

МАШИНОСТРОЕНИЕ И МАШИНОВЕДЕНИЕ

Научная статья
УДК 004.272.3
DOI: 10.34759/vst-2021-4-245-256

ПАМЯТЬ-ЦЕНТРИЧЕСКИЕ МОДЕЛИ СИСТЕМ УПРАВЛЕНИЯ ДВИЖЕНИЕМ ПРОМЫШЛЕННЫХ РОБОТОВ

Александр Александрович Зеленский¹ ✉, Юрий Владимирович Илюхин², Андрей Армович Грибков³

^{1,2,3}Московский государственный технологический университет “СТАНКИН”; Москва, Россия

¹zelenskyaa@gmail.com ✉

²y.ilyukhin@stankin.ru

³a.gribkov@stankin.ru

Аннотация. Рассказывается о значении систем управления движением в режиме реального времени для обеспечения глобальной конкурентоспособности и технологической безопасности страны в условиях реализации четвертой промышленной революции. Рассмотрены актуальные задачи повышения производительности вычислений, которые могут быть решены за счет совершенствования архитектуры вычислительных машин: уменьшения объема потока обрабатываемых данных, увеличения скорости передачи данных между элементами вычислительной машины, устранения очередей при одновременном обращении к одной памяти нескольких вычислительных устройств. На основе анализа возможных путей решения поставленных задач предлагается концептуальная модель системы управления движением промышленным роботом. В системе, построенной согласно предлагаемой концептуальной модели, задача уменьшения объема обрабатываемого потока данных решается за счет использования дополнительных вычислительных модулей — сопроцессоров и ускорителей, выполняющих параллельные вычисления. При этом основная часть вычислений осуществляется без управления из ядра системы. Для описания работы системы управления движением промышленного робота, построенной в соответствии с данной концептуальной моделью, выбирается акторная инструментальная модель, обеспечивающая эмуляцию параллельных вычислений и взаимодействия функциональных модулей.

Ключевые слова: промышленные роботы, система управления промышленными роботами, мехатронные системы контурной обработки, станки с ЧПУ, память-центрическая архитектура управления, высокопроизводительные вычисления в памяти, концептуальная модель систем управления движением, акторная модель управления

Финансирование: Исследование выполнено за счет гранта Российского научного фонда № 21-79-10392, <https://rscf.ru/project/21-79-10392/>

Для цитирования: Зеленский А.А., Илюхин Ю.В., Грибков А.А. Память-центрические модели систем управления движением промышленных роботов // Вестник Московского авиационного института. 2021. Т. 28. № 4. С. 245-256. DOI: 10.34759/vst-2021-4-245-256

MACHINE-BUILDING ENGINEERING AND MACHINE SCIENCE

Original article

MEMORY-CENTRIC MODELS OF INDUSTRIAL ROBOTS CONTROL SYSTEMS

Aleksandr A. Zelenskii¹ ✉, Yurii V. Ilyukhin², Andrei A. Gribkov³

^{1,2,3}Moscow State University of Technology “STANKIN”, Moscow, Russia

¹zelenskyaa@gmail.com ✉

²y.ilyukhin@stankin.ru

³a.gribkov@stankin.ru

Abstract

The article recounts the significance of real-time traffic control systems for global competitiveness and technological security ensuring amidst the fourth industrial revolution realization. As far as the growth potential of computers elements base running speed is close to exhaustion, and further development in this trend is being associated with significant technical complexions and economic efficiency reduction, computers architecture improvement should be considered as the main trend of the computer productivity increasing. The article considered pressing tasks of the computations productivity increasing, which may be solved at the cost of computers architecture improvement. These tasks include the processed data flow volume reduction; increasing data transmission speed between computer elements; eliminating queues while several computing devices simultaneously accessing the same memory. The authors propose conceptual model of the industrial robot movement control based on the analysis of the possible ways of the set problems solving. The problem of the processed data flow reduction is being solved in the system built according to the conceptual model being proposed by application of extra computing modules, such as coprocessors and accelerators, performing parallel computing. The main portion of computations herewith is being performed without control from the system core. The problem of data transmission speed increasing between the system functional elements and blocks is being solved by the memory-centric architecture employing, with which all devices requiring high speed of data exchange with memory for their operation, are being integrated into the memory. The queues elimination problem is being solved by dynamic random access memory (DRAM) splitting into local areas accessible only by a single device. Interaction between devices is being implemented in the high-speed static random access memory (SRAM) employing minimum data volumes, as well as through the communication network ensuring direct communication between the devices without delays occurrence. The actor instrumental model, ensuring emulation of parallel computing and functional modules interaction, is being selected to describe the industrial robot movement control system operation built according to the presented conceptual model.

Keywords: industrial robots, industrial robots control system, contouring machining mechatronic systems, CNC machines, memory-centric control architecture, high-performance computing in memory, conceptual model of motion control systems, actor control model

Funding: this work is supported by the Russian Science Foundation under grant No 21-79-10392, <https://rscf.ru/project/21-79-10392/>

For citation: Zelenskii A.A., Ilyukhin Y.V., Gribkov A.A. Memory-centric models of industrial robots control systems. *Aerospace MAI Journal*, 2021, vol. 28, no. 4, pp. 245-256. DOI: 10.34759/vst-2021-4-245-256

Введение

Базовым трендом мирового технологического развития в условиях реализации четвертой промышленной революции стала цифровая трансформация экономики, ключевыми инструментами которой являются промышленные роботы, станки и другое оборудование с системами числового программного управления [1]. В автомобилестроении, авиационно-космическом машиностроении, станкостроении и других отраслях машиностроительного комплекса использование оборудования с системами ЧПУ является определяющим фактором глобальной конкурентоспособности страны, обороноспособности и технологической независимости [2, 3].

Системы управления движением (ЧПУ и более сложные) составляют существенную часть стоимости оборудования (от 25 до 50%) и являются технологически наиболее сложной его составляющей. В настоящее время глобальная конкурентоспособность страны в значительной степени определяется наличием у нее собственного научно-технического потенциала в области разработки и производства систем управления движением роботов, станков и другого оборудования. Это обусловлено определяющей ролью производства средств производства в обеспечении экономического развития страны. Технологическое отставание в области наиболее значимых средств производства (промышленные роботы и технологическое оборудование с ЧПУ), зависимость от импорта (в России в 2020 г., согласно данным ФТС и Росстата, 91% в сегменте металлообрабатывающего оборудования с ЧПУ и 94% — в сегменте роботов) это прямая угроза технологической безопасности страны.

Характерной чертой формируемого в ходе четвертой промышленной революции производства, декларированной в рамках программы Индустрия 4.0, является разработка и использование технических систем, управляемых в режиме реального времени [4]. Как уже было отмечено, для цифровой трансформации экономики наиболее значимыми являются такие технические системы как промышленные роботы и станки с ЧПУ.

Таким образом, одной из наиболее актуальных задач научно-технического развития в мире, и России в частности, является создание (и последующее производство) систем управления движением в режиме реального времени для промышленных роботов, станков с ЧПУ и другого

технологического оборудования. На базе имеющихся технических решений в области вычислительных машин сформировать систему управления движением в режиме реального времени с функциональными возможностями, соответствующими требованиям цифрового производства, пока не представляется возможным. Существующие системы управления в режиме реального времени имеют крайне ограниченные функциональные возможности и низкую скорость реакции вследствие недостаточной производительности вычислений.

В течение длительного периода времени с начала развития технологий ЭВМ рост производительности вычислений базировался на развитии быстродействия элементной базы. В настоящее время предел быстродействия электронных компонентов вычислительных машин практически достигнут [5]. В среднесрочной перспективе (15—25 лет) дальнейшее увеличение быстродействия элементной базы будет сопровождаться огромными расходами, неоправданными для решения подавляющего большинства задач, в том числе задач управления движением.

Экономически оправданным и технологически осуществимым является решение задачи повышения производительности вычислительных машин за счет совершенствования их архитектуры.

Сформулируем основные задачи повышения производительности, имеющие решение за счет совершенствования архитектуры вычислительных машин.

Задача 1. Необходимо уменьшить объем обрабатываемого потока данных. Эта задача решается за счет использования параллельных вычислений с применением сопроцессоров (векторных, матричных, тензорных и др.) или ускорителей (графических, SoC и др.), к которым переходит (перехватывается сопроцессором или передается ускорителю командой центрального процессора) часть вычислительной работы.

Машины, реализующие параллельные вычисления, классифицируются по двум признакам: наличие параллелизма в потоках команд и данных (классификация Флинна), а также архитектура используемой памяти.

Классификация Флинна определяет четыре класса архитектур вычислительных машин по признакам наличия параллелизма в потоках команд и данных [6]. Для задачи построения системы управления движением интерес представ-

ляют две архитектуры, обеспечивающие реальное повышение производительности вычислений:

— SIMD (ОКМД) — вычислительная машина с одиночным потоком команд и множественным потоком данных. При использовании данной архитектуры параллельные вычисления реализуются за счет графических и других векторных ускорителей, сопроцессоров, матричных сопроцессоров, тензорных и других нейронных процессоров и др.

— MIMD (МКМД) — вычислительная машина с множественным потоком команд и множественным потоком данных. Для реализации данной архитектуры используют многопроцессорные и мультипроцессорные вычислительные машины, в которых процессоры или ядра обрабатывают множественные потоки данных.

Выбор той или иной архитектуры определяется степенью автономности отдельных подсистем управления движением. Использование архитектуры MIMD (МКМД) предпочтительнее в том случае (обычном для управления движением в промышленных роботах или станках), когда в системе управления движением реализуется одновременное управление большим числом параметров и элементов системы, существенная часть которых соответствует условию чрезвычайной параллельности (нет зависимости или связи между параллельными задачами, их результаты не влияют друг на друга).

Классификация по архитектуре используемой памяти устанавливает три основных варианта [7]: параллельные вычисления с общей памятью (например, многопроцессорные архитектуры класса SMP и NUMA); параллельные вычисления с распределенной памятью (массово-параллельные, кластерные и другие архитектуры); параллельные вычисления с многоуровневой (иерархической) памятью (например, современные SMP-системы, в которых кроме основной памяти имеются несколько уровней сверхоперативной кэш-памяти, встраиваемой в процессоры). Для систем управления движением актуальны все существующие варианты архитектур используемой памяти. Наибольшие возможности открывает применение многоуровневой памяти с выделением локальной оперативной памяти, используемой отдельными устройствами, а также общей оперативной памяти для обмена данными между устройствами.

Наряду с параллельными вычислениями существуют и другие пути уменьшения объема обрабатываемого потока данных. В частности, оп-

ределенные перспективы имеет развитие гарвардской архитектуры вычислительных машин с разделением памяти на память данных и память команд. Такая архитектура применяется, в частности, в самом быстродействующем процессорном кэше 1-го уровня (L1 cache). Гарвардская архитектура пока имеет ограниченное применение ввиду необходимости большого числа шин и неоптимального использования объема памяти. В настоящее время эти недостатки удалось в существенной степени преодолеть.

Задача 2. Необходимо увеличить скорость передачи данных между элементами вычислительной машины. Это может быть достигнуто за счет уменьшения расстояний между элементами, что расширяет возможности использования параллельного и секционного доступа к шине данных. Уменьшение расстояний при размещении всех элементов на одном чипе достигается в системах на кристалле (SoC). Снижение расстояний передачи данных между процессором и памятью достигается при использовании архитектуры обработки в памяти (PIM — processing-in-memory или processor-in-memory) [8].

Обработка в памяти — это один из подходов к преодолению узкого места архитектуры фон Неймана, представляющий собой ограничение пропускной способности, вызванное задержкой доступа к данным. Включение элементов обработки в архитектуру памяти сокращает расстояние передачи данных между процессором и памятью, тем самым сокращая время ожидания для доступа к памяти и энергопотребление. Технической реализацией PIM-архитектуры является процессор, интегрированный в память, как правило, на одном кремниевом кристалле, либо оперативная память с интегрированными вычисляющими элементами. Результат называют PIM-ядром (PIM core), PIM-машиной (PIM engine) или процессором в памяти (PIM — processor-in-memory) [9].

Задача 3. Необходимо устранить очереди при одновременном обращении к одной памяти нескольких вычислительных устройств. Возможны три подхода к решению данной задачи:

— использование предикативных алгоритмов исполнения команд (динамическое предсказание переходов, ветвлений, гибридные предикторы, спекулятивное выполнение команд и др.). Данный подход уже реализован на практике, в частности в новой микроархитектуре ядра Efficient, разработанного Intel, в котором предусмотрен кэш подсистемы предсказания ветвлений на 5000

записей [10]. Реализация более сложных предикативных алгоритмов, основанных на структурном анализе обращений к памяти вычислительного устройства (например, системы управления движением промышленного робота), предполагает включение в систему дополнительного вычислительного модуля. Принимая во внимание ограниченный потенциал достижимой экономии времени, использование таких предикативных алгоритмов в большинстве случаев экономически не оправдано;

– использование специальных систем диспетчеризации (например, матричных коммутаторов, омега-сети и др.). При этом возможно существенное сокращение задержки доступа к данным. Решение проблемы, однако, остается неполным. Кроме того, техническая сложность и стоимость быстродействующих систем диспетчеризации достаточно высоки;

– физическое разделение памяти или распределение памяти на (виртуальные) разделы. В первом случае возможно полное устранение задержек; во втором случае задержки могут быть существенно снижены (при общем снижении скорости работы), однако их полное устранение не представляется возможным.

Наибольший потенциал повышения быстродействия систем управления движением связан с использованием физического разделения памяти, обеспечивающего окончательное решение проблемы очередей, а также не требующее усложнения и повышения стоимости вычислительной системы.

Концептуальная модель системы управления движением

В модель-ориентированной системной инженерии (model-based systems engineering — MBSE) [11] под концептуальной моделью понимается «модель моделей» (метамодель, репрезентативная модель) выбранного класса объектов, которая должна отражать свойства и закономерности, одновременно присущие всему множеству их моделей. Затем уже, с использованием свойств этой метамодели, осуществляется построение моделей конкретных объектов этого класса.

Исходя из существующих представлений в области систем ЧПУ [12–14], роботов [15, 16] и систем управления роботами [17–19], можно предложить концептуальную модель системы управления движением промышленного робота (рис. 1).

1. Ядро системы управления движением, имеющее техническую реализацию в виде микроконтроллера (MCU) или компьютера (в частности, SoC). Одной из основных функций ядра являются интерпретация заданной рабочей программы (например, в виде G-кодов) и отправка управляющих команд на устройства системы управления. В случае существенно ограниченного функционала ядро системы управления движением эквивалентно ядру системы ЧПУ (NCK).

2. Блок очувствления, включающий систему технического зрения и модуль обработки аналоговых и дискретных входов, по которым в систему управления поступают данные с датчиков и информация от сенсоров. К блоку очувствления также относятся встроенные в сенсоры микропроцессоры, обеспечивающие обработку исходных данных и их преобразование в удобную для последующего использования информацию.

3. Исполнительный блок, включающий в себя модули для кинематических, динамических и других вычислений, необходимых для управления движением в промышленном роботе, а также регуляторы исполнительных устройств. Основные вычислительные модули: модуль трансформации (для решения в режиме реального времени прямой и обратной задачи кинематики для задания и определения позиции рабочего органа; для наиболее мощных систем управления — решение прямой и обратной задачи динамики для задания и определения положения и прилагаемого момента силы рабочего органа), модуль интерполяции (линейной, круговой, спиральной и др. — для определения промежуточных точек траектории движения рабочего органа), модуль разгона/торможения (расчет параметров приводов системы управления для автоматического трапецеидального ускорения/замедления, ускорения/замедления в виде S-образной кривой и др.), модуль эквидистантной коррекции — эквидистантный конвертор (выполняет расчет непрерывного контура с учетом радиуса инструмента), модуль предварительного просмотра кадров управляющей программы (Look-Ahead; позволяет читать некоторое число еще не обработанных кадров управляющей программы, анализировать резкие изменения направления движения и соответственно регулировать рабочую подачу) и др. Регуляторы исполнительных устройств — сервоприводы и другие исполнительные устройства с автоматической коррекцией состояния через

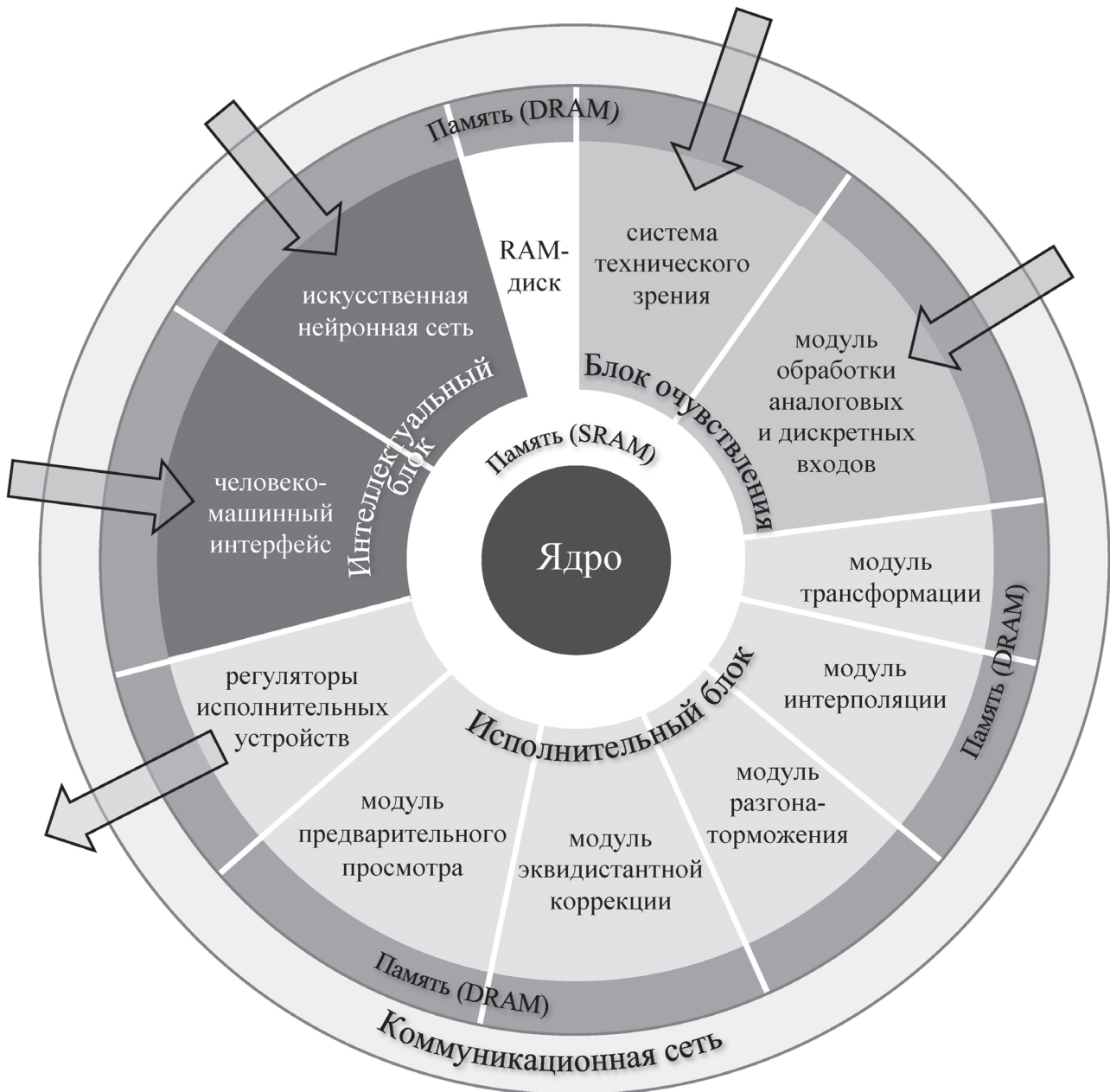


Рис. 1. Концептуальная модель системы управления движением промышленного робота

внутреннюю отрицательную обратную связь, в соответствии с параметрами, заданными извне.

4. Интеллектуальный блок, включающий в себя два модуля: человеко-машинный интерфейс (ММИ), обеспечивающий взаимодействие с системой управления человека-оператора, а также искусственный интеллект (AI) или систему машинного обучения (ML), реализованные в виде искусственной нейронной сети (ИНС).

Интеллектуальный блок ответственен за принятие общих управляющих решений, а также решение задач, которые не могут быть решены с использованием стандартных вычислительных

алгоритмов. К числу таких задач относится машинное зрение — частный случай технического зрения, связанный с технологиями искусственного интеллекта, используемыми для нахождения, отслеживания, классификации и идентификации визуальных объектов. Другой задачей, для решения которой необходимо использование ИНС, является распознавание человеческой речи, особенно востребованное при управлении коллаборативными роботами [20].

5. Оперативная память системы управления, которую условно можно разделить на три части: динамическая память с произвольным доступом

(DRAM), статическая память с произвольным доступом (SRAM), а также дополнительная виртуальная память в виде RAM-диска (реализуется на базе DRAM).

Динамическая память с произвольным доступом физически разделена на локальные области, к каждой из которых имеет доступ только одно устройство (модуль или устройство в составе модуля). Это позволяет полностью решить проблему очередей при обращении вычислительных устройств к памяти.

Общая оперативная память (кэш-память) системы реализуется в виде статической памяти с произвольным доступом. Общая оперативная память имеет небольшой объем и используется для обмена данными между различными устройствами системы управления и формирования (в зависимости от используемой инструментальной модели системы управления) виртуальных «пространства сообщений», «пространства событий» и др. Поскольку к общей оперативной памяти могут одновременно обращаться несколько устройств, неизбежно будут образовываться очереди. Для их минимизации необходимо использование системы диспетчеризации, а также памяти с наибольшей скоростью работы. Этим обстоятельством, а также необходимостью увеличения продолжительности времени сохранения данных в общей памяти, обусловлен выбор статической памяти с произвольным доступом, характеризующейся наибольшим быстродействием и сохранением (в процессе работы) данных без перезаписи.

При работе системы управления в ряде случаев могут накапливаться значительные объемы взаимосвязанной информации, которая должна быть доступна для высокоскоростного чтения и дополнения. Такая необходимость, например, может возникать при работе искусственной нейронной сети с большими данными или при формировании цифрового двойника промышленного робота (в рамках решения оптимизационных задач и других задач моделирования). В таких случаях может возникнуть необходимость включения в систему управления дополнительной памяти в виде RAM-диска — виртуального диска, хранящегося в оперативной памяти.

6. Коммуникационная сеть, представляющая собой систему цифровых и аналоговых каналов связи, коммутационного оборудования и устройств для преобразования сигналов (цифроаналоговых, аналого-цифровых и др.), реализующая заданный низкоуровневый протокол пере-

дачи данных между устройствами внутри системы управления, а также между устройствами системы управления и внешними устройствами (сервоприводами, датчиками, сенсорами и др.). Скорости передачи данных по коммуникационной сети могут существенно различаться в зависимости от используемых видов каналов связи. Наименьшая скорость — при использовании беспроводной связи (для передачи данных на большие расстояния), наибольшая — при использовании высокоскоростных оптических кабелей.

Несмотря на то, что коммуникационная сеть связывает устройства, входящие в систему управления, она не может заменить общую оперативную память. Вариант взаимодействия устройств через коммуникационную сеть соответствует модели параллельных вычислений «обмен сообщениями» [21], функциональные возможности которой существенно ограничены, особенно в части выполнения сложных протоколов, требующих синхронизации работы нескольких устройств. Поэтому для систем управления движением в режиме реального времени необходимы оба механизма обмена данными между устройствами: через общую оперативную память, а также через коммуникационную сеть.

Предлагаемая концептуальная модель системы управления движением является память-центрической, т.е. реализует память-центрическую архитектуру вычислений, при которой данные в процессе вычислений не перемещаются между процессором и памятью (что требует затрат времени, энергии и ограничено пропускной способностью каналов связи), а остаются в памяти, в которую интегрируется процессор. Данные, перемещаемые между различными вычислительными устройствами системы управления, представляют собой результаты вычислений, имеющие незначительный объем.

Из анализа предлагаемой концептуальной модели системы управления движением промышленного робота можно видеть, как решаются сформулированные нами основные задачи повышения производительности:

1. *Задача уменьшения объема обрабатываемого потока данных.* Решается за счет использования дополнительных вычислительных модулей — сопроцессоров и ускорителей, выполняющих параллельные вычисления. Основная часть вычислений осуществляется без управления из ядра системы.

2. *Задача увеличения скорости передачи данных между функциональными элементами и блоками*

системы управления. Решается путем использования память-центрической архитектуры (PIM). Все устройства, для работы которых необходима высокая скорость обмена данными с памятью, интегрируются в память (каждое устройство в свою локальную DRAM). Кроме того, для обеспечения максимальной скорости передачи данных между устройствами PIM могут устанавливаться на одну интегральную схему, формируя общую систему на кристалле (SoC). Наряду с установкой на одном кристалле (решение, обеспечивающее наиболее высокое быстродействие) PIM и другие элементы системы управления могут связываться с общей памятью и друг с другом через коммуникационную сеть: последовательно-параллельную шину данных, промышленную сеть (на базе Ethernet или оптико-волоконной связи) [22] или беспроводные сети (Wi-Fi, WiMAX, Bluetooth). Если подключаемые таким образом устройства работают автономно (обращения к общей памяти в режиме реального времени не требуются), то быстродействие системы управления движением может остаться высоким.

3. *Задача устранения очередей.* Решается путем разделения оперативной памяти (DRAM) на локальные области, доступные только одному устройству. Взаимодействие между устройствами осуществляется в высокоскоростной статической памяти (SRAM) с использованием минимальных объемов данных, а также через коммуникационную сеть, обеспечивающую прямую связь между устройствами без возникновения задержек.

Инструментальная модель системы управления движением

Для описания работы системы управления движением необходима соответствующая инструментальная модель, представляющая собой средство построения, исследования и/или использования прагматических и/или познавательных моделей. В свою очередь, прагматическая модель (обычно прикладная) — это средство организации практических действий, рабочего представления целей системы для ее управления, а познавательная модель (обычно теоретическая) — форма организации и представления знаний, средство соединения новых и старых знаний [23].

Адекватной инструментальной моделью работы системы управления движением, обеспечивающей эмуляцию параллельных вычислений и взаимодействия функциональных модулей, является модель акторов [24, 25]. Актор — универ-

сальный примитив исполнения, наделенный заданными свойствами и взаимодействующий посредством обмена сообщениями с другими акторами, обеспечивая совместно с ними функционирование системы. Акторы представляют собой виртуальные сущности, либо могут иметь физическую реализацию в виде процессора или иного устройства.

Акторная модель системы управления движением промышленного робота (рис. 2) включает в себя акторы, соответствующие элементам (модулям) концептуальной модели, рассмотренной нами выше:

- ядро — актор, генерирующий управляющие сообщения, задающие исходную программу и (через модули исполнительного блока) изменения работы исполнительных устройств;

- вычислительные модули (BM-1, BM-2...) — акторы, выполняющие вычисления и обработку данных, полученных из ядра, для получения информации, передаваемой регуляторам исполнительных устройств (РИУ-1, РИУ-2...). Получаемые из ядра данные включают в себя интерпретированные данные программы работы, а также данные от различных датчиков;

- регуляторы исполнительных устройств (РИУ-1, РИУ-2...) — акторы, получающие от вычислительных модулей информацию по заданной (требуемой) активности исполнительных устройств (например, перемещении и скорости по осям рабочих органов) и посылающие управляющие сигналы исполнительным устройствам (ИУ-1, ИУ-2...) через коммуникационную сеть;

- модули оцувствления (МО-1, МО-2...) — акторы, содержащие данные о состоянии исполнительных устройств (и робота в целом), полученные от датчиков (Д-1, Д-2...) через коммуникационную сеть, и передающие эти данные в ядро напрямую либо (например, в случае передачи информации от системы технического зрения) через искусственную нейронную сеть (ИНС), где эти данные интерпретируются;

- искусственная нейронная сеть (ИНС) — актор, ответственный за решение интеллектуальных задач (распознавание образов, человеческой речи и др.). В перспективе может взять на себя часть наиболее сложных вычислительных задач;

- человеко-машинный интерфейс (ЧМИ) — актор, получающий информацию от человека-оператора, датчиков, ИНС, регуляторов исполнительных устройств и передающий информацию (команды человека-оператора) в ядро системы;

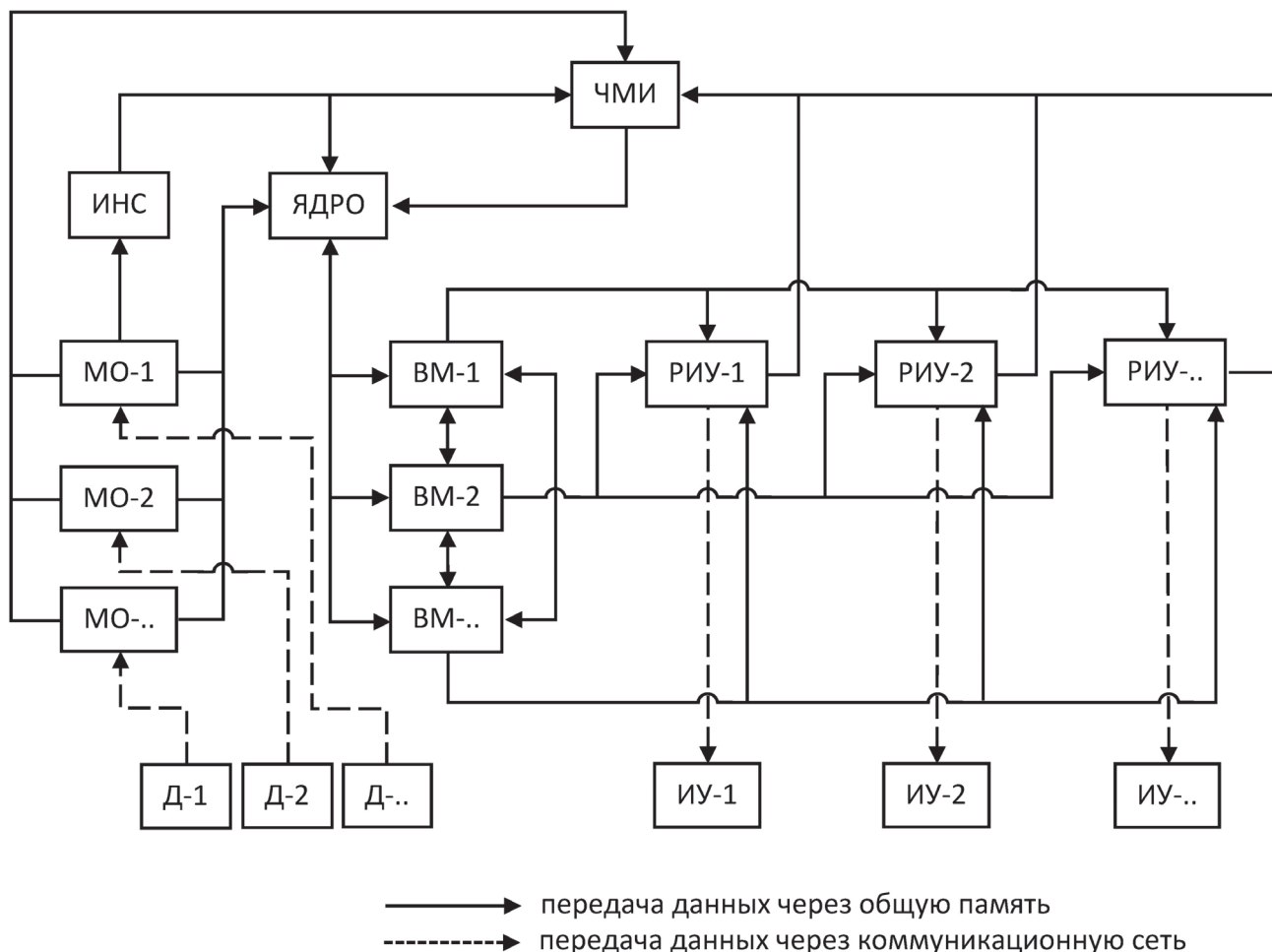


Рис. 2. Акторная модель системы управления движением промышленного робота

• датчики (Д-1, Д-2...) и исполнительные устройства (ИУ-1, ИУ-2...) не входят в систему управления движением и поэтому не являются акторами.

Развитием модели акторов является модель реляционных акторов (реакторов) [26, 27], управляемых событиями, а не сообщениями, как в базовой модели акторов. Реакторы могут подписываться и реагировать на потоки асинхронно возникающих событий. При этом реактор может реагировать на события из множества разных источников в любое время. Благодаря использованию событий в качестве инструмента управления существенно упрощается композиция протоколов исполнения, т.е. объединение в рамках одного протокола исполнения методов, свойств и других требований, которые соответствуют определенному конкретному заданию или функции. К числу важных достоинств модели реакторов также относится возможность объединения (упаковки) операций потока событий в модули,

которые затем могут быть многократно использованы [28].

Как акторная, так и реакторная инструментальные модели хорошо соответствуют архитектуре и характеру взаимодействия элементов системы управления движением промышленного робота. В обеих моделях обмен данными должен происходить в общей памяти, куда передаются сообщения (в модели акторов) или возникают события (в модели реакторов). При этом сообщения принимаются только теми акторами, которые имеют адрес отправителя сообщения; все прочие акторы сообщение не обнаруживают. События обнаруживаются только теми реакторами, которые подписаны на данный вид событий.

Выводы

Для практической реализации систем управления движением промышленных роботов в режиме реального времени необходима высокая производительность проблемно-ориентирован-

ных (кинематических, динамических и нейроморфных) вычислений при минимальном (не более нескольких десятков системных тактов) и детерминированном (отсутствие джиттера) времени реакции на системное прерывание в процессе поддержания цикла управления не более нескольких десятков микросекунд. Такая производительность в настоящее время не может быть обеспечена при традиционных подходах к построению систем управления.

Решение проблемы повышения производительности вычислений необходимо искать в области совершенствования архитектуры систем управления.

Основными задачами повышения производительности, имеющими решение в результате совершенствования архитектуры вычислительных машин, являются: уменьшение объема обрабатываемого потока данных, увеличение скорости передачи данных, устранение очереди при одновременном обращении к одной памяти нескольких вычислительных устройств.

Указанные задачи решаются при построении систем управления движением промышленных роботов в соответствии с предлагаемой в статье концептуальной память-центрической моделью.

В качестве инструментальной модели системы управления движением промышленного робота предлагается использовать модель акторов или модель реакторов, хорошо соответствующие архитектуре и характеру взаимодействия элементов системы управления движением промышленного робота.

Список источников

1. Указ Президента Российской Федерации от 01.12.2016 №642 «О Стратегии научно-технологического развития Российской Федерации». URL: <http://www.kremlin.ru/acts/bank/41449>
2. Гришин Д.В. Разработка эффективных форм кадрового обеспечения производственного процесса в авиастроительной отрасли // Вестник Московского авиационного института. 2018. Т. 25. № 1. С. 209-219.
3. Ребров С.Г., Янчур С.В., Дрондин А.В., Зернов О.Д. Разработка концепции роботизированной сборки солнечных энергоустановок на орбите // Вестник Московского авиационного института. 2019. Т. 26. № 1. С. 201-211.
4. Tay S., Chuan L., Aziati A., Ahmad A. An Overview of Industry 4.0: Definition, Components, and Government Initiatives // Journal of Advanced Research in Dynamical and Control Systems. 2018. Vol. 10. Special Issue 14, pp. 1379-1387.
5. Leiserson C.E., Thompson N.C., Emer J.S. et al. There's plenty of room at the Top: What will drive computer performance after Moore's law? // Science. 2020. Vol. 368. No. 6495. DOI: 10.1126/science.aam9744
6. Flynn M. Some Computer Organizations and Their Effectiveness // IEEE Transactions on Computers. 1972. Vol. 21. No. 9, pp. 948-960 DOI: 10.1109/TC.1972.5009071
7. Ежова Н.А., Соколинский Л.Б. Обзор моделей параллельных вычислений // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2019. Т. 8. №3. С. 58-91. DOI: 10.14529/cmse190304
8. Connolly M. A Programmable Processing-in-Memory Architecture for Memory Intensive Applications. Master's Thesis on Computer Engineering. Rochester Institute of Technology, 2021, 42 p.
9. Ghose S., Hsieh K., Boroumand A., Ausavarungnirun R., Mutlu O. Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms, Future Research Directions. 2018, 45 p. URL: CoRR abs/1802.00320
10. Бахур В. Intel представила большие архитектурные изменения в новых поколениях процессоров. 2021. URL: https://www.cnews.ru/news/line/2021-08-19_intel_predstavila_bolshie_arhitekturnye
11. Estefan J.A. Survey of model-based systems engineering (MBSE) methodologies // IncoSE MBSE Focus Group. 2008, 70 p. URL: http://www.omg-sysml.org/MBSE_Methodology_Survey_RevB.pdf
12. Suh S.H., Kang S.K., Chung D.H., Stroud I. Theory and design of CNC systems. — Springer Science & Business Media, 2008. — 476 p.
13. Зеленский А.А., Харьков М.А., Ивановский С.П., Абдуллин Т.Х. Высокопроизводительная система числового программного управления на базе программируемых логических интегральных схем // Вестник Воронежского государственного технического университета. 2018. Т. 14. №5. С. 8-12.
14. Мартинов Л.И., Мартинов Г.М. Организация межмодульного взаимодействия в распределенных системах ЧПУ. Модели и алгоритмы реализации // Мехатроника, автоматизация, управление. 2010. №11. С. 50-55.
15. Фу К., Гонсалес Р., Лу К. Робототехника / Пер. с англ. А.А. Сорокина и др.; Под ред. В.Г. Градецкого. — Мир, 1989. — 620 с.
16. Ruishu Z., Chang Z., Weigang Z. The status and development of industrial robots // 4th International Conference on Applied Materials and Manufacturing Technology (25–27 May 2018, Nanchang, China). 2018. Vol. 423. No. 1. DOI: 10.1088/1757-899X/423/1/012051
17. Романов А.М. Обзор аппаратно-программного обеспечения систем управления роботов различного масштаба и назначения. Ч. 1. Промышленная робототехника // Российский технологический журнал. 2019. Т. 7. № 5(31). С. 30–46. DOI: 10.32362/2500-316X-2019-7-5-30-46

18. Зеленский А.А., Абдуллин Т.Х., Илюхин Ю.В., Харьков М.А. Высокопроизводительная цифровая система на основе ПЛИС для управления движением многокоординатных станков и промышленных роботов // СТИН. 2019. №8. С. 5-8.
19. Илюхин Ю.В. Синергетический (мехатронный) подход к проектированию систем управления технологических роботов // Мехатроника. 2000. № 2. С. 7-12.
20. Zelenskii A.A., Pismenskova M.M., Voronin V.V. Control of collaborative robot systems and flexible production cells on the basis of deep learning // Russian Engineering Research. 2019. Vol. 39. No. 12, pp. 1065-1068 DOI:10.3103/S1068798X19120256
21. Модели параллельных вычислений. URL: <http://www.ccas.ru/paral/prog/models.html>
22. Зеленский А.А., Шадрин Н.Г., Абдуллин Т.Х., Харьков М.А. Высокая скорость промышленной сети реального времени киберфизических систем // Вестник компьютерных и информационных технологий. 2019. №11. С. 46-52.
23. Попова Н.В. Математические методы: электронный учебник. 2005. URL: <http://matmetod-popova.narod.ru/Index1.htm>
24. Burgin M. Systems, Actors and Agents: Operation in a multicomponent environment. 2017, 28 p. URL: [arXiv:1711.08319](https://arxiv.org/abs/1711.08319)
25. Rinaldi L., Torquati M., Mencagli G., Danelutto M., Menga T. Accelerating Actor-based Applications with Parallel Patterns // 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing. 2019, pp 140-147. DOI: 10.1109/EMPDP.2019.8671602
26. Shah V., Vaz Salles M.A. Reactors: A case for predictable, virtualized actor database systems // International Conference on Management of Data. 2018, pp. 259-274. DOI: 10.1145/3183713.3183752
27. Lohstroh M., Menard C., Bateni S., Lee E. Toward a Lingua Franca for Deterministic Con-current Systems // ACM Transactions on Embedded Computing Systems. 2021. Vol. 20. No. 4, pp. 1-27. DOI:10.1145/3448128
28. Laigner R., Lifschitz S., Kalinowski M., Poggi M., Vaz Salles M.A. Towards a Technique for Extracting Relational Actors from Monolithic Applications // Anais do XXXIV Simpósio Brasileiro de Banco de Dados. 2019, pp. 133-144. DOI: 10.5753/sbbd.2019.8814

References

1. О Стратегии научно-технологического развития Российской Федерации. Указ Президента Российской Федерации от 01.12.2016, № 642 (On Strategy of Scientific and Technological Development of the Russian Federation. Decree of the President of the Russian Federation dated 01.12.2016 no. 642). URL: <http://www.kremlin.ru/acts/bank/41449>
2. Grishin D. V. Development of effective forms of production process stuffing in aircraft building industry. *Aerospace MAI Journal*, 2018, vol. 25, no. 1, pp. 209-219.
3. Rebrov S.G., Yanchur S.V., Drondin A.V., Zernov O.D. Developing the concept of solar energy units robotic assembly in orbit. *Aerospace MAI Journal*, 2019, vol. 26, no. 1, pp. 201-211.
4. Tay S., Chuan L., Aziati A., Ahmad A. An Overview of Industry 4.0: Definition, Components, and Government Initiatives. *Journal of Advanced Research in Dynamical and Control Systems*, 2018, vol. 10, special issue 14, pp. 1379-1387.
5. Leiserson C.E., Thompson N.C., Emer J.S. et al. There's plenty of room at the Top: What will drive computer performance after Moore's law? *Science*, 2020, vol. 368, no. 6495. DOI: 10.1126/science.aam9744
6. Flynn M. Some Computer Organizations and Their Effectiveness. *IEEE Transactions on Computers*, 1972, vol. 21, no. 9, pp. 948-960. DOI: 10.1109/TC.1972.5009071
7. Ezhova N.A., Sokolinskii L.B. *Vestnik YuUrGU. Seriya Vychislitel'naya matematika i informatika*, 2019, vol. 8, no. 3, pp. 58-91. DOI: 10.14529/cmse190304
8. Connolly M. *A Programmable Processing-in-Memory Architecture for Memory Intensive Applications*. Master's Thesis on Computer Engineering. Rochester Institute of Technology, 2021, 42 p.
9. Ghose S., Hsieh K., Boroumand A., Ausavarungnirun R., Mutlu O. *Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms, Future Research Directions*. 2018, 45 p. URL: [CoRR abs/1802.00320](https://arxiv.org/abs/1802.00320)
10. Bakhur V. *Intel predstavila bol'shie arkhitekturnye izmeneniya v novykh pokoleniyakh protsessorov*, 2021. URL: https://www.cnews.ru/news/line/2021-08-19_intel_predstavila_bolshie_arhitekturnye
11. Estefan J.A. Survey of model-based systems engineering (MBSE) methodologies. *IncoSE MBSE Focus Group*, 2008, 70 p. URL: http://www.omg-sysml.org/MBSE_Methodology_Survey_RevB.pdf
12. Suh S.H., Kang S.K., Chung D.H., Stroud I. *Theory and design of CNC systems*. Springer Science & Business Media, 2008, 476 p.
13. Zelenskii A.A., Khar'kov M.A., Ivanovskii S.P., Abdullin T.Kh. *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta*, 2018, vol. 14, no. 5, pp. 8-12.
14. Martinov L.I., Martinov G.M. *Mekhatronika, avtomatizatsiya, upravlenie*, 2010, no. 11, pp. 50-55.
15. Fu K.S., Gonzalez R.C., Lee C.S.G. *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company, 1987, 580 p.
16. Ruishu Z., Chang Z., Weigang Z. The status and development of industrial robots. *4th International Conference on Applied Materials and Manufacturing*

- Technology* (25–27 May 2018, Nanchang, China), 2018, vol. 423, no. 1. DOI: 10.1088/1757-899X/423/1/012051
17. Romanov A.M. *Rossiiskii tekhnologicheskii zhurnal*, 2019, vol. 7, no. 5(31), pp. 30–46. DOI: 10.32362/2500-316X-2019-7-5-30-46
 18. Zelenskii A.A., Abdullin T.Kh., Ilyukhin Yu.V., Khar'kov M.A. *STIN*, 2019, no. 8, pp. 5–8.
 19. Ilyukhin Yu.V. *Mekhatronika*, 2000, no. 2, pp. 7–12.
 20. Zelenskii A.A., Pismenskova M.M., Voronin V.V. Control of collaborative robot systems and flexible production cells on the basis of deep learning. *Russian Engineering Research*, 2019, vol. 39, no. 12, pp. 1065–1068. DOI:10.3103/S1068798X19120256
 21. *Modeli parallel'nykh vychislenii*. URL: <http://www.ccas.ru/paral/prog/models.html>
 22. Zelenskii A.A., Shadrin N.G., Abdullin T.Kh., Khar'kov M.A. *Vestnik komp'yuternykh i informatsionnykh tekhnologii*, 2019, no. 11, pp. 46–52.
 23. Popova N.V. *Matematicheskie metody: elektronnyi uchebnik* (Mathematical methods: electronic textbook), 2005. URL: <http://matmetod-popova.narod.ru/Index1.htm>
 24. Burgin M. *Systems, Actors and Agents: Operation in a multicomponent environment*. 2017, 28 p. URL: [arXiv:1711.08319](https://arxiv.org/abs/1711.08319)
 25. Rinaldi L., Torquati M., Mencagli G., Danelutto M., Menga T. Accelerating Actor-based Applications with Parallel Patterns. *27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, 2019, pp. 140–147. DOI: 10.1109/EMPDP.2019.8671602
 26. Shah V., Vaz Salles M.A. Reactors: A case for predictable, virtualized actor database systems. *International Conference on Management of Data*, 2018, pp. 259–274. DOI: 10.1145/3183713.3183752
 27. Lohstroh M., Menard C., Bateni S., Lee E. Toward a Lingua Franca for Deterministic Con-current Systems. *ACM Transactions on Embedded Computing Systems*, 2021, vol. 20, no. 4, pp. 1–27. DOI:10.1145/3448128
 28. Laigner R., Lifschitz S., Kalinowski M., Poggi M., Vaz Salles M.A. Towards a Technique for Extracting Relational Actors from Monolithic Applications. *Anais do XXXIV Simpósio Brasileiro de Banco de Dados*, 2019, pp. 133–144. DOI: 10.5753/sbbd.2019.8814

Статья поступила в редакцию 12.10.2021; одобрена после рецензирования 18.10.2021; принята к публикации 18.10.2021.

The article was submitted on 12.10.2021; approved after reviewing on 18.10.2021; accepted for publication on 18.10.2021.