является использование PBA-PINN (Physics-informed neural networks with Physics-based architecture, физически-информированные нейронные сети с архитектурой, основанной на физике) моделей, в которых физико-математическая информация учитывается не только в функции потерь, но и в самой архитектуре нейросетевой модели [3,4]. В отличие от чисто эмпирических моделей типа «чёрного ящика», такие модели наследуют часть структуры исходного описания объекта: численной схемы, low-fidelity модели, приближённого дифференциального уравнения или аналитического преобразования. Обучаемая нейросетевая часть встраивается в эту структуру как корректирующий, замыкающий или параметризующий блок.

Цель настоящей работы — предложить архитектуру иерархической системы цифровых двойников, в которой такие PBA-PINN модели используются прежде всего на нижних уровнях иерархии. Их задача состоит не только в аппроксимации локальной динамики, но и в формировании компактной модели, которую нейросетевой агент может передавать на вышестоящий уровень. Верхний уровень, в свою очередь, задаёт цели, ограничения и новые данные, по которым локальные модели дообучаются и адаптируются. Таким образом, основной акцент работы переносится с прямого численного решения локальных уравнений на построение адаптивных нейросетевых представителей нижних уровней управления.

Основной вклад работы состоит в следующем: предложена архитектура иерархических цифровых двойников, в которой нижние уровни передают наверх не только данные или результаты расчётов, но и компактные PBA-PINN-модели; описана роль нейросетевых агентов как механизмов межуровневой передачи таких моделей; сформулирован принцип адаптации моделей нижнего уровня под воздействием данных наблюдений и управляющих сигналов верхних уровней.

## II. ИЕРАРХИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Рассмотрим сложную систему, представленную в виде набора взаимосвязанных уровней управления. На каждом уровне выделяется объект или подсистема, для которой строится локальный цифровой двойник. Пусть для блока с номером  $k$  состояние обозначается через  $x_k(t)$ , управляющее воздействие – через  $u_k(t)$ , измеряемый выход – через  $y_k(t)$ . Динамика блока зависит не только от локального состояния и управления, но и от информации, поступающей с вышестоящего и нижестоящих уровней [5]. В общем виде это можно записать как.

$$\frac{dx_k}{dt} = f_k(x_k, u_k, z_k^{up}, z_k^{down}, p_k, t),$$

$$y_k = g_k(x_k, u_k, p_k, t),$$

где  $z_k^{up}$  описывает управляющие или целевые ограничения, поступающие с верхнего уровня,  $z_k^{down}$  – агрегированную информацию от нижних уровней,  $p_k$  – параметры модели и внешних условий.

Задача управления состоит в выборе локальных управляющих воздействий  $u_k(t)$  и передаваемых вниз целевых ограничений таким образом, чтобы обеспечить достижение набора целевых показателей. В общем случае эти показатели образуют векторный функционал

$$J_k = (J_{k,1}, J_{k,2}, \dots, J_{k,m}),$$

компоненты которого могут описывать экономическую эффективность, риск, устойчивость, качество продукции, расход ресурсов, сроки выполнения операций и другие характеристики. Поэтому задача естественным образом формулируется как многокритериальная оптимизация. В результате вместо единственного оптимального решения формируется множество Парето-оптимальных вариантов, выбор конкретной точки на котором может осуществляться лицом, принимающим решения, либо интеллектуальным модулем поддержки выбора.

Для практической реализации такой постановки требуется не только решить оптимизационную задачу, но и обеспечить регулярное уточнение функций  $f_k$  и  $g_k$  по данным наблюдений. Если на каждом нижнем уровне сохранять полную модель в виде системы дифференциальных уравнений, то верхнему уровню приходится многократно запрашивать результаты численного моделирования при различных сценариях управления. Это затрудняет параллельную работу

уровней и делает систему чувствительной к изменению данных и целей. Поэтому далее предлагается заменить прямую передачу результатов решения локальных уравнений передачей обучаемых PBA-PINN моделей, которые выступают в роли компактных представителей нижних уровней.

## III. PBA-PINN МОДЕЛИ НИЖНИХ УРОВНЕЙ

Ключевая идея предлагаемой архитектуры состоит в том, что на нижних уровнях иерархии цифровых двойников используются не только исходные математические модели объектов, но и их обучаемые нейросетевые представители. Эти представители строятся не как произвольные полносвязные сети, а как PBA-PINN модели. Под PBA-PINN в данной работе понимается физически-информированная нейросетевая модель, в которой априорная физико-математическая информация используется не только в функции потерь, но и при построении архитектуры. Форма вычислений в такой модели задаётся логикой локальной задачи: численной схемой, аналитико-численным преобразованием, приближённым уравнением или другим low-fidelity описанием объекта.

Такой подход связан с изменением роли модели в интеллектуальной системе. Если автономная цифровая модель предназначена главным образом для отдельного расчёта, то цифровой двойник должен участвовать в прогнозе, анализе сценариев и поддержке решений. Поэтому нижний уровень иерархии должен передавать наверх не только отдельные результаты расчёта, но и компактное представление своего поведения в области сценариев, существенных для текущей задачи управления.

Для таких моделей важны четыре требования: вычислительная эффективность, компактность, интерпретируемость, адаптация к данным и физико-математическая согласованность. Обычный data-driven суррогат может быть записан как

$$x_{k,n+1} = N_k(x_{k,n}, u_{k,n}, z_k^{up}, p_k, h; \theta_k),$$

где  $x_{k,n}$  – состояние нижнего уровня на временном слое  $n$ ,  $u_{k,n}$  – локальное управление,  $z_k^{up}$  – цели, ограничения или сценарные сигналы верхнего уровня,  $p_k$  – параметры объекта и внешних условий,  $h$  – шаг по времени,  $\theta_k$  обозначает набор обучаемых параметров нейросетевой модели (веса и смещения слоев). Такая модель может быть эффективной как аппроксиматор, но её архитектура обычно выбирается эмпирически и не фиксирует явно структуру исходной математической постановки.

Классические физически-информированные нейронные сети [6, 7] частично решают эту проблему, включая уравнения и условия задачи в функцию потерь. Однако в базовой постановке PINN физико-математическая информация в первую очередь используется как ограничение при обучении модели, а не как принцип построения её вычислительной структуры. В предлагаемом подходе часть структуры

переносится непосредственно в архитектуру модели. Иными словами, численная схема рассматривается не только как алгоритм расчёта, но и как шаблон вычислительной структуры.

Пусть локальная динамика нижнего уровня в обозначениях раздела II задаётся параметризованной задачей Коши

$$\frac{dx_k}{dt} = f_k(x_k(t), u_k(t), z_k^{up}, p_k, t), x_k(t_0) = x_{k,0}, t \in [t_0, T].$$

Обычный численный метод строит приближённые значения решения в заранее заданных узлах сетки. В этом случае результатом расчёта является набор значений  $x_{k,0}, x_{k,1}, \dots, x_{k,n}$ , зависящий от выбранного интервала интегрирования и шага. Такой результат можно использовать для одного сценария, но он ещё не является компактной моделью, которую удобно многократно передавать и дообучать внутри иерархии цифровых двойников.

В предлагаемом подходе используется аналитическая модификация численной схемы. Её смысл состоит в том, что конечный момент времени рассматривается как аргумент модели, а не как заранее фиксированная граница расчёта. Для заданного числа шагов  $n$  вводятся зависящие от конечного момента  $t$  узлы

$$h(t) = \frac{t-t_0}{n}, t_i(t) = t_0 + ih(t), i = 0, \dots, n.$$

Тогда каждый шаг схемы также становится функцией конечного момента  $t$ :

$$x_{k,i+1} = \Phi_{k,i}(f_k, x_{k,i}(t), u_k(t_i(t)), z_k^{up}, p_k, h(t), \alpha_{k,i}), i = 0, \dots, n-1.$$

где  $\Phi_{k,i}$  – оператор шага выбранной численной схемы,  $\alpha_{k,i}$  – параметры схемы или локальные величины, определяющие положение точки аппроксимации, коэффициенты смешивания, начальное приближение или другие внутренние элементы метода.

После композиции шагов получается не просто численное значение в последнем узле, а аналитико-численное представление приближённого решения как функции аргументов модели:

$$\tilde{x}_k(t, u_k, z_k^{up}, p_k) = x_{k,n}(t) = \Phi_{k,n-1} \circ \dots \circ \Phi_{k,0}(x_{k,0}).$$

Именно это представление является основой нейросетевой архитектуры нижнего уровня. Оно сохраняет вычислительную логику численного метода, но уже может рассматриваться как параметризованная функция, принимающая на вход конечный момент времени, управляющие воздействия, параметры объекта и сигналы верхнего уровня. Поэтому нижний уровень получает не только результат отдельного расчёта, а компактную аналитико-численную модель, пригодную для многократного вызова, дообучения и передачи на вышестоящий уровень.

Сама PBA-PINN модель возникает, когда отдельные параметры или локальные элементы схемы становятся

обучаемыми. В простейшем случае фиксированный параметр схемы заменяется оптимизируемой величиной, в более общем случае параметр или локальный элемент схемы задаётся компактной нейросетевой функцией

$$\alpha_{k,n+1} = N_{k,n}(x_{k,n}, u_{k,n}, z_k^{up}, p_k, h_n; \theta_{k,n}).$$

Тогда шаг схемы принимает вид

$$x_{k,n+1} = \Phi_{k,n}(x_{k,n}, u_{k,n}, z_k^{up}, p_k, h_n, N_{k,n}(x_{k,n}, u_{k,n}, z_k^{up}, p_k, h_n; \theta_{k,n})).$$

Здесь нейросеть не добавляется к готовому решению как внешняя поправка. Она определяет внутренний элемент вычислительной схемы: параметр, вес, точку аппроксимации, локальное замыкающее соотношение или другой компонент, через который строится следующий шаг. Поэтому сохраняется общая аналитико-численная структура модели, а обучение действует локально внутри этой структуры.

Такой нижний уровень включает четыре компоненты: исходное локальное описание объекта или процесса; PBA-PINN модель, построенную на его основе; модуль адаптации по данным наблюдений и сигналам верхнего уровня; нейросетевого агента, передающего на верхний уровень компактное отображение между ресурсными или сценарными параметрами нижнего уровня и достижимыми значениями целевых показателей.

Таким образом, нижний уровень перестаёт быть только источником численных решений исходной модели. Он становится активным агентом, который строит, адаптирует и передаёт наверх собственную PBA-PINN модель. Именно это отличает предлагаемую архитектуру от схем, где нейросети используются только как внешние суррогаты после завершения численного моделирования [8].

#### IV. ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ УРОВНЕЙ

Взаимодействие уровней иерархии строится как обмен не полными внутренними состояниями, а обучаемыми моделями, агрегированными зависимостями и ограничениями. Нижестоящий уровень на основе собственного цифрового двойника и PBA-PINN модели формирует компактное представление своего поведения для верхнего уровня. Это представление может иметь вид зависимости между выделяемыми ресурсами, управляющими сценариями, ожидаемыми результатами, рисками и достижимыми значениями целевых показателей.

В такой схеме нейросетевой агент нижнего уровня выполняет две функции. Первая функция — аппроксимационная: агент строит модель отклика подсистемы на управляющие воздействия и ограничения. Вторая функция — коммуникационная: агент передаёт эту модель или её агрегированные характеристики на верхний уровень, где они используются для распределения ресурсов, уточнения целей и выбора компромиссных решений.

Верхний уровень не обязан каждый раз запускать полное численное моделирование нижней подсистемы. Вместо этого он работает с переданной нейросетевой моделью как с адаптивным представителем нижнего уровня. При изменении целей или появлении новых

данных верхний уровень может возвращать вниз обновлённые ограничения, сценарии или области интереса, после чего локальная модель дообучается. Поэтому взаимодействие уровней имеет итеративный характер: нижний уровень передаёт вверх модель достижимого поведения, верхний уровень уточняет требования, нижний уровень адаптирует модель в соответствии с новыми условиями.

Такой подход позволяет уменьшить объём передаваемой информации и повысить вычислительную масштабируемость системы. Каждый локальный цифровой двойник может обучаться и адаптироваться относительно независимо, а обмен между уровнями сводится к передаче параметров, прогнозов, ограничений или компактных суррогатных моделей. Это создаёт предпосылки для параллельной и распределённой реализации алгоритмов.

Возможны несколько режимов адаптации. В режиме параметрической адаптации изменяются только веса обучаемых блоков внутри РВА-PINN модели. В режиме структурной адаптации может изменяться сама форма нейросетевой вставки, набор входных переменных, число локальных моделей или способ агрегации информации. В режиме сценарной адаптации система выбирает те локальные модели, которые лучше описывают поведение объекта в текущей области управляющих воздействий.

Отдельное значение имеет согласование локальных и глобальных критериев. Оптимальное решение на одном уровне не обязательно является оптимальным для всей иерархии. Поэтому архитектура должна поддерживать итеративный обмен информацией: верхний уровень задаёт ограничения и приоритеты, нижний уровень оценивает достижимые области решений с помощью своей обучаемой модели, после чего верхний уровень уточняет распределение ресурсов и целевые показатели. Нейросетевые агенты в этой схеме выполняют роль адаптивных моделей межуровневого взаимодействия.

## V. ОБСУЖДЕНИЕ

Предложенная архитектура не является завершённым алгоритмом для конкретной прикладной задачи, а задаёт общий способ включения РВА-PINN моделей в иерархические системы цифровых двойников. Её ключевое отличие состоит в том, что нижние уровни иерархии рассматриваются не только как источники данных или численных решений, но и как источники обучаемых моделей. Эти модели наследуют структуру локального математического описания и одновременно адаптируются по данным и сигналам верхних уровней управления.

Такой подход может быть полезен для систем, где прямое моделирование всех уровней в единой модели затруднено, а принятие решений требует учёта данных, физических или технологических ограничений и экспертных предпочтений. К таким системам можно отнести производственные комплексы, распределённые технические объекты, системы управления проектами и процессы ресурсного планирования.

Дальнейшая валидация подхода требует построения модельных задач, в которых можно сравнить несколько

вариантов архитектуры: прямое использование полных локальных моделей, набор независимых нейросетевых суррогатов и предложенную иерархическую схему со РВА-PINN моделями и нейросетевыми агентами. В качестве критериев сравнения могут использоваться точность прогноза, устойчивость к изменению условий, вычислительные затраты, качество найденных компромиссных решений и скорость адаптации по новым данным.

## VI. ЗАКЛЮЧЕНИЕ

В работе предложена архитектура иерархических цифровых двойников на основе РВА-PINN моделей. Рассмотрена постановка задачи, в которой нижние уровни иерархии не только описываются локальными математическими моделями, но и формируют их обучаемые нейросетевые представления для передачи на верхние уровни управления. В отличие от прямого использования систем дифференциальных уравнений на каждом уровне, предлагается использовать компактные модели, наследующие структуру численной или приближённой математической схемы и адаптируемые по данным, изменениям внешних условий и управляющим сигналам сверху.

Работа носит концептуально-методологический характер. Основными направлениями дальнейших исследований являются разработка конкретных алгоритмов обучения межуровневых нейросетевых агентов, построение тестовых задач для валидации архитектуры и исследование устойчивости таких систем при изменении данных, целей и структуры управления.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Grieves M., Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems // *Transdisciplinary Perspectives on Complex Systems* (ed. by F.-J. Kahlen, S. Flumerfelt, A. Alves). Cham : Springer, 2017. P. 85–113. DOI: 10.1007/978-3-319-38756-7\_4.
- [2] Böttjer T., Tola D., Kakavandi F., Wewer C. R., Ramantujan D., Gomes C., Larsen P. G., Iosifidis A. “A review of unit level digital twin applications in the manufacturing industry,” // *CIRP Journal of Manufacturing Science and Technology*. 2023. Vol. 45. P. 162–189. DOI: 10.1016/j.cirpj.2023.06.011.
- [3] Tarkhov D. A., Lazovskaya T. V., Malykhina G. F. “Constructing Physics-Informed Neural Networks with Architecture Based on Analytical Modification of Numerical Methods by Solving the Problem of Modelling Processes in a Chemical Reactor,” // *Sensors*. 2023. Vol. 23, No. 2. Article 663. DOI: 10.3390/s23020663.
- [4] Lazovskaya T., Tarkhov D. “Numerics as Neural Networks: A Compact Low-Fidelity Layer for Multi-fidelity Modelling,” // *Advances in Neural Computation, Machine Learning, and Cognitive Research IX*. 2026. P. 381–391. DOI:10.1007/978-3-032-07690-8\_31
- [5] W. Findelsen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak, *Control and Coordination in Hierarchical Systems*. Chichester, U.K.: Wiley, 1980.
- [6] Lagaris I. E., Likas A., Fotiadis D. I. “Artificial Neural Networks for Solving Ordinary and Partial Differential Equations,” // *IEEE Transactions on Neural Networks*. 1998. Vol. 9, No. 5. P. 987–1000.
- [7] Raissi M., Perdikaris P., Karniadakis G. E. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” // *Journal of Computational Physics*. 2019. Vol. 378. P. 686–707. DOI: 10.1016/j.jcp.2018.10.045.
- [8] Herrmann L., Kollmannsberger S. “Deep learning in computational mechanics: a review,” // *Computational Mechanics*. 2024. Vol. 74. P. 281–331. DOI: 10.1007/s00466-023-02434-4