



УДК 004.021, 004.415

© 2017 г. **В.А. Бобков**, д-р техн. наук,
С.В. Мельман, канд. техн. наук,
М.А. Морозов,
Г.В. Тарасов

(Институт автоматизации и процессов управления ДВО РАН, Владивосток)

МОДЕЛИРУЮЩИЙ КОМПЛЕКС ДЛЯ ПОДВОДНОГО РОБОТА НА РАСПРЕДЕЛЕННОЙ АРХИТЕКТУРЕ С ИСПОЛЬЗОВАНИЕМ КЛАСТЕРА*

Представлена версия распределенного моделирующего комплекса (МК) для автономных подводных роботов (АПР) с использованием многопроцессорного вычислительного кластера. Проанализированы особенности режима работы кластера в составе МК с распределенными вычислительными узлами, объединенными в локальную сеть. Рассмотрены вопросы имитационного моделирования сенсорного оборудования на борту АПР и реализации алгоритмов реалистичной визуализации подводной обстановки для разных режимов.

Ключевые слова: моделирующий комплекс, виртуальная реальность, распределенная архитектура.

DOI: 10.22250/isu.2017.54.32-42

Введение

В настоящее время подводные роботы активно используются для выполнения широкого круга практически важных работ. Программно-алгоритмический интеллект автономного подводного робота (АПР) должен обеспечивать формирование и отработку рабочих миссий, автоматическое планирование траектории движения в условиях априори неизвестной обстановки. До установки на борт АПР все функции и алгоритмы этой системы должны быть тщательно протестированы. Для тестирования на первых этапах активно используется имитационное моделирование, то есть проведение испытаний АПР в виртуальной среде. Это ус-

* Работа выполнена при финансовой поддержке РФФИ (проект № 16-07-00350).

коряет и удешевляет натурные испытания, обеспечивая при этом достоверность выводов об эффективности применяемых алгоритмов и миссии в целом. О важности использования имитационных сред при разработке роботов свидетельствует большое количество исследований, создаваемых сред и фреймворков для симуляции при разработке и тонкой настройке поведения роботов [1 – 3]. Сложность разрабатываемых систем закономерно приводит к использованию таких технологий как параллельные вычисления, распределенные, облачные [4] и вычисления GPGPU.

Однако, несмотря на большое количество попыток [1 – 6] создания универсальных подходов к построению систем симуляции, до сих пор остается открытым вопрос тонкой настройки на конкретный класс задач. Авторами был разработан моделирующий комплекс [7], ориентированный на моделирование аппаратно-программной части системы управления подводными роботами разработки ИМПТ ДВО РАН. Модульная архитектура [8] позволяет использовать широкий набор бортового оборудования и датчиков – от компаса и сонаров [9] и до реалистичной симуляции работы подводных фото- и видеокамер. Распределенные вычисления [10] позволяют вычислительно сложным алгоритмам работать в режиме реального времени на разных узлах, когда симуляция производится со скоростью не меньшей, чем была бы при натурных испытаниях. В данной работе предлагается развитие МК, направленное на использование многопроцессорного вычислительного кластера в составе МК.

Схема функционирования распределенного МК

В ходе имитационного моделирования рабочей миссии выполняется большое количество процессов:

- моделирование бортового оборудования;
- моделирование динамики аппарата;
- визуализация процесса для оператора;
- сбор и хранение информации о показаниях датчиков для дальнейшего анализа;
- работа алгоритмов планирования траектории и управления.

Последний предполагает обработку на бортовом вычислителе ПР, остальные призваны подменять реальную информацию из подводной среды и могут быть реализованы на моделирующей ЭВМ. Чем выше точность (достоверность) имитации, тем точнее можно будет прогнозировать исход реальной миссии, но высокая точность требует серьезных вычислительных затрат. Для режима НП (реальное оборудование в цикле имитации) необходимо, чтобы моделирование шло в режиме реального времени, что требует реализации распределенно-параллельных вычислений с применением суперкомпьютера. В предыдущих ра-

ботах авторов [7 – 10] был описан подход к построению МК с распределенными вычислительными узлами, объединенных локальной сетью (рис. 1).

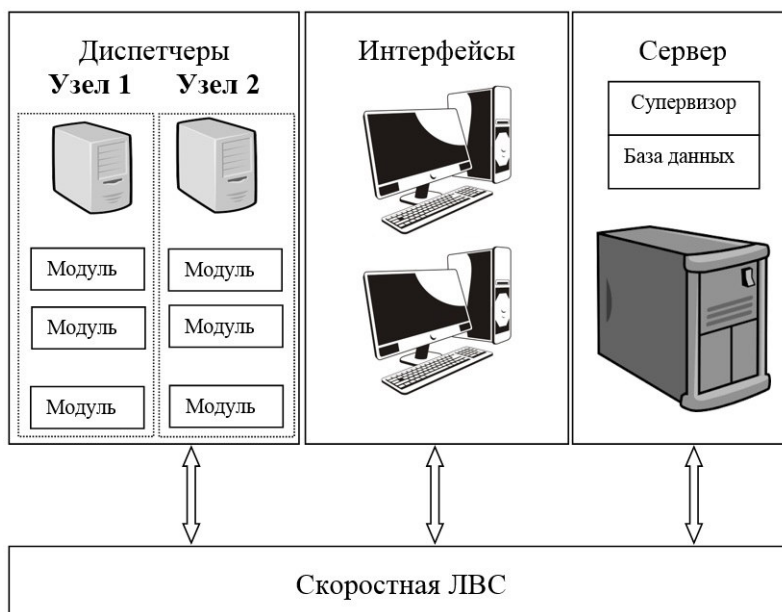


Рис. 1. Схема распределенного моделирующего комплекса.

Таковыми узлами могут быть как персональные ЭВМ, так и кластер супер-ЭВМ. Распределенный МК состоит из трех программных компонент:

сервер (включает базу данных и супервизор, координирующий работы распределенной системы);

диспетчер (компонент распределенной системы, позволяющий объединять вычислительные модули на узле);

приложение – реализующее интерфейс пользователя.

Функциональные возможности МК

Применение МК значительно упрощает решение задач по разработке, исследованию и отладке алгоритмов и методов, используемых в интеллектуальной начинке робота. Примерами таких задач являются:

алгоритмы навигации;

поиск затонувших объектов, обзорные миссии;

построение карт местности;

алгоритмы «интеллектуального поведения» (планирование траектории движения, обход препятствий, обработка внештатных ситуаций, групповые работы).

Это возможно благодаря имитации всех бортовых систем подводного аппарата и окружающей среды в МК.

Другие функциональные возможности позволяют:

использовать МК в качестве тренажерного комплекса для операторов АПР;

задавать и исследовать миссии АПР в изменяемых условиях;

тестировать работоспособность оборудования и бортового ПО подводного аппарата при его подключении к виртуальной среде моделирующего комплекса в режиме ННЛ.

Система предоставляет, наряду с возможностями моделирования и управления миссиями, также развитые инструменты визуализации результатов моделирования, позволяя выбрать любой момент миссии для изучения как информации с сенсоров, так и визуальной информации в 3D. Наличие централизованной БД дает возможность реализовать многопользовательский доступ к результатам моделирования для совместного анализа их несколькими операторами.

Реалистичная визуализация

Виды искажений. При моделировании фото- и видеоаппаратуры используется аппаратная поддержка OpenGL для визуализации подводной среды. Реалистичность изображения зависит от детализации описания мира с помощью триангуляционных моделей, текстур и качества подпрограмм отрисовки сцены. Применение высокодетализированных сцен и текстур неприемлемо для реалистичной визуализации, и традиционно требуется компромисс между качеством и скоростью рендеринга. В моделирующем комплексе речь идет об имитации аппаратуры для отработки/обкатки алгоритмов компьютерного зрения (КЗ), таких как: навигация и планирование траектории. В компьютерной графике при визуализации используется модель камеры обскуры, но реальная фототехника далека от этого идеала. Зная специфику работы алгоритмов КЗ, можно выделить три основных фактора, влияющих на качество их работы: геометрические искажения, вносимые линзами камеры; шумы в фоточувствительном элементе; освещенность сцены. Таким образом, под «реалистичностью» в контексте моделирующего комплекса подразумевается имитация этих искажений.

Моделирование шума камеры. Шум, возникающий в сенсоре видеокамеры, обусловлен несовершенством технологий и квантовой природой света. Как правило, для практических целей исследование шума направлено на улучшение изображения [11]. Выделяются три основные компоненты шума – линейная, структурная и случайная. Наиболее значимые искажения вносятся случайной составляющей. Для нашей задачи экспериментально было установлено, что использование нормального (гауссова) распределения для добавление аддитивного белого шума к идеальному изображению вполне достаточно для имитации шума цифровой матрицы. С учетом особенностей нелинейного преобразования сигнала на фоточувствительных элементах предложена следующая схема вычисления аддитивного случайного шума. Изначально виртуальной камере присваивалось значе-

ние «величина зашумленности» σ , которая в гауссовом распределении играет роль дисперсии. Так как шум – это аддитивная компонента, математическое ожидание выбирается равным нулю, итоговое значение в пикселе вычисляется по формуле

$$I = I_p + \frac{G(\sigma)}{(k \cdot I_p + 1)},$$

где I_p – интенсивность пикселя незашумленного изображения; $G(\sigma)$ – функция генерации случайной нормально распределенной величины; k – коэффициент нелинейности при исходной интенсивности пикселя. Далее I – итоговая интенсивность пикселя – приводится к интервалу $[0, 1]$. Смысл коэффициента k – в ослаблении влияния шума в зависимости от начальной интенсивности пикселя и выбирается эмпирически.

Моделирование геометрических искажений объектива камеры. Для имитации несовершенства технологических процессов при производстве оптической системы и других компонентов камеры используются три вида геометрических искажений: радиальные, тангенциальные, смещение оптического центра и соотношения сторон кадра.

Радиальные искажения возникают из-за несовершенства оптической системы (объектива) камеры. Наличие радиальных искажений приводит к тому, что прямые линии на снимках становятся дугообразными.

Модель радиальных искажений:

$$x_{Rdistorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6),$$

$$y_{Rdistorted} = y(1 + k_4 r^2 + k_5 r^4 + k_6 r^6),$$

где $r = x^2 + y^2$, k_i – коэффициенты.

Другой вид искажений – тангенциальные. Возникают вследствие неточностей при монтаже сенсорной матрицы камеры относительно объектива, именно – непараллельности плоскости матрицы и линз.

Модель тангенциальных искажений:

$$x_{Tdistorted} = x + (p_2(r^2 + 2x^2) + 2p_1xy),$$

$$y_{Tdistorted} = y + (p_1(r^2 + 2y^2) + 2p_2xy),$$

где p_i – коэффициенты.

Последние два искажения можно записать в виде:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{Odistorted} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \\ c_x & c_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix},$$

где (f_x, f_y) – информация о фокусе; (c_x, c_y) – об оптических центрах матрицы.

Суммарные геометрические искажения – это последовательно выполненные приведенные операции. Так как эти искажения постоянны, то можно один раз вычислить таблицу сдвигов для изображения и применять к каждому изображению,

сформированному в процессе моделирования.

Моделирование источника света. При моделировании подводных сцен следует учитывать, что проникновение солнечного света ограничивается прозрачностью воды. Для большинства случаев наличие естественного освещения возможно до глубин порядка 10-20 м, в идеальных условиях чистой воды – до 40-60 м. Причем уровень освещенности падает с ростом глубины (рис. 2).

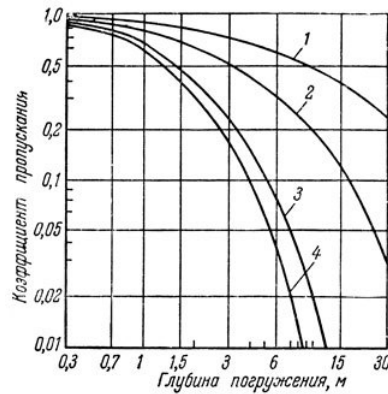


Рис. 2. Графики зависимости пропускаемого светового потока от глубины (в метрах):
1 – океанская вода высокой прозрачности; 2 – океанская вода средней прозрачности;
3 – прибрежная вода средней прозрачности; 4 – мутная вода [12].

Поглощение света в воде зависит от длины волны (рис. 3). Наиболее сильное поглощение происходит в красной части спектра, лучше всего в водную толщу проходят зеленые лучи.

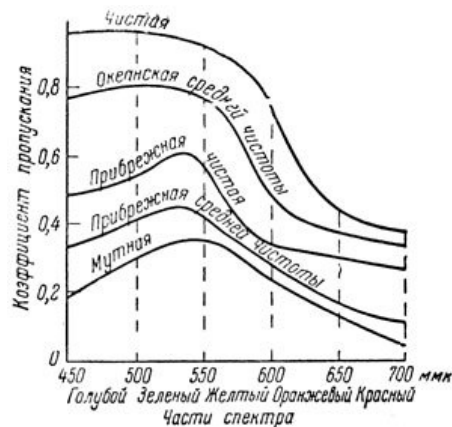


Рис. 3. Графики коэффициента пропускания, зависящие от длины волны, при пути света в воде 3 м [12].

Из графиков, приведенных на рис. 2 и 3, следует, что для больших глубин необходимо искусственное бортовое освещение. Источник света должен быть смонтирован рядом с камерой и направлен параллельно оси съемки. Диаграмма направленности источника света должна быть согласована с углом зрения камеры, в результате можно добиться равномерного освещения сцены. Таким образом, в сцене имеются два источника света – естественное освещение, затухающее с

глубиной, и точечный источник, закрепленный на аппарате.

Расчет освещенности возлагается на шейдер, который учитывает два источника света: с постоянным вектором на источник – «солнце» и с вектором, указывающим на бортовой источник, который движется, закрепленный на аппарате. В шейдере учитывается затухание света, представленное графиками на рис. 2, 3. Для источника, имитирующего солнце, берется глубина объекта под уровнем моря, для источника на аппарате – расстояние от источника до объекта, вычисляемое по буферу глубины. Пример работы модуля приведен на рис. 4.



Рис. 4. Пример работы модуля: моделирование фотореалистичного изображения подводной обстановки.

Кластер в качестве вычислительного узла

Традиционная схема построения вычислительных кластеров состоит в соединении однотипных вычислительных узлов несколькими высокоскоростными сетями. Вычислительные ресурсы (процессорные ядра и память) этой группы узлов часто называют вычислительным (или расчетным) полем, обозначая, таким образом, общие возможности кластера для исполнения параллельных программ. Одним из главных факторов, влияющих на скорость расчетов, является скорость доступа к памяти в процессе вычислений. Структура памяти вычислительной машины носит иерархический характер, который можно представить в виде пирамид с ранжированием уровней пирамиды по скорости доступа. Вершиной пирамиды является внутренняя память ядра с минимальным временем доступа (регистры, кэш-память), ниже уровнем идет оперативная память узла, связанная через контроллер памяти с ядрами процессора. Комплекс устройств, связывающих все ядра и память узла, называют внутренним интерконнектом. Еще ниже уровнем идет внешняя по отношению к узлу память, доступ к которой осуществ-

ляется через дополнительное сетевое устройство, подключенное к процессору через внутренний интерконнект. Под внешней памятью можно понимать память соседнего узла или дисковую память системы хранения данных, связанные вместе внешним интерконнектом. Для внешнего интерконнекта используются сверхвысокоскоростные сетевые технологии с минимальным показателем латентности: Infiniband FDR или EDR (до 100 Гбит/с), Intel Omni-Path (100 Гбит/с) и другие. Таким образом, кластер, построенный на традиционной процессорной архитектуре – это группа ядер и памяти, связанных между собой общим интерконнектом. На программном уровне управление ресурсами обеспечивается специальным программным обеспечением. Каждый узел работает под управлением собственного образа операционной системы – здесь нет отличий от обычного сервера. А вот на уровне запуска прикладных программ в операционной системе работает специальный процесс, который следит за тем, какой вычислительной программе и сколько выделяется ядер и памяти. Также он сопровождает ход исполнения программы, в том числе когда используется многопоточная модель параллелизма или модель параллелизма с использованием MPI. При запуске вычислительной программы пользователь обычно указывает, сколько ядер, памяти и какое время требуется для исполнения программы. Все эти данные аккумулируются на специальном управляющем узле кластера, который не входит в расчетное поле, и на котором работает процесс-планировщик. Этот процесс управляет прохождением всех задач от всех пользователей, составляя расписание выполнения и контролируя очереди задач, если в данный момент недостаточно ресурсов для их исполнения.

Использование вычислительных ресурсов кластера обычно предполагает фоновый режим работы. Пользователь готовит задание, ставит его в очередь планировщика, получает от планировщика уникальный номер задания и далее просто ожидает либо автоматического извещения об окончании расчетов (с ошибкой или без), либо в ручном режиме запрашивает у планировщика состояние своей задачи.

Для работы моделирующего комплекса требуется другая схема использования кластера, поскольку необходимо выполнение следующих условий:

- кластер подключен к локальной сети, в которой размещены супервизор с базой данных расчетов;

- узлы кластера имеют прямой доступ в локальную сеть, а значит, прямой доступ в базу данных и к супервизору.

Эти требования ограничивают круг кластеров, на которых можно было бы использовать данную систему, так как обычно узлы вычислительного поля не имеют доступа за пределы общего интерконнекта кластера. Вместе с тем планируемые преимущества эффективного использования гибридного многопроцессорного кластера в предлагаемом ниже режиме его работы оправдывают требуемые изменения в схеме сетевых соединений кластера.

Общая схема интеграции вычислительных кластеров в систему моделирования (рис. 5) предполагает следующий порядок работы. Пользователь заходит на кластер стандартным образом (через узел доступа) и запрашивает ресурсы для запуска заданного количества диспетчеров.

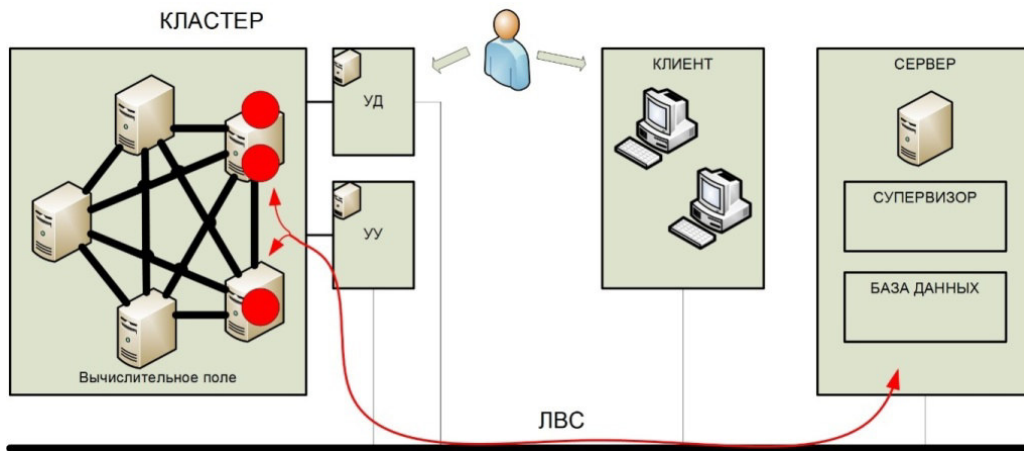


Рис. 5. Схема интеграции вычислительного кластера в ЛВС, где УУ – управляющий узел, УД – узел диспетчера.

Стандартный запрос ресурсов большинства существующих кластеров составляет: количество ядер, объем оперативной памяти, время счета. Если свободных ресурсов кластера в настоящий момент нет, то данные задания попадают в очередь и ожидают возможности запуститься. Если свободные ресурсы имеются, то запускается заданное количество диспетчеров. На рис. 5 запущенный диспетчер показан закрашенным кругом. Под каждый диспетчер резервируется определенное количество ресурсов – ядер и памяти. Каждый диспетчер инициализируется и переходит в цикл ожидания запросов от супервизора. После этого пользователь может перейти к процессу моделирования и передачи соответствующих команд супервизора. В случае, когда для диспетчера заканчивается ресурс времени, пользователь может тем же путем зайти на кластер и запросить ресурсы на запуск дополнительных диспетчеров. Этот процесс может продолжаться до тех пор, пока есть потребность в вычислительных ресурсах для проведения модельных расчетов. Также следует заметить, что такой подход позволяет задействовать ресурсы других вычислительных кластеров, без ограничения их числа. Ограничением является лишь ограниченная способность супервизора обрабатывать общее количество соединений от диспетчеров.

Каждый диспетчер работает по следующей схеме. На рис. 6 показана диаграмма взаимодействия супервизора, диспетчера и базы данных. На первом этапе после запуска диспетчер осуществляет процедуру своей регистрации у супервизора, сообщая ему свои сетевые параметры для последующей передачи команд моделирования и объем имеющихся вычислительных ресурсов, которые были оп-

ределены пользователем при запуске. После регистрации диспетчер переходит в цикл обработки команд моделирования от супервизора. По каждому запросу диспетчер осуществляет запуск модуля, который проводит расчеты и записывает результаты расчетов через интерфейс доступа к базе данных. После завершения модуля диспетчер посылает команду завершения и переходит к ожиданию следующих команд от супервизора. По завершении выделенного на кластере времени для работы диспетчера системой управления посылается сигнал об приближающемся окончании времени. В обработчике данного сигнала диспетчер посылает супервизору сообщение о прекращении регистрации. После чего завершает свое выполнение.

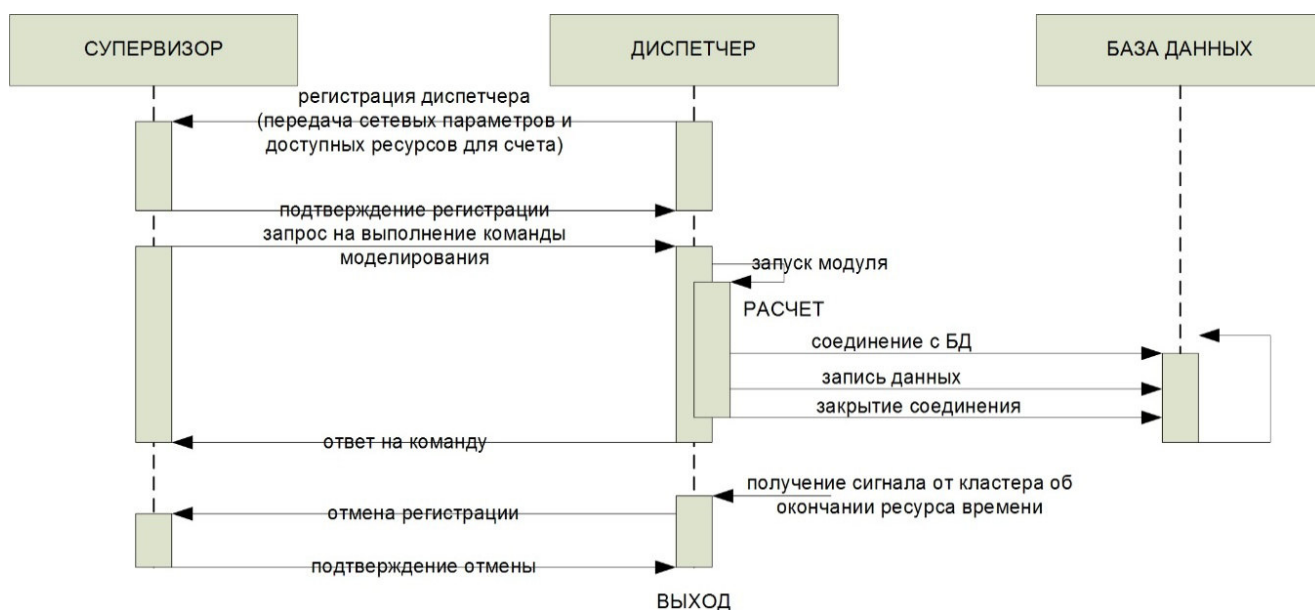


Рис. 6. Диаграмма взаимодействия диспетчер-супервизор.

Мировая практика использования кластеров в таком режиме указывает на большое число создаваемых средств моделирования в молекулярной динамике, компьютерном моделировании инженерных объектов и конструкций и в ряде других областей. Ресурсы кластера подключаются в режиме «по требованию» тогда, когда пользователь подготовил вычислительную сетку для расчетов. Визуализация и обработка результатов выполняются непосредственно на рабочем месте исследователя.

Заключение

Предложенная в работе схема включения гибридного вычислительного кластера в состав разработанного авторами распределенного моделирующего комплекса позволит пользователям в полной мере использовать вычислительный потенциал кластера, реализующего параллельные и распределенные вычисления. Для полномасштабной оценки эффективности предложенного решения по инте-

грации вычислительного кластера в локальную вычислительную сеть планируются разноплановые эксперименты по моделированию рабочих миссий АПР.

ЛИТЕРАТУРА

1. Wittmeier S., Jantsch M., Dalamagkidis K., et. CALIPER: AUniversal Robot Simulation Framework for Tendon-Driven Robots // International Conference on Intelligent Robots and Systems. September 25-30 – San Francisco, CA, USA, 2011 – P.1063-1068.
2. Open Robotics Automation Virtual Environment (OpenRAVE). URL:<http://www.openrave.programmingvision.com> (дата обращения: 18.07.17).
3. Gazebo simulations. URL:<http://gazebosim.org> (дата обращения: 18.07.17).
4. Boyer H., Agüero C.E., Gerkey B. CloudSim: Robotics Simulation in the Cloud // NRI Workshop on Cloud Robotics: Challenges and Opportunities – March 2013.
5. Parodi O., Lapierre L., Jouvencel B. Hardware-in-The-Loop Simulators for multi-vehicles scenarios: survey on existing solutions and proposal of a new architecture // In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009. – St. Louis, MO, USA.
6. Sehgal A., Cernea D. A multi-AUV missions simulation framework for the USARSim Robotics Simulator // Proceedings of 18th Mediterranean Conference on Control & Automation (MED). – 23-25 June 2010. – P.1188-1193.
7. Бобков В.А., Морозов М.А., Багницкий А.В. и др. Имитационный моделирующий комплекс для обследовательского автономного подводного робота // Научная визуализация. – 2013. – Т. 5, № 4. – С.47-70.
8. Bobkov V.A., Borisov Yu.S., Inzartsev A.V., Mel'man S.V Simulation program complex for studying motion control methods for autonomous underwater vehicles // Programming and Computer Software, 2008. – Vol. 34, №5. – P.257-266.
9. Бобков В.А, Морозов М.А. Моделирование работы гидролокационных устройств средствами машинной графики // Подводные исследования и робототехника. – 2012. – №1(13) – С.47-51.
10. Мельман С.В., Бобков В.А., Инзарцев А.В. и др. Программный моделирующий комплекс для автономных подводных аппаратов на базе многопроцессорной архитектуры // Подводные исследования и робототехника. – 2015. – №1(19). – С.23-32.
11. Buades A., Coll B., Morel J.M. A review of image denoising algorithms, with a new one // SIAM Multiscale Modeling and Simulation. – Vol. 4. – 2005. – P. 490-530.
12. Злыгостев А.С. Освещенность и фотосъемка под водой при естественном свете. URL: <http://istoriya-foto.ru/books/item/f00/s00/z0000006/st013.shtml> (дата обращения: 18.07.17).

Статья представлена к публикации членом редколлегии Н.В. Кинштом.

E-mail:

Бобков Валерий Александрович – bobkov@iacp.dvo.ru;

Мельман Сергей Владимирович – melman@dvo.ru;

Морозов Михаил Александрович – torozov@iacp.dvo.ru;

Тарасов Георгий Витальевич – george@dvo.ru.