

¹ АО «МЦСТ», г. Москва, Россия² Институт электронных управляющих машин им И.С. Брука, г. Москва, Россия³ Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия**РЕШЕНИЕ ЗАДАЧ КОГНИТИВНОГО УПРАВЛЕНИЯ ГРУППОЙ РОБОТОВ НА МНОГОЯДЕРНЫХ МИКРОПРОЦЕССОРАХ «ЭЛЬБРУС»*****Аннотация**

При разработке современных автономных наземных робототехнических комплексов одной из главных задач является создание когнитивной интеллектуальной системы управления. При этом, актуальным является создание роботов, работающих как по одиночке, так и в составе группы. Существенным, но не решенным вопросом является оснащение таких систем управления вычислительной техникой, разработанной на базе отечественных микропроцессоров и программным обеспечением отечественной разработки. Цель работы – исследовать применимость вычислительных средств на базе многоядерных микропроцессоров «Эльбрус» для решения задач когнитивного управления в таких интеллектуальных системах. Авторами были исследованы задачи управления группой роботов и задачи работы с внешними датчиками, как наиболее сложные и ресурсоемкие задачи при создании таких систем. Авторами были исследованы алгоритмы поиска пути на графе, распознавания и работы системы стереозрения и разработаны программные средства, моделирующие данные алгоритмы с учетом особенностей микропроцессоров «Эльбрус». Получены временные характеристики для алгоритмов построения карты проходимости, поиска пути на графе, стереорекострукции и калибровки. Показано, что вычислительные комплексы на базе микропроцессоров «Эльбрус» могут удовлетворять требованиям, выдвигаемым к когнитивным системам управления современными робототехническими комплексами.

Ключевые слова

Когнитивное управление; многоядерные вычислительные системы; моделирование алгоритмов группового управления.

Bocharov N.A.¹, Paramonov N.B.², Alexandrov A.V.², Slavin O.A.³¹ JSC «MCST», Moscow, Russia² The institute of electronic control computers, Moscow, Russia³ Federal Research Center Computer Science and Control of the Russian Academy of Sciences, Moscow, Russia**SOLVING OF TASKS OF COGNITIVE CONTROL A ROBOTS GROUP IN MULTI-CORE MICROPROCESSORS «ELBRUS»****Abstract**

One of the main part in development of the modern autonomous robotic complexes is creation of the cognitive intelligent control system. Also, there is an actual task to create such robots, that could operate both individually and in groups. A significant but unresolved issue is equipping such systems by computing hardware, developed on the basis of domestic microprocessors and software, developed by domestic companies. The purpose of this work is researching the applicability of computer systems based on multicore «Elbrus» microprocessors in such intelligent systems for solving tasks of cognitive control. The authors have researched problem of controlling a group of robots and task of working with external sensors, as the most complex and expensive tasks in the creation of such systems. The authors have researched algorithms of the pathfinding on the graph, recognition and

* Труды II Международной научной конференции «Конвергентные когнитивно-информационные технологии» (Convergent'2017), Москва, 24-26 ноября, 2017

Proceedings of the II International scientific conference "Convergent cognitive information technologies" (Convergent'2017), Moscow, Russia, November 24-26, 2017

stereovision and have developed software that simulate these algorithms with the features of microprocessors "Elbrus". The temporal characteristics for building a passability map, pathfinding on the graph, stereo reconstruction and calibration algorithms were obtained. Obtained results display possibility of using computer systems of «Elbrus» series as computer for modern cognitive intelligent control system.

Keywords

Motion modelling; search path on the graph; modelling algorithms of group control.

Введение

Современной тенденцией развития наземной робототехники является постепенный переход от дистанционно управляемых к полуавтономным, а в перспективе – к автономным робототехническим комплексам (РТК). При этом важным направлением является применение не единичных образцов, а группировок роботов, способных более эффективно выполнять определенные задачи. В связи с этим актуальным является создание интеллектуальных систем управления группировками автономных наземных робототехнических комплексов. Существенным, но не решенным вопросом создания систем управления РТК является оснащение вычислительной техникой, разработанной на базе отечественных микропроцессоров и программного обеспечения отечественной разработки.

Поскольку робототехника является одним из перспективных направлений применения вычислительных комплексов (ВК) и программного обеспечения семейства «Эльбрус» [1], то целью данной работы было исследовать и показать применимость микропроцессора «Эльбрус-8С» [2] для решения задач когнитивного управления группой роботов.

Результатом данной работы является программный комплекс, позволяющий моделировать поведение группы автономных роботов на местности и моделировать работу группы средств наблюдения за роботами. Также были разработаны вспомогательные программы для моделирования систем технического зрения роботов и средств резервирования. Данный программный комплекс позволил определить временные и нагрузочные характеристики, которые позволили сделать вывод о применимости микропроцессора «Эльбрус-8С» для решения задач когнитивного управления группой роботов, а также помог проводить работы по оптимизации программного обеспечения в ОС «Эльбрус».

Постановка задачи

В работе [3] авторами было проведено моделирование алгоритмов поиска пути и работы с системой стереозрения отдельным роботом на прошлом поколении процессоров «Эльбрус». Для тестирования был использован вычислительный комплекс, состоящий из четырех процессоров «Эльбрус-4С», объединенных в кластер и использующих общую NUMA-память.

Движение робота в модели было сведено к поиску пути на графе, соответственно был проведен анализ существующих алгоритмов поиска пути на графе. В качестве используемого алгоритма был выбран алгоритм A^* с некоторыми изменениями в эвристической функции для учета дополнительных параметров. Для повышения реалистичности модели в нее был включен учет таких параметров робота, как: скорость, ускорение, радиус поворота, радиус обнаружения препятствий. В работе [4] было проведено улучшение алгоритмов для моделирования уже не одного, а группы из нескольких роботов. В данной работе были проведены оптимизации алгоритмов и проведено моделирование на новом поколении микропроцессоров «Эльбрус-8С»

Формально, в рассматриваемой задаче поиска пути группой роботов обрабатываются три множества: множество роботов, множество целей и множество внешних факторов. В разработанной программе было реализовано моделирование всех этих множеств.

Задача поиска пути

Задача поиска пути была сведена к задаче поиска пути на графе из узла-старта до узла-финиша [5, 6]. Пара $(V(G), E(G))$ называется графом, если $V(G)$ — непустое конечное множество элементов, называемых узлами, а $E(G)$ — конечное множество неупорядоченных пар различных элементов из $V(G)$, называемых рёбрами. При этом поиск необходимо осуществлять, учитывая проходимость различных опорных поверхностей и радиус поворота робота.

Для задачи поиска пути роботом также необходимо учитывать длину пути, что можно обозначить через вес рёбер. При этом, если робот может переместиться из точки A в точку B , то не обязательно, что он может и переместиться из точки B в точку A . Например, если точка A — это точка на открытой местности, а точка B — точка у стены с азимутом, направленным прямо в стену. Таким образом, для поставленной задачи подходит использование взвешенного ориентированного графа.

Узел графа по определению представляет собой элемент графа, обозначающий объект любой природы. В данной задаче узел графа обозначает область на местности, в которую робот имеет возможность встать. Так как необходимо учитывать радиус поворота робота, то необходимо и учитывать азимут робота при построении пути. Таким образом, узел графа задается не только координатами x, y , но и азимутом.

Ребрам графа обозначается траектория между узлами, которые она соединяет, по которой может перемещаться робот. При этом траектория строится с учетом радиуса поворота робота. Так как при поиске пути надо учитывать проходимость различных опорных поверхностей, то вес ребра обозначим как средний коэффициент проходимости на всём протяжении траектории данного ребра.

Путь робота проходит от узла к узлу. Но эти узлы не могут быть соединены ребрами, являющимися прямыми траекториями, так как робот не имеет возможность мгновенно развернуться в нужном направлении. Поэтому между узлами необходимо находить траекторию с плавными поворотами, соответствующими радиусу поворота робота. При этом необходимо проверять эти траектории на наличие препятствий на пути.

Учет радиуса поворота происходит при построении траектории от узла к узлу. Соответственно, каждый узел, помимо координат, характеризовался еще и азимутом. Итоговая траектория между двумя узлами состоит из трех сегментов – двух дуг и отрезка прямой. При этом могут возникнуть два случая, различающиеся подсчетом основных точек: когда обе окружности обходятся в одном направлении и наоборот. Выбирается та, которая имеет наименьшую длину. При построении пути есть два случая, различающихся расчетом траектории: когда обе окружности поворота обходятся в одном направлении (например, обе по часовой стрелке) и когда две окружности поворота обходятся в противоположных направлениях (одна окружность по часовой стрелке, другая — против часовой стрелки).

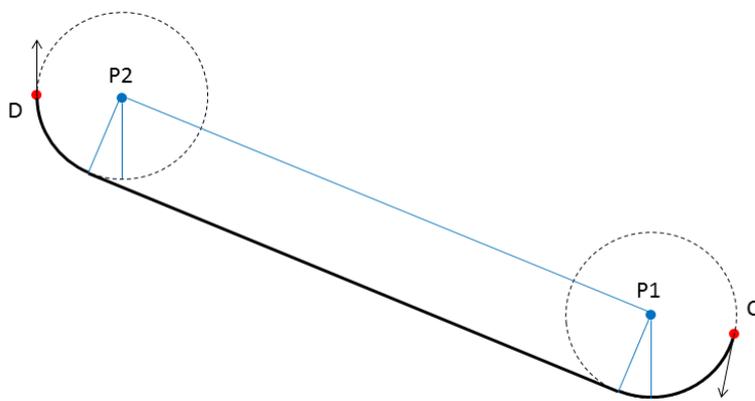


Рисунок 1. Обход окружностей в одном направлении.

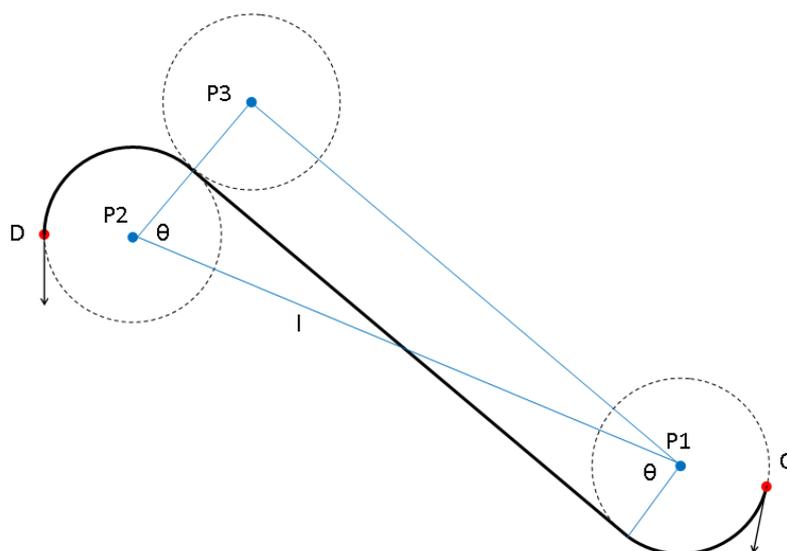


Рисунок 2. Обход окружностей в противоположных направлениях.

В обоих случаях задача построения траектории сводится к поиску угла первой окружности, на котором начинается движение по прямому участку траектории, угла второй окружности, на котором заканчивается движение по прямому участку, и длину этого прямого участка. Варианты обхода окружностей изображены на рисунках 1 и 2.

Задача расчета проходимости

Задача поиска наилучшего пути могла бы сводиться к задаче поиска кратчайшего пути. Но в реальности данное решение не всегда приемлемо. Путь может проходить по различным опорным поверхностям. Например, есть более длинный путь по асфальтированной дороге и более короткий по песку. Заранее известно, что робот движется по песку намного медленнее, чем по асфальту и поэтому лучший путь был бы более длинным.

Робот оценивает проходимость двумя методами: посредством анализа загружаемой в него карты и посредством своих датчиков.

Первый метод формирует карту проходимости на основе загружаемой в робота карты местности. Карта местности отображает виды опорных поверхностей, а карта проходимости отображает коэффициенты проходимости этих поверхностей.

Второй метод анализирует реальную проходимость и меняет карту проходимости, если реальный коэффициент отличается от коэффициента на карте.

Расчет карты проходимости

При расчете карты проходимости карта местности в формате OpenStreetMap разбивается на квадраты, соответствующие размерам робота. В каждом таком квадрате считается средний коэффициент проходимости по каждому пикселю. Если хотя бы один пиксель в этой области окажется непроходимым, тогда вся область считается непроходимой. Иначе, робот помечает эту область рассчитанным средним коэффициентом проходимости.

Но с растровыми картами могут возникнуть некоторые проблемы. Во-первых, между разными областями на карте в результате сглаживания, некоторые пиксели не соответствуют ни одному цвету из легенды. Во-вторых, в результате того же сглаживания, некоторые пиксели могут обозначать непроходимую область, хотя на самом деле это не так. И в-третьих, некоторые пиксели могут немного отличаться от цветов в легенде (обычно не больше чем на 2 в одном из каналов).

Первая проблема решается запоминанием значения веса последнего определенного пикселя. То есть, если цвет текущего пикселя не содержится в легенде, то это переходный цвет от одной области к другой и с большой вероятностью эта область имеет вес предыдущего проанализированного пикселя.

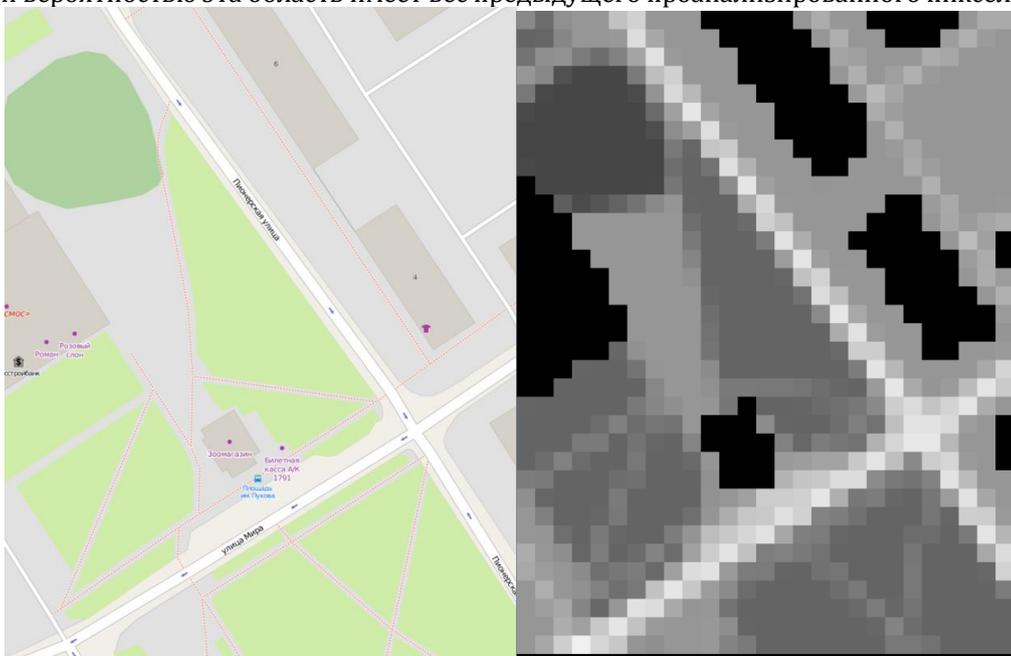


Рисунок 3. Пример карты местности и карты проходимости

Вторая проблема решается проверкой всех восьми пикселей вокруг обнаруженного непроходимого. По этим пикселям считается средний коэффициент проходимости, но если обнаружен хотя бы еще один

непроходимый пиксель, то этот коэффициент становится равным 0. Затем рассчитанный по восьми пикселям коэффициент проходимости присваивается текущему пикселю.

Третья проблема решается сверкой с легендой, хранящейся в работе, с некоторой погрешностью.

Пример формирования карты местности и сформированной карты проходимости приведен на рисунке 3.

Алгоритм поиска пути

Алгоритмы поиска пути ищут путь на графе из стартового узла в узел-финиш. При этом, в зависимости от алгоритма, путь может быть кратчайшим. Кроме того, некоторые алгоритмы позволяют учитывать вес узлов.

Алгоритм A* считается одним из лучших алгоритмов поиска пути [6]. Он объединяет в себя достоинства двух алгоритмов: учет длины пути из алгоритма Дейкстры и учет эвристической функции из алгоритма «лучший первый».

Алгоритм A* [7] использует формулу эвристики, которой в общем случае имеет вид:

$$f(n)=g(n)+h(n),$$

где $f(n)$ — значение оценки для узла n , $g(n)$ — стоимость пути из узла-старта в узел n , $h(n)$ — эвристическое приближение стоимости пути из узла n в узел-финиш.

Функция $h(n)$ должна быть допустимой эвристической оценкой, то есть не должна переоценивать расстояние до узла-финиша. Одним из способов задания такой функции является длина прямой, соединяющий узел n и узел-финиш.

Алгоритм работает аналогично алгоритму Дейкстры, где вместо длины пути учитывается функция $f(n)$, а когда узлу $n1$ устанавливается родитель $n2$, пересчитывается функция $g(n)$ следующим образом:

$$g(n1) = g(n2) + d(n1, n2),$$

где $d(n1, n2)$ — расстояние между узлами $n1$ и $n2$.

Так как подразумевается использование различных опорных поверхностей с разными коэффициентами проходимости и учет радиуса поворота, то функции $g(n)$ и $h(n)$ были изменены.

Функция $g(n)$ должна учитывать не только длину ребра, но и его вес. При этом желательно внести штраф к поворотам. При указании узлу $n1$ родителя $n2$ происходит пересчет функции $g(n)$ следующим образом:

$$g(n)=g(n2)+d(n1, n2)*w(n1, n2)+r(n1, n2)*w(n1, n2),$$

где $d(n1, n2)$ — длина пути от узла $n1$ до узла $n2$, $w(n1, n2)$ — вес ребра, соединяющего узлы $n1$ и $n2$, $r(n1, n2)$ — суммарный угол поворота в радианах ребра, соединяющего узлы $n1$ и $n2$.

Оценка $h(n)$ рассчитывается как длина пути от узла n до узла финиша с учетом направлений узла n и узла финиша, умноженная на средний вес. Средний вес в данном случае рассчитывается как среднее значение веса в точках узла n и узла финиша.

Алгоритм движения робота от узла к узлу

На вход данному алгоритму подается связь, по которой робот должен проехать от текущего узла к следующему узлу. При выполнении движения происходит цикл до тех пор, пока связь, которую робот должен проехать, не будет пройдена. Цикл также может прерываться замеченными изменениями карты.

В каждой итерации цикла робот проверяет свои сенсоры. Если при этой проверке было замечено сильное изменение какого-либо квадрата, то движение прерывается. Иначе, робот изменяет свою скорость, следуя одному из следующих вариантов:

- если сенсорами была замечена плохо проходимая область, то происходит торможение до минимальной скорости;
- если робот достиг момента торможения до финиша, то происходит торможение до нуля;
- если роботу была дана команда остановки, то происходит торможение до нуля;
- если ничего из вышеперечисленного не выполнено, то происходит ускорение до обычной скорости.

После того, как робот проедет некоторое расстояние, он проверяет условие начала торможения. Если скорость не стала равна нулю вследствие остановки, то робот продолжает движение.

Алгоритм движения робота по всему пути

Перед началом движения робот создает узел старта и узел финиша. После этого происходит поиск пути с помощью алгоритма A*. Если путь не был найден, то происходит плавная остановка с объездом препятствий. В противном случае происходит движение до каждого узла из найденного пути по алгоритму движения робота от узла к узлу. При этом, если было замечено изменение карты, то движение прерывается.

При выходе из цикла проверяется, достиг ли робот финиша. Если нет, то считается, что из цикла прохода по всему пути робот вышел из-за изменения карты. Тогда под роботом создается узел и пересчитывается путь. Иначе движение алгоритм заканчивает свою работу.

После движения робота на карте проходимости часто происходят изменения, но изменения весов в связях графа происходят только, когда алгоритм поиска пути их проанализирует. Таким образом, рекомендуется после окончания движения робота пересчитывать веса всех связей в графе, иначе алгоритм поиска пути не будет находить лучший путь.

Блок-схема алгоритма движения робота по всему пути приведена на рисунке 4.

Поиск пути группой роботов

Для разрабатываемой модели было принято использовать схему независимого управления с использованием общих ресурсов. Каждый робот движется независимо от других в свою целевую точку, считая остальных роботов препятствиями, которые надо объезжать. Общение между роботами происходит через внешнюю управляющую машину, которая уведомляется обо всех обнаруженных несоответствиях карты местности и карты реальности. Далее, внешняя управляющая машина уведомляет всех роботов о необходимости изменении карты проходимости, вследствие чего все роботы имеют одинаковую и актуальную карту проходимости, позволяющую более точно планировать маршрут. Блок-схема алгоритма поведения робота при обнаружении другого робота перед собой приведена на рисунке 5. На рисунке 6 изображено окно программы с группой из трех роботов.

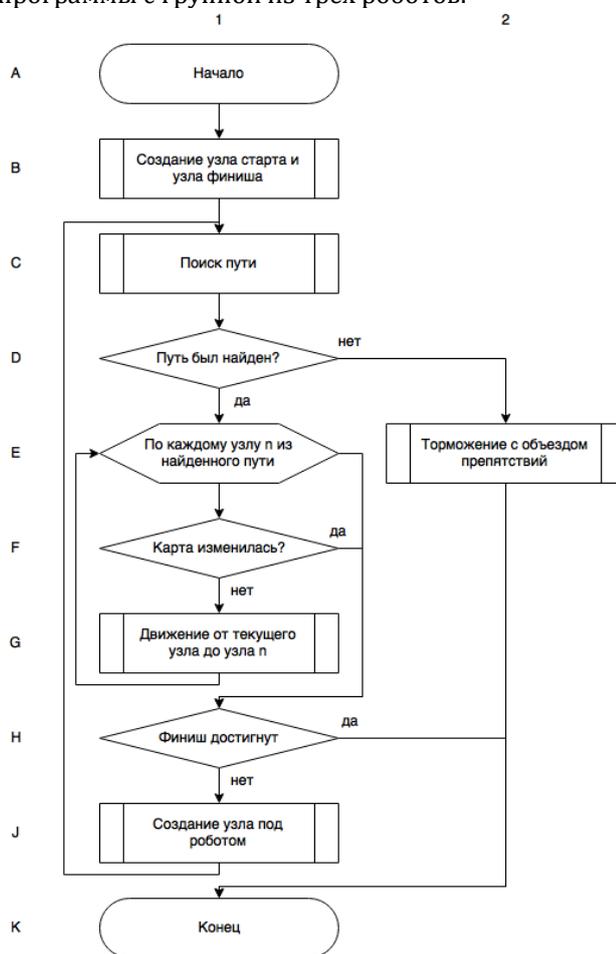


Рисунок 4 — Схема алгоритма движения робота по всему пути

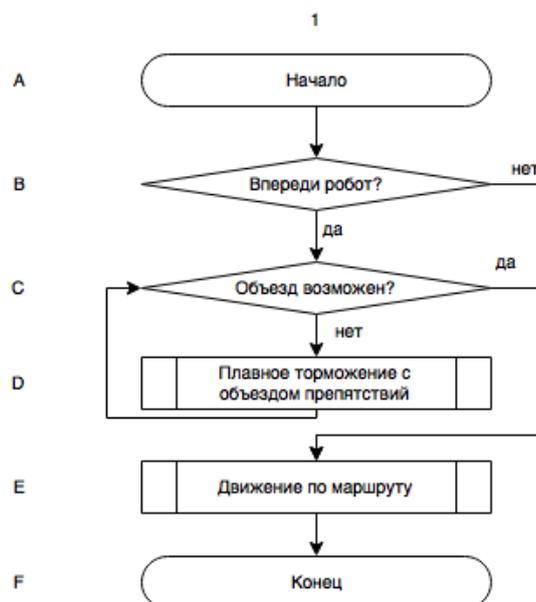


Рисунок 5. Блок-схема алгоритма поведения робота при обнаружении другого робота

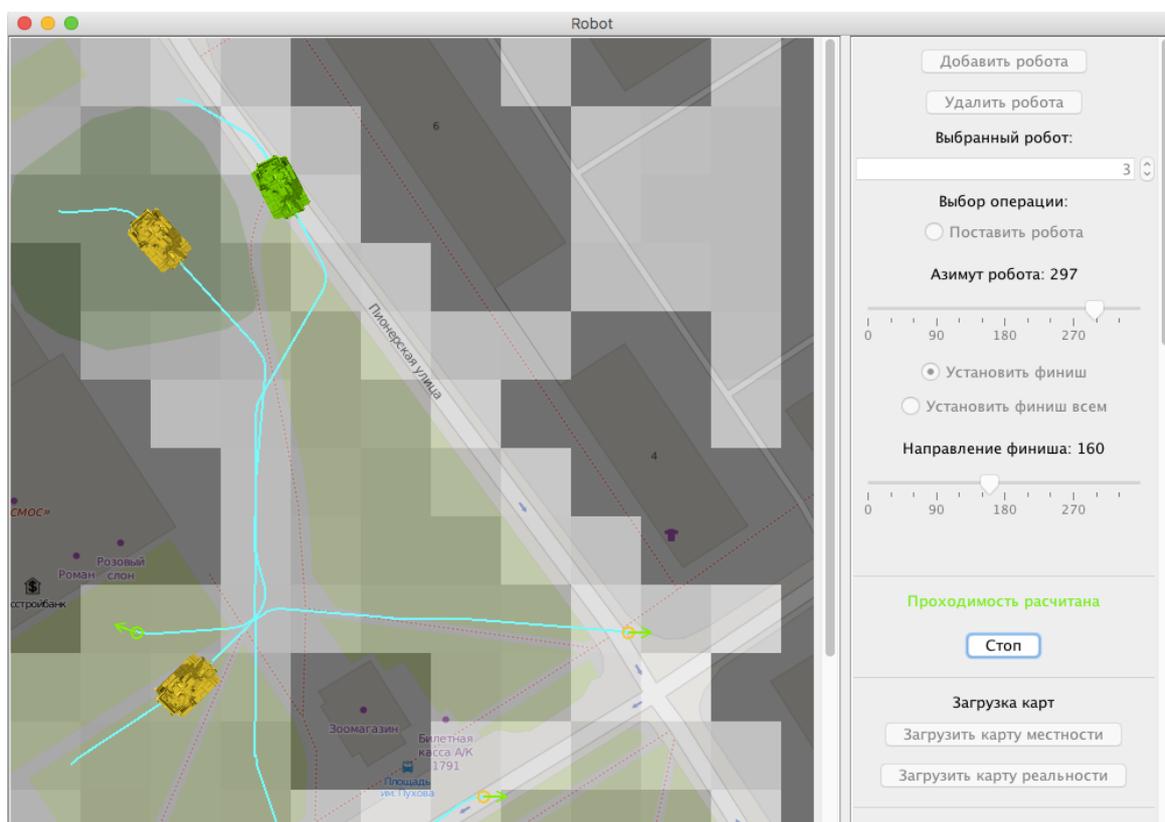


Рисунок 6. Окно программы с группой из трех роботов

Наблюдение за группой роботов

Наблюдение за роботами осуществляется группой камер с такими различными параметрами, как радиус обзора и угол обзора. В данной модели предполагается, что отдельная камера способна отследить в своей области видимости отдельных роботов и вести их от момента въезда в область видимости до момента выезда из области видимости. Если робот выезжает из области видимости на любое время большее минимального кванта, воспринимаемого программой, то считается, что камера его «потеряла» и следующий въезд будет восприниматься как въезд другим роботом. Далее алгоритм поиска соответствий ищет точки въезда и выезда и областей видимости, которые могли бы принадлежать одному роботу. При поиске соответствий помимо местоположения учитываются такие характеристики роботов, как:

скорость, ускорение и азимут. Если области видимости камер перекрываются, и робот проехал через это перекрытие хотя бы в одной точке, то соответствие между траекториями с камер определяется отдельно и однозначно. На рисунке 7 изображено окно программы с добавленными камерами, обнаруженными траекториями и некоторыми соответствиями.

Система стереозрения

Система стереоскопического зрения решает задачи обнаружения препятствий роботом на расстояниях свыше 30 метров. Применение данной системы возможно и на меньших расстояниях в случае неожиданного выхода из строя датчиков, использовавшихся на небольших расстояниях, например, сканирующих лазерных дальномеров. Эта система является одной из важнейших систем для навигации в автономном РТК. Для ее моделирования был разработан программный стенд с использованием открытой библиотеки OpenCV. Программный стенд реализует алгоритмы калибровки стереокамеры и алгоритма стереорекострукции.

В качестве способа калибровки был выбран путь смешанной калибровки. В основе данного подхода лежит использование простого плоского калибровочного шаблона в виде «шахматной доски». Шаблон снимается каждой камерой в некотором количестве положений, после чего по снимкам оцениваются параметры камеры. Данный подход не требует применения высокоточного шаблона и вспомогательного оборудования, при этом обеспечивает достаточную (до 1 мм) точность и пригоден для применения в полевых условиях.

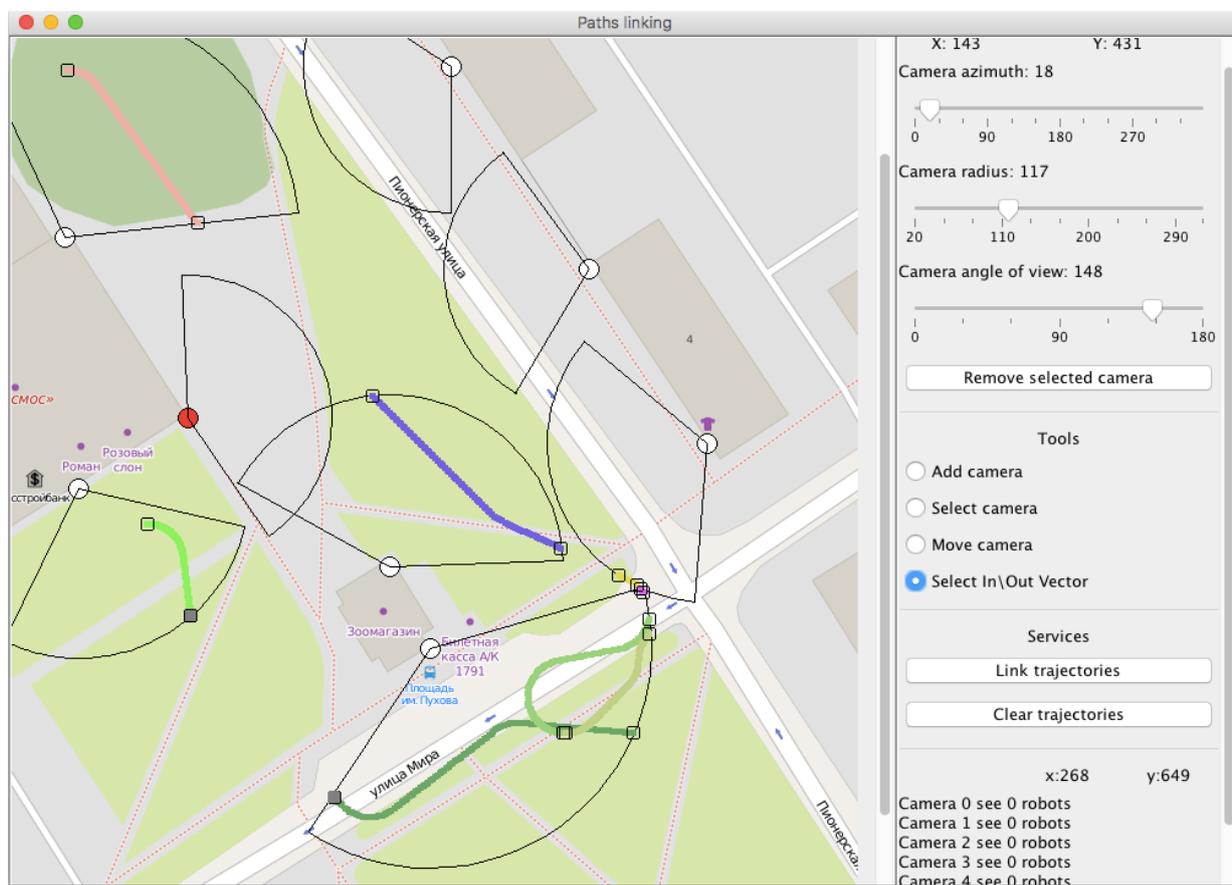


Рисунок 7. Окно программы наблюдения за группой роботов

В модели была реализована возможность установить параметры стереопары (расстояние между камерами, поворот), и провести калибровку с использованием виртуальной шахматной доски. В программную модель заложены несколько тестов, которые можно использовать для быстрой проверки работоспособности алгоритмов и оценки времени выполнения. Также в модель можно загружать сторонние изображения, в таком случае калибровка не требуется, на этапе стереорекострукции программа сама подберет подходящие параметры. На рисунке 8 представлено окно программы с полученной картой смещений для одного из встроенных тестов.

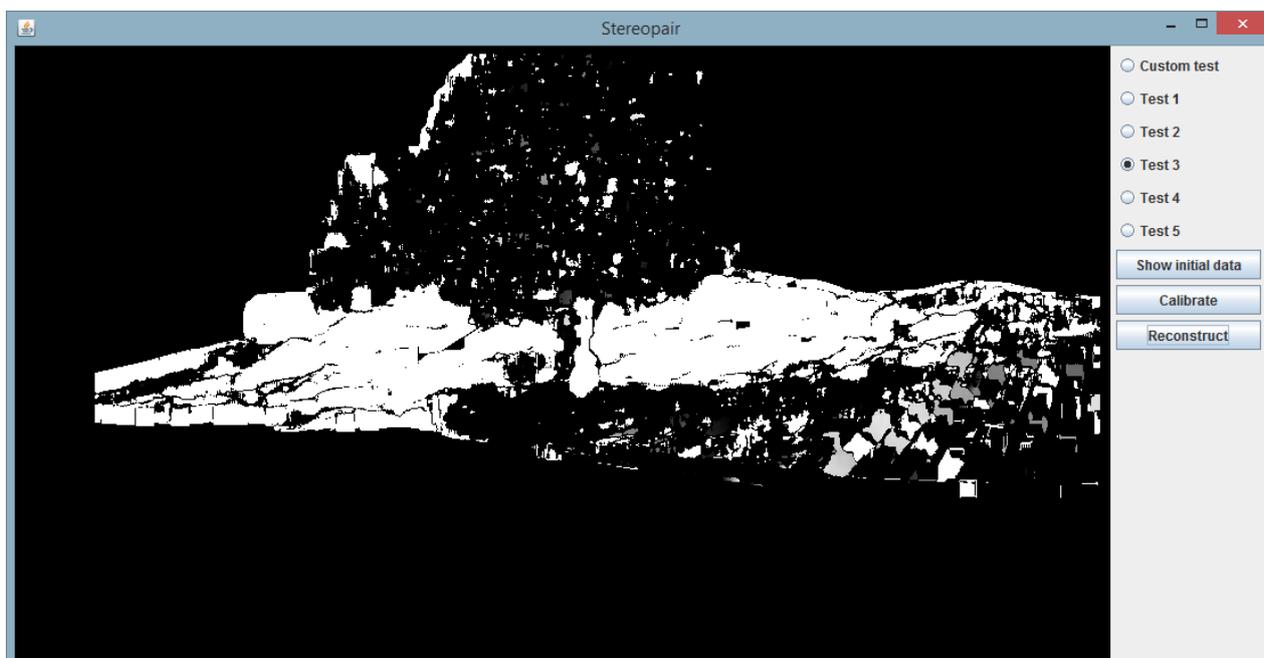


Рисунок 8. Результат трехмерной реконструкции для встроенного теста

Тестирование

Разработанный программный стенд позволяет проводить эксперименты для оценки скорости работы алгоритмов поиска пути как для одиночного робота, так и в составе группы, а также алгоритмов стереозрения. Было проведено сравнительное тестирование на микропроцессорах Intel и Эльбрус прошлого и текущего поколений. Для задачи поиска пути ключевые этапы — это построение графа проходимости и поиск пути на графе одним роботом. Для группы роботов время растет линейно. Результаты представлены на рисунках 9 и 10. Результаты тестирования алгоритма стереорекострукции представлены на рисунке 11. Для тестирования использовалась пара изображений размером 640x480 пикселей. Тестирование на архитектуре Intel проводилось на процессоре Intel Core-i7 4700. Тестирование на архитектуре Эльбрус проводилось на ВК «Эльбрус-401 РС» и ВК «Эльбрус-801». «Эльбрус-401 РС» представляет собой четырехпроцессорную систему с неравномерным доступом к памяти (NUMA). Каждый процессор имеет по 4 ядра с тактовой частотой 800 МГц. «Эльбрус-801» представляет собой вычислительный комплекс, содержащий один процессор Эльбрус-8С. Процессор имеет 8 ядер с тактовой частотой 1300 МГц.

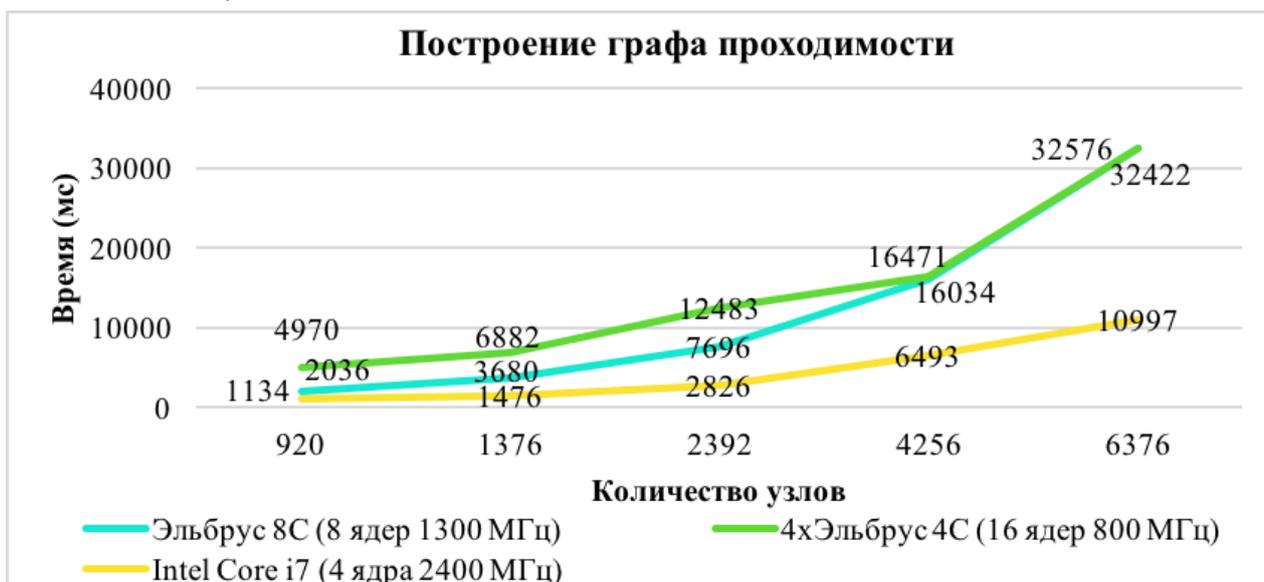


Рисунок 9. Временные характеристики алгоритма построения графа проходимости

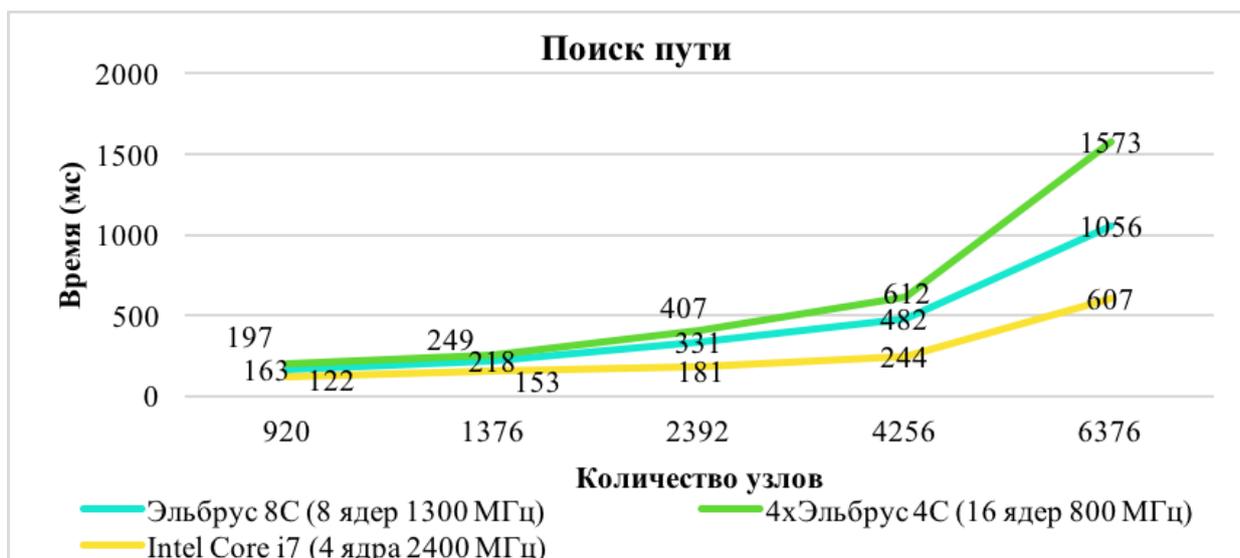


Рисунок 10. Временные характеристики алгоритма поиска пути

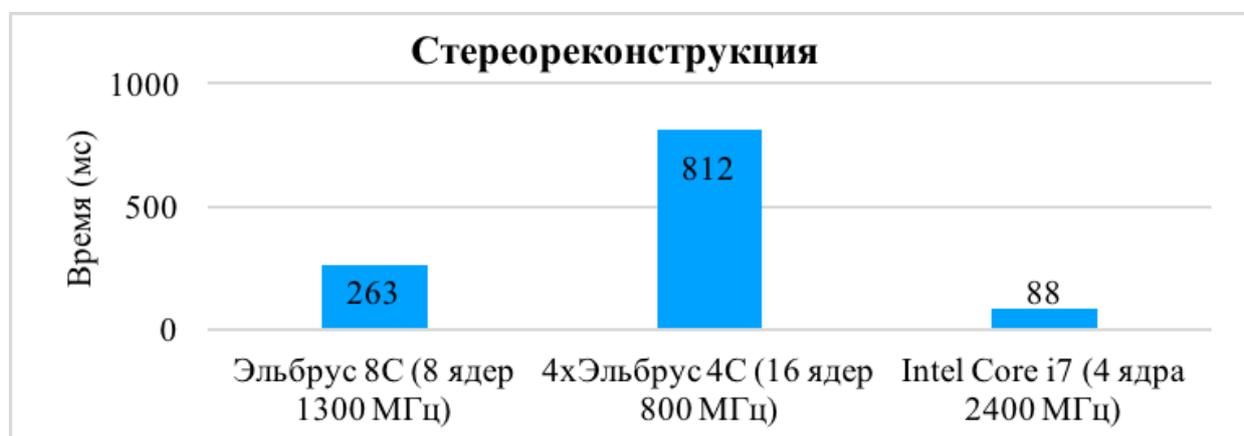


Рисунок 11. Затраты времени на стереореконструкцию по стереопаре

Заключение

В статье показано, что вычислительные средства на основе микропроцессора Эльбрус-8С могут удовлетворять требованиям, предъявляемым РТК в области планирования маршрута, систем стереозрения и наблюдения за группой роботов. Показаны временные характеристики для соответствующих алгоритмов.

Использование отечественных вычислительных средств и сертифицированного ОПО «Эльбрус» позволяет говорить о перспективах решения задач импортозамещения в области робототехники.

Благодарности

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект №17-29-03297)

Литература

1. Парамонов Н.Б., Ржевский Д.А., Перекатов В.И. Доверенная программно-аппаратная среда «Эльбрус» бортовых вычислительных средств робототехнических комплексов // Вопросы радиоэлектроники, сер. ЭВТ, 2015, вып. 1.
2. Д.М. Альфонсо, Р.В. Деменко, А.С. Кожин, Е.С. Кожин, Р.Е. Колычев, В.О. Костенко, Н.Ю. Поляков, Е.В. Смирнова, Д.А. Смирнов, П.А. Смольянов, В.В. Тихорский. Микроархитектура восьмиядерного универсального микропроцессора «Эльбрус-8С» // Вопросы радиоэлектроники. 2016. Т. 4. № 3. С. 6–13.
3. Бочаров Н.А., Сапачев И.Д., Парамонов Н.Б. Макеты задач робототехнических комплексов на языке Java в среде ОС «Эльбрус» // Москва: Наноиндустрия. Спецвыпуск 2017(74) Издательство «Техносфера» С. 122-127.
4. Бочаров Н.А., Парамонов Н.Б., Сапачев И.Д. Реализация алгоритмов группового управления на языке в среде ОС «Эльбрус» // Современные информационные технологии и ИТ-образование, 2016, Том 12, № 1, С. 108-115
5. В.Е., Таланов В.А. Графы. Модели вычислений. Структуры данных. — Нижний Новгород: Издательство Нижегородского гос. университета, 2005. — 307 с.
6. Берж К. Теория графов и ее применения / Под ред. И. А. Вайнштейна. — Москва: Издательство иностранной литературы, 1962. — 320 с.

7. Bryan Stout (оригинальная статья) Maxim Kamensky (перевод). Алгоритмы поиска пути [Электронный ресурс] // Программирование магических игр [Сайт] URL: <http://pmg.org.ru/ai/stout.htm> (дата обращения: 25.08.2017).

References

1. Paramonov N.B., Rzhetskij D.A., Perekatov V.I. Doverennaja programmno-apparatnaja sreda «Elbrus» bortovyh vychislitel'nyh sredstv robototekhnicheskikh kompleksov // Voprosy radioelektroniki, ser. JeVT, 2015, vyp. 1.
2. D.M. Al'fonso, R.V. Demenko, A.S. Kozhin, E.S. Kozhin, R.E. Kolychev, V.O. Kostenko, N.Ju. Poljakov, E.V. Smirnova, D.A. Smirnov, P.A. Smol'janov, V.V. Tihorskij. Mikroarhitektura vos'mjadernogo universal'nogo mikroprocessora «Elbrus -8C» // Voprosy radioelektroniki. 2016. T. 4. № 3. S. 6–13.
3. Bocharov N.A., Sapachev I.D., Paramonov N.B. Makety zadach robototekhnicheskikh kompleksov na jazyke Java v srede OS « Elbrus » // Moskva: Nanoindustrija . Specvypusk 2017(74) Izdatel'stvo «Tehnosfera» S. 122-127.
4. Bocharov N.A., Paramonov N.B., Sapachev I.D. Realizacija algoritmov gruppovogo upravlenija na jazyke v srede OS « Elbrus » // Sovremennye informacionnye tehnologii i IT-obrazovanie, 2016, Tom 12, № 1, С. 108-115
5. V.E., Talanov V.A. Grafy. Modeli vychislenij. Struktury dannyh. — Nizhnij Novgorod: Izdatel'stvo Nizhegorodskogo gos. universiteta, 2005. — 307 s.
6. Berzh K. Teorija grafov i ee primenenija / Pod red. I. A. Vajnshtejna. — Moskva: Izdatel'stvo inostrannoju literatury, 1962. — 320 s.
7. Bryan Stout (original'naja stat'ja) Maxim Kamensky (perevod). Algoritmy poiska puti [Jelektronnyj resurs] // Programirovanie magicheskikh igr [Sajt] URL: <http://pmg.org.ru/ai/stout.htm> (data obrashhenija: 25.08.2017).

Об авторах:

Бочаров Никита Алексеевич, магистр, выпускник кафедры информатики и вычислительной техники Московского физико-технического института, инженер-программист 1 категории, АО «МЦСТ», bocharov.na@phystech.edu

Парамонов Николай Борисович, доктор технических наук, профессор, главный научный сотрудник, АО «МЦСТ», paramonov.n.b@mail.ru

Александров Александр Владимирович, кандидат технических наук, начальник отдела, АО «МЦСТ», Alexander.V.Alexandrov@mcst.ru

Славин Олег Анатольевич, доктор технических наук, заведующий лабораторией Института системного анализа, Федеральный исследовательский центр «Информатика и управление» РАН, oslavin@isa.ru

Note on the authors:

Bocharov Nikita A., Master, Engineer-programmer of the 1st category, JSC "MCST", bocharov.na@phystech.edu

Paramonov Nikolay B., Doctor of Technical Sciences, Professor, Chief Researcher, JSC "MCST", paramonov.n.b@mail.ru

Alexandrov Alexander V., Candidate of technical sciences, head of the department, JSC "MCST", Alexander.V.Alexandrov@mcst.ru

Slavin Oleg A., Doctor of Technical Sciences, Head of the Laboratory, Institute for Systems Analysis, Federal Research Center Computer Science and Control of the Russian Academy of Sciences, oslavin@isa.ru