

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

РОБОТОТЕХНИКА И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ. ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ

Материалы научно-практической конференции
молодых ученых и студентов
25–26 октября 2012 года

Брест 2012

УДК 004.896

ББК 32.816

Редакционная коллегия:

В.С. Рубанов, БрГТУ, к.ф.-м.н., доцент – главный редактор

В.А. Головкин, БрГТУ, д.т.н., профессор – зам. главного редактора

В.М. Ракецкий, БрГТУ, к.ф.-м.н., доцент

Л.П. Махнист, БрГТУ, к.т.н., доцент

А.П. Дунец, БрГТУ, доцент

Рецензенты:

Муравьев Г.Л., профессор кафедры интеллектуальных информационных технологий БрГТУ, к.т.н.;

Матюшков Л.П., доцент кафедры экономики и управления БрГУ им. А.С. Пушкина, к.т.н.

Робототехника и искусственный интеллект. Проблемы и перспективы: материалы научно-практической конференции молодых ученых и студентов, Брест, 25–26 октября 2012 г. / Брестский государственный технический университет; под ред. Рубанова [и др.]. – Брест: изд-во БрГТУ, 2012. – 64 с.

ISBN 978-985-493-244-6

Представлены статьи по современным проблемам робототехники в научных и прикладных исследованиях, распознавания образов и анализа изображений, искусственного интеллекта и нейронных сетей. Представлены проекты Студенческой научно-исследовательской лаборатории «Робототехника».

УДК 004.896

ББК 32.816

ISBN 978-985-493-244-6

© Издательство БрГТУ, 2012

Кабыш А.С.

Научный руководитель: д.т.н., профессор Головки В.А.

МОДЕЛЬ КООРДИНАЦИИ ПОВЕДЕНИЯ АГЕНТОВ НА ОСНОВЕ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Резюме: В данной работе описывается модель для нахождения оптимального поведения многоагентной структуры через организацию в ней оптимальных взаимодействий между агентами. Модель включает две основные техники. Модель графов координации позволяет явно выразить зависимость между агентами, что позволяет разбить целевую функцию поведения в линейную сумму индивидуальных целевых функций. Модель оценки влияний позволяет оценить влияния других агентов на действия друг друга и в результате позволяет им координировать свои действия. В работе приведена реализация данной модели на основе обучения с подкреплением и экспериментальные результаты применения данной модели.

Ключевые слова: многоагентные системы, обучение с подкреплением, Q-Learning, обучение через влияние, граф координации

Введение

Огромный класс задач сводится к теории многоагентных систем, где рассматриваются способы взаимодействия между агентами. Типовые задачи, решаемые в теории многоагентных систем, включают в себя [1]: автоматизированный трейдинг, ведение переговоров, управление нагрузкой сети, организацию коллектива роботов-футболистов, распределенную оптимизацию, распределенное планирование задач [2] и другие.

Традиционно **многоагентная система** состоит из совокупности агентов, разработанных для кооперации друг с другом для достижения некоторой цели. **Агент** понимается как сущность, обладающая состоянием, способная воспринимать окружающую среду и выполнять в ней какие-то действия.

Свойства и поведение многоагентной системы зависят от суммарных свойств и поведения входящих в неё агентов. В зависимости от алгоритма коллективного поведения многоагентная система выдвигает различные требования к агентам, которые могут быть использованы в ней для данного алгоритма. Например большинство алгоритмов стайного поведения не предполагает наличия глобальной коммуникации между агентами, а алгоритмы ведения переговоров требуют, чтобы агенты были разными и по возможности держали в секрете свои модели.

Представленная в данной статье модель также предъявляет ряд требований к агентам. Агент рассматривается как **актор** [3] – активный агент, существующий параллельно (и одновременно) с другим таким же агентом, взаимодействующий с другими посредством стандартного протокола внутреннего взаимодействия в виде сообщений. Агенты должны обладать способностью к коммуникации, а многоагентная система – обеспечивать параллельность работы агентов.

Стоит отметить, что коммуникация подразумевает обмен сообщениями в каком-либо виде, а не восприятие другого агента посредством сенсоров. Агент должен иметь восприятие среды и восприятие сообщений, как нечто разделенное. Модель не специфицирует конкретный способ коммуникации между аген-

тами. Это может быть как прямой обмен сообщениями, так и использование среды, в качестве посредника для передачи сообщений, однако это повлияет на динамику работы алгоритма.

При обучении многоагентных систем либо изобретаются новые, гибридные алгоритмы, например в [4], либо адаптируются уже существующие. Во втором подходе разрабатывается какой-либо новый алгоритм, регулирующий способ совместной работы набора уже существующих алгоритмов, примененных к отдельным агентам.

Существуют 4 основные парадигмы адаптации алгоритмов обучения к многоагентным системам [5]:

- **Командное обучение** (*team learning*) развивает идею, что команду агентов можно обучать так, как если бы это был один агент. Этот подход не требует никаких изменений в алгоритмах обучения, но имеет ограничения в применимости. При увеличении количества агентов экспоненциально растет и размер пространства состояний/действий, в котором ведется поиск решения (проблема проклятья размерности).

- **Индивидуальное обучение** (*individual learning*) рассматривает применение индивидуального алгоритма обучения каждому агенту, игнорируя дополнительные данные от других агентов. Проблемой данного подхода является то, что совокупность оптимальных индивидуальных стратегий не обязательно представляет оптимальную командную стратегию.

- **Совместное обучение** (*joint action learning*) фокусируется на обучении оптимальным действиям в ответ на действия других агентов. Каждый агент обучается выполнять наилучшие действия в объединенном контексте действий других агентов. Этот подход позволяет построить оптимальную коллективную стратегию, но обладает тем же проклятьем размерности, что и командное обучение.

- **Обучение через влияние** (*influence-value learning*) основано на идее изменения поведения агента под влиянием мнения других агентов. Данный подход берет на вооружение множественные социальные факторы и использует их в качестве эвристик для конкретизации отношений внутри многоагентной системы.

В данной работе будет рассмотрена обобщенная, адаптивная модель организации взаимодействий внутри многоагентной системы на основе обучения через влияние для формирования в ней целенаправленного поведения посредством нахождения оптимальных взаимодействий между агентами. Под ожидаемым поведением многоагентной системы понимается совокупное поведение входящих в неё агентов, которое можно оценить как одно целое, приводящее к достижению некоторой цели. Примерами таких целей могут быть формирование и поддержание формации, коллективное управление нагрузкой или ресурсами, построение целостной карты окружающего пространства и др.

Модель обладает следующими свойствами:

- Данная модель является обобщенной, т.е. она не предъявляет никаких первоначальных требований к агентам, кроме способности получать и отправлять сообщения другим агентам.

- Модель является адаптивной в том, что она позволяет изменить структуру многоагентной системы и поведение входящих в неё агентов, если изменилась целевая функция поведения.

- Модель фокусируется на определении тех оптимальных взаимодействий между агентами, которые приводят к решению поставленной задачи, сохраняя аспекты индивидуального обучения.

1. Модель организации коллективного поведения

Ключевой концепцией является понятие **взаимодействия** или **влияния** между агентами. В реальном мире, при взаимодействии людей друг с другом, все действия отдельного индивида оцениваются, как и им самим, так и другими людьми в разрезе получаемого опыта. Примерами таких оценок являются ожидания, похвалы или наказания до или после совершенного действия.

В коллаборативной многоагентной системе все агенты могут потенциально влиять друг на друга. Под **влиянием** (*influence*) понимается оценка другими агентами, направленная на текущего агента. Другие агенты могут оценивать поведение агента и его регулировать с той целью, чтобы действия, выбранные индивидуальным агентом, соответствовали оптимальным решениям группы в целом. Любой агент, инициирующий влияние на другого агента, называется *отправителем*. Агент, принимающий влияние и как-то на него реагирующий, называется *получателем*.

Каждый агент i выбирает индивидуальное действие a_i из множества A_i , находясь в некотором состоянии. Значение влияния между агентом i и группой из N агентов относительно выбранного агентом действия определяется следующей формулой [5]:

$$I_i = \sum_{j=1, i \neq j}^N \beta_i(j) * Op_j(i), \quad (1)$$

где $\beta_i(j)$ – коэффициент влияния агента j на агента i ($0 \leq \beta \leq 1$), $Op_j(i)$ – коэффициент оценки агентом j действия агента i .

Коэффициент влияния β определяет, будет ли агент подвержен влиянию других агентов. Так, если β будет стремиться к нулю, то агент предпочтет действовать индивидуально. Op – это оценка другими агентами действия выполняемого текущим агентом.

Рисунок 1 показывает пример формирования влияния агентом j после выбора агентом i действия.

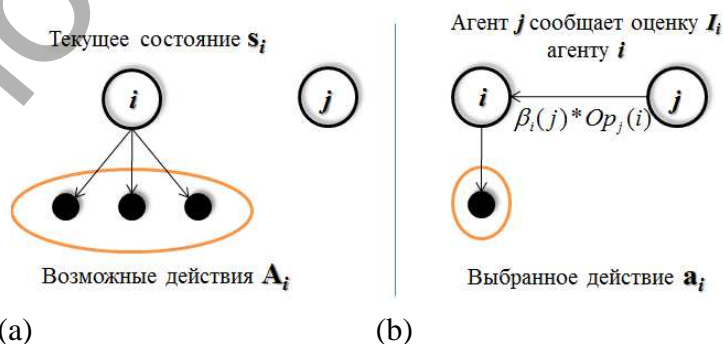


Рисунок 1 – Пример модели влияний. (а) – Агент i выбирает действие. Выбрав действие, агент j сообщил свою оценку i , сформировав влияние

Многоагентная система формирует результирующее объединенное действие (joint action) $a = \{a_1, a_2, \dots, a_N\}$. Проблема координации [6] поведения состоит в том, чтобы найти такое оптимальное объединенное действие a^* , которое мак-

симизирует полезность объединённого действия коллективного поведения системы $u(a)$, что $a^* = \arg \max_a u(a)$.

В большинстве реальных проблем действия одного агента зависят лишь от небольшого числа других агентов. Например, в задаче коллективного сбора ресурсов, агенты могут координировать свои действия только с соседями. Для явного отслеживания таких зависимостей предложена модель **графов координации** (coordination graphs, (CG)) [7].

Пусть действия агента i зависят только от некоторого подмножества других агентов, $j \in M(i)$. Тогда целевая функция многоагентной системы $u(a)$ может быть разбита на линейную комбинацию локальных целевых функций по следующей формуле:

$$u(a) = \sum_{i=1}^N f_i(sub_i), \quad (2)$$

где sub_i подмножество всего множества действий a , соответствующее действиям агентов $j \in M(i)$, от которых зависит агент i . Таким образом, глобальная проблема координации заменена на совокупность локальных проблем координации, каждая из которых включает малое подмножество агентов.

Зависимость между агентами может быть отражена в виде графа координации, где вершины соответствуют агентам, а ребра связывают зависимых агентов, которые обязаны координировать свои действия. По ребрам зависимостей агенты могут обмениваться сообщениями, а сами ребра специфицировать конкретный протокол общения между агентами. Пример графа координации изображен на рисунке 2.

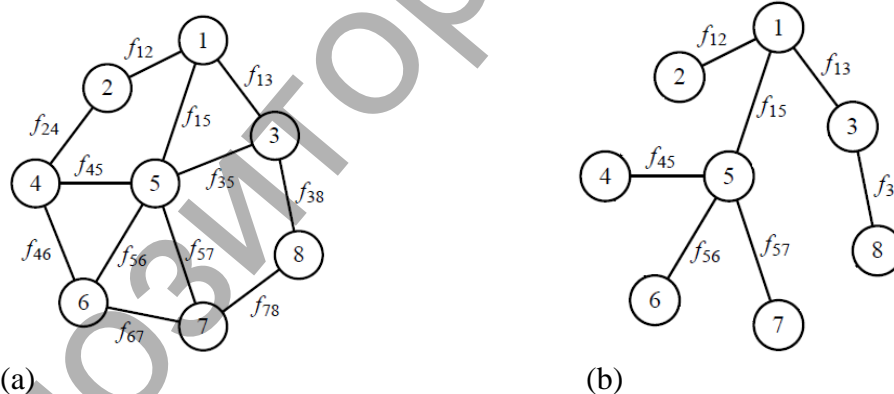


Рисунок 2 – Пример графа координации для восьми агентов до (a) и после (b) декомпозиции зависимостей; каждое ребро представляет зависимость координации

В данном разделе не были специфицированы следующие основные пункты:

- как происходит расчет коэффициента $Op_j(i)$ между агентами;
- как происходит перерасчет составляющих формулы (1) с течением времени;
- как агент изменяет свое поведение под воздействием влияний от других агентов;
- как агент оценивает влияние, оказываемое на него другими агентами;
- каков алгоритм нахождения оптимальных зависимостей в многоагентной системе;

Эти пункты специально были оставлены открытыми для максимальной обобщенности алгоритма под конкретную задачу и алгоритм.

Например, в работе [5] коэффициент $Op_j(i)$ увеличивался, если агент j выполнял действие с меньшей ценностью, чем агент i , и наоборот, уменьшался, если агент j выполнял более выгодное действие. Это отражает факт того, что люди думают хорошо о тех действиях, которые приносят другим больше прибыли. В следующем разделе представлена реализация данной модели на основе подкрепляющего обучения.

Сформулируем задачу организации многоагентной системы следующим образом:

- Пусть имеется N агентов, каждый из которых может выполнять некоторые действия (агенты могут быть как гетерогенными, так и гомогенными). Агенты имеют некоторое начальное состояние. Агенты объединены в многоагентную систему, обеспечивающую их параллельность и коммуникацию.

- Пусть имеется некоторая известная цель, достижение которой одним агентом либо невозможно, либо неэффективно по сравнению с многоагентным подходом. Агент или многоагентная система может узнать, достигнута цель или нет.

- Требуется определить такое поведение агентов, которое приводит к достижению цели оптимальным образом. В частных случаях данная задача уже может решаться различными способами [1], без коммуникации между агентами.

- С учетом коммуникации между агентами требуется определить оптимальные взаимодействия между ними, приводящие к коллективному решению задачи.

- Необходимо декомпозировать граф координации, при условии, что не имеется априорной информации о взаимодействиях между агентами, или она ограничена.

Решение поставленной задачи выполняется посредством двух процессов, которые могут выполняться как последовательно, так и параллельно, в зависимости от используемых алгоритмов:

- Определение оптимальной структуры графа координации многоагентной системы с целью декомпозиции пространства решений.

- Определение оптимальных взаимодействий между агентами порождает оптимальное поведение.

2. Обучение с подкреплением для нахождения оптимальной структуры взаимодействий

Особенностью поставленной задачи является динамическое определение оптимальных влияний между агентами и графа координации. Для достижения этой цели требуется итеративный процесс настройки структуры многоагентной системы. За время функционирования многоагентной системы можно оценить разные взаимодействия между агентами, приводящими к решению задачи, и усилить конструктивные и ослабить деструктивные. Если взаимодействие между агентами не является необходимым, то, следовательно, координация между ними может не выполняться и зависимость на графе координации может быть устранена.

В качестве итеративного процесса оценки и формирования структуры многоагентной системы в большинстве работ [1] выступает **обучение с подкреплением** (*Reinforcement Learning*, (RL)) [9].

Ключевой особенностью данного метода является то, что он является активным методом обучения, направленным на взаимодействие со средой и с другими агентами в случае многоагентной системы. В классической трактовке, обучение с подкреплением – это нахождение оптимального поведения агента методом проб и ошибок через его взаимодействие со средой. В данной работе, по отношению к многоагентной системе, обучение с подкреплением – это нахождение оптимальной структуры системы методом проб и ошибок через взаимодействия агентов.

Модель обучения с подкреплением формулируется следующим образом. Агент i выбирает действие $a_i(t)$ из множества A_i , находясь в состоянии $s_i(t)$. После выполнения действия агент переходит в следующее состояние $s_i(t+1)$ и получает из внешней среды числовое значение награды $r_i(t+1)$, являющееся оценкой выполненного действия и совершенного изменения состояния. В состоянии $s_i(t+1)$ агент выбирает и выполняет следующее действие $a_i(t+1)$. Разница между ценностью следующего состояния с учетом награды и ценностью текущего состояния называется **ошибкой временной разности** (temporal-difference error).

Последовательность изменений состояния агента изображена на рис. 3 в виде **графа переходов** (*transition graph*). Вершинами графа переходов являются состояния агента, а ребрами отмечаются выбранные действия и переходы между состояниями вместе с ассоциированной с переходом значением награды. Глобальное – вся среда, в которой действует агент – описывается в виде Марковского процесса принятия решений.

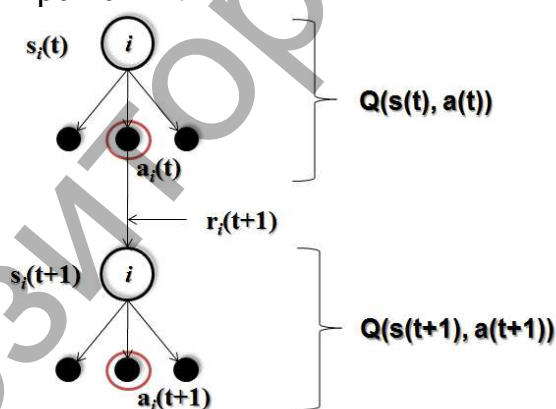


Рисунок 3 – Граф переходов агента.

Изображен переход агента из состояния $s(t)$ в состояние $s(t+1)$

С любой парой «состояние-действие» (s, a) ассоциировано значение ценности $Q(s, a)$, обозначающее полезность выполнения действия a в состоянии s . Значения $Q(s, a)$ рассчитываются Q – функцией и заранее неизвестны. Цель обучения – аппроксимировать истинные значения оценок Q – функции путем последовательного посещения пар (s, a) и получения ассоциированной с ними награды r , используя значение награды в качестве фактора, изменяющего ценность состояния. Способ отображения состояния s на ассоциированное с ним действие a называется политикой $\pi(s, a)$. Следовательно, задача обучения с подкреплением сводится к нахождению оптимальной политики, которая на ка-

ждом шаге выбирает оптимальные действия, ведущие к максимальной суммарной награде в будущем.

Самым популярным алгоритмом обучения с подкреплением является *Q-learning* [9], обновляющий значения функции ценности на каждой итерации по формуле (3):

$$Q(s(t), a(t)) = Q(s(t), a(t)) + \alpha [r(t+1) + \gamma \max_{a \in A} Q(s(t+1), a(t+1)) - Q(s(t), a(t))], \quad (1)$$

где в квадратных скобках [...] рассчитывается ошибка временной разности, α – шаг обучения ($0 < \alpha \leq 1$), а γ – коэффициент обесценивания ($0 < \gamma \leq 1$) отдаленных ценностей.

Из алгоритма видно, что последующие значения ценностей влияют на предыдущие значения. Следовательно, в момент времени $t(t+1)$ могут быть обновлены значения ценностей всех пар «состояние-действие», которые привели агента к текущему состоянию. Такая последовательность пар «состояние-действие» называется **следами преемственности** (*eligibility traces*). Модифицированная версия алгоритма называется *Watkins-Q(λ)* [9].

Поскольку на начальном этапе обучения неизвестно, какие взаимодействия между агентами являются оптимальными, а какие нет, действия выбираются случайно, методом проб и ошибок (фаза исследования). В конце обучения истинные значения ценности аппроксимированы, и агент использует их как руководство для оптимального поведения (фаза использования) во внешней среде в контексте других агентов.

3. Поиск оптимальных команд в многоагентной системе

В разделе 1 было введено понятие влияния как контекста оценок, создаваемого другими агентами по отношению целевому. Рассмотрим частный случай модели влияний, где влияние выступает в виде частного случая – активной команды.

В отличие от влияния, **команда** – это (1) активное сообщение от отправителя к клиенту, способное изменить его состояние, принятое или проигнорированное, а также (2) оценка этого сообщения. Если влияние принято, то оно может изменить текущее состояние агента или его последующие действия. Если влияние не принято, то оно возвращается с соответствующей пометкой и отправитель может отметить данное влияние как неконструктивное либо не влияющее на получателя.

Конструктивность команды – это характеристика, оценивающая ценность данного влияния по отношению к задаче, решаемой многоагентной системой. Конструктивные влияния имеют положительную оценку полезности, с некоторыми ограничениями, в рамках которых эта полезность сохраняется.

Получатель может проигнорировать влияние либо ввиду того, что отправитель не имеет достаточно веса, с точки зрения получателя, либо получатель выполняет собственные «эгоистичные» действия, ведущие его к цели, и нечувствителен к командам.

Рассмотрим пример двух агентов i и j . Оба агента в момент времени t находятся в некоторых состояниях и готовы выбрать действие для перехода в следующее состояние. Пусть агент i может повлиять на агента j , указав, какое действие ему выполнить. Посылаемую команду можно трактовать как выбор

агентом i некоторого действия $a_i(t)$, которое не переводит его в новое состояние $s_i(t+1)$. Агент j принимает команду в качестве указания и выполняет указанное действие, $a_j(t)$, переходя в новое состояние $s_j(t+1)$ и получая награду за выполнение действия $r_j(t+1)$.

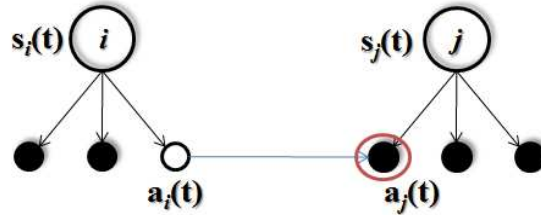


Рисунок 4 – Диаграмма переходов для агентов i и j .
Агент i иницирует команду над агентом j

После перехода агент j может скорректировать ценность перехода $Q(s_j(t), a_j(t))$ по формуле (3), а также выполнить оценку команды. В простейшем случае предполагаем, что коэффициент $\beta_i(j)$ равен 1. Значение $Op_j(i)$ содержит оценку агентом i действия агента j . Примером оценки являются качественные показатели следующего состояния агента j : награда $r_j(t+1)$ и обновленное значение ценности $Q'(s_j(t), a_j(t))$. Эти показатели передаются в качестве обратной связи агенту i , который может использовать их для обновления значения ценности команды.

Если обратная связь постоянно отрицательна, то ценность команды $Q(s_i(t), a_i(t))$ падает. Таким образом, выполняя разные команды над агентом j и получая обратную связь, агент i обучается оптимальному управлению над ним, формируя диапазон состояний, при котором сохраняется конструктивность команды.

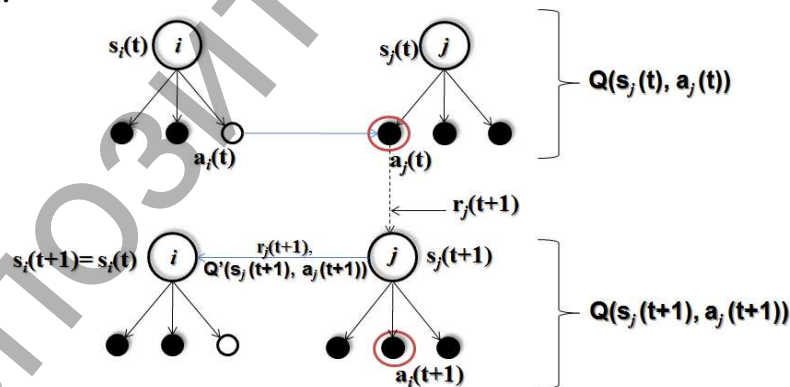


Рисунок 5 – Диаграмма переходов для агентов i и j . Продолжение. Агент j выполнил команду, совершил переход, обновил значение ценности и вернул обратную связь агенту i

Модифицированная формула для агента i , обновляющая ценность команды по оценке, полученной от агента j , имеет следующий вид:

$$\Delta Q(s_i(t), a_i(t)) = \alpha(Op_j(i) - Q(s_i(t), a_i(t))) \quad (1)$$

$$Op_j(i) = r_j(t+1) + \gamma Q_j^*(s_j(t+1), a_j(t+1)) \quad (2)$$

$$Q_j^*(s_j(t+1), a_j(t+1)) = \arg \max_{a_j \in A_j} Q(s_j(t+1), a_j(t+1)) \quad (3)$$

Если агенты не имеют конструктивных отношений друг с другом, то зависимость в поведении между этими агентами не является необходимой, что ведет к естественной декомпозиции графа координации.

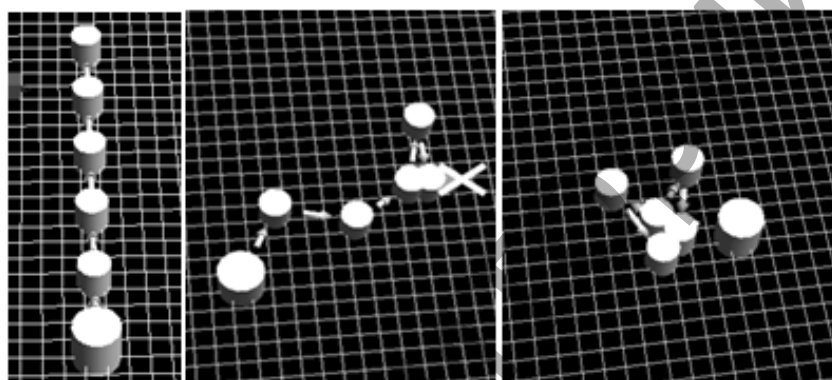
В этом примере агент i не мог выполнить одновременно команду и действие, т.к. выполнение команды трактовалось как отдельное действие. В зависимости от модели многоагентной системы, агенты могут вести переговоры и смешивать команды и действия в один такт времени.

4. Экспериментальные результаты

Для описания графа координации, графа переходов и графа Марковских процессов принятия решений была разработана библиотека моделирования на графах [10], предназначенная для задач робототехники и многоагентного моделирования.

Описанный в главе 1 подход применялся для задачи формирования и поддержания формации заданной формы на модельных агентах [11].

Описанная в главе 3 адаптация на основе команд применялась в задаче моделирования многозвенного робота, обладающего 5-ю степенями свободы [12]. Среда моделирования и многоагентная система, имитирующая робота, изображена на рис. 6.



(a) Начальное положение.
(b) Оптимальная последовательность команд найдена.
(c) «Организационный хаос»

(a) (b) (c)

Рисунок 6 – Моделирование многозвенного «робота-манипулятора»

Каждый сегмент, кроме последнего, мог командами изменять положение всех последующих сегментов. Итоговое расстояние до цели определялось после выполнения действий всеми сегментами. Цель эксперимента – обучение робота достижению цели путем его самоорганизации посредством поиска оптимальных влияний между сегментами. Обученный робот оказался робастным к удалению или добавлению необученных сегментов, а также быстро переучивался при смене цели. Поскольку граф координации был задан заранее, явная декомпозиция пространства поиска по сегментам (текущий сегмент зависел только от предыдущего) показала эффективность в скорости сходимости алгоритма по сравнению с командным и совместным обучением, подверженным проклятию размерности. Рисунок 7 иллюстрирует графики уменьшения суммарной (по всем сегментам) ошибки временной разности для трех алгоритмов обучения с подкреплением в сравнении с совместным обучением (JAL).

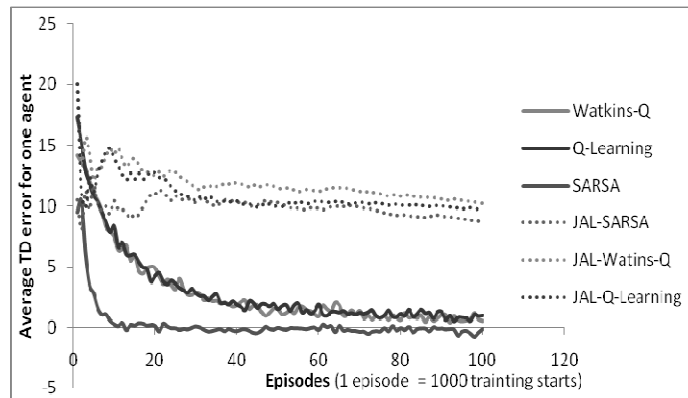


Рисунок 7 – Графики изменения ошибки временной разности между совместным обучением и моделью команд

Выводы

Представленная в работе модель организации взаимодействий в многоагентной содержит набор техник, которые позволяют конкретизировать, оценить и оптимизировать отношения между агентами в системе с целью координации их действий. Представленный подход к адаптации рассмотренных концепций не единственный. В работах [5,6] представлены адаптации данного подхода для задач теории игр, коллективного фуражирования, коллективного принятия решений и ряда других.

Эксперимент с использованием модели показал, что эвристики, конкретизирующие отношения между агентами, применимы для широкого класса задач и могут быть более эффективны, чем изобретение нового алгоритма.

Литература

1. Panait, L. Cooperative Multi-Agent Learning: The State of the Art / L. Panait, S. Luke // *Autonomous Agents and Multi-Agent Systems*, 2005. – (11), 3. – P. 387-434.
2. Gabel, T. *Multi-Agent Reinforcement Learning Approaches for Distributed Job-Shop Scheduling Problems*. PhD Thesis / University of Osnabrueck. – 2009. – 175 p.
3. Lee, E. Actor-oriented design of embedded hardware and software systems / E. Lee, S. Neuen-dorffer, M.J. Wirthlin // *Journal of Circuits, Systems and Computers*. – 2003. – (12). – P. 231-260.
4. Monekosso, N. The analysis and performance evaluation of the pheromone-Q-learning algorithm / N. Monekosso, P. Remagnino // *Expert Systems* – 2004. – 21 (2). – P. 80-91.
5. Dennis, B. Goncalves, *Influence Value Q-Learning: A Reinforcement Learning Algorithm for Multi Agent Systems*, in: Meng Joo Er and Yi Zhou (Eds.), *Theory and Novel Applications of Machine Learning*, Book, I-Tech / B. Dennis, M. Luiz, G. Goncalves. – Vienna, Austria, 2009. – P. 376.
6. Kok, J.R. Sparse cooperative q-learning. *In Proceedings of the XXI international conference on Machine Learning* / J.R. Kok, N. Vlassis. – Banff, Alberta, Canada. – 2004. – P. 61.
7. Guestrin, C. Multiagent planning with factored MDPs. *In Advances in Neural Information Processing Systems (NIPS) 14*. The MIT Press / C. Guestrin, D. Koller, R. Parr. – 2002. – P. 1523-1530.
8. Kok, J. Using the max-plus algorithm for multiagent decision making in coordination graphs, *RoboCup 2005: Robot Soccer World Cup IX* / J Kok, N Vlassis. – 2006.
9. Sutton, R.S. *Reinforcement Learning: An Introduction*. MIT Press / R.S. Sutton, A.G. Barto. – 1998.
10. Kabysh, A. Graph Modeling Framework. BrSTU Robotics Wiki (www.robotics.bstu.by/mwiki), (2012), link: http://robotics.bstu.by/mwiki/index.php?title=Библиотека_моделирования_на_графах (in Russian).
11. Golovko, V.A. Collective Behavior in Multiagent Systems Based on Reinforcement Learning / V.A. Golovko, A.S. Kabysh // *Proceedings of the Tenth International Conference «Pattern recognition and image processing» (PRIP-2009)*, Minsk, Belarus, (19–21 may 2008) – Minsk, 2009. – P. 260-264
12. Kabysh, A. Influence Learning for Multi-Agent System Based on Reinforcement Learning / A. Kabysh, V. Golovko, A. Lipnickas // *Journal of Computing*. – 2012. – (11), 1 – P. 39-44.

ОЦЕНКА ПОГРЕШНОСТИ ОДОМЕТРОВ МОБИЛЬНОГО РОБОТА С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

Введение

Одной из важнейших задач в робототехнике является задача определения точного местоположения робота – проблема локализации или позиционирования. Данная проблема очень важна, так как информация о точном местоположении робота необходима для решения более сложных и комплексных задач навигации, построения пути и построения карты окружающей среды. На сегодняшний день существует несколько различных подходов к решению проблемы локализации. Эти подходы применяют различные сенсоры и алгоритмы обработки данных. Так, один из подходов к локализации основан на анализе данных сенсоров пройденного расстояния и описан в [1], вероятностный подход к локализации и картографированию на основе SLAM-методик с использованием лазерного сканера или дальномеров представлен в [2]. Любые методы оценки позиции робота имеют погрешности, обусловленные различными факторами физической среды. Для оценки этих погрешностей и уточнения реальной позиции робота на данный момент также существуют различные методики. Так, в [3] рассмотрен подход на основе расчета матриц ковариации, метод оценки ошибки одометров на основе данных навигации мобильного робота представлен в [4]. Классической в данной области является работа [5], где разработана методика калибровки и коррекции погрешности одометров, предложены методы проведения эксперимента для оценки различных факторов, влияющих на погрешность. В работе [6] выполнен сравнительный анализ нескольких различных подходов к оценке погрешности одометров, одним из которых является метод нейронных сетей. В работе применен многослойный персептрон для оценки ошибки одометров, который показал лучшие результаты из рассмотренных методов. Этим обусловлена актуальность исследований в области нейросетевых технологий оценки погрешностей сенсоров позиционирования.

1. Постановка задачи

Основная цель исследований, описанных в данной статье, это повышение точности системы позиционирования реального мобильного робота на основе данных одометров с помощью методов искусственных нейронных сетей. Позиционирование робота на основе показаний одометров – наиболее дешевое и, таким образом, распространенное решение в робототехнике. Однако применение одометров для позиционирования робота в пространстве связано с проблемой

быстро накапливающейся погрешности, вызванной различными факторами. В данной работе анализируется влияние различных факторов для конкретного мобильного робота и производится оценка точности показаний одометров с помощью искусственных нейронных сетей. Отличием данной работы от предыдущих работ, рассмотренных выше, являются:

- исследование различных структур нейронной сети для оценки погрешностей робота;
- исследование наилучших стратегий управления роботом для нейросетевой оценки погрешности одометров;
- использование мобильного робота с малыми линейными размерами (в 5 раз меньше чем в [6]). Это значительно повышает погрешности системы позиционирования из-за малой базы и массы робота, однако позволяет оценить качество применения нейросетевой методики.

2. Описание экспериментов

Традиционно для исследования ошибки одометров применяется метод UMBMark, предложенный в [5]. Этот метод основан на анализе движения робота по сторонам квадрата (сторона 4 м) и заключается в применении различных алгоритмов оценки качества этого движения. В данной работе ставились эксперименты над мобильным роботом малых размеров PololuBot (диаметр 12 см, рис.1), что усложняет применение метода UMBMark. Таким образом, для сбора данных при постановке экспериментов робот перемещался по прямой. Были проведены эксперименты для скоростей 0.2 м/с, 0.5 м/с, 0.7 м/с, время движения подбиралось таким образом, чтобы робот проезжал примерно 1,5 метра. При движении робота проводилось измерение показаний одометров с периодичностью в 200 мс, также фиксировалась конечная точка с помощью внешних измерений. На основе полученных данных рассчитывались восемь промежуточных позиций робота по данным одометров. Из данной информации формировались входные образы на нейронную сеть. Более подробно данный метод рассмотрен в 3-м разделе.

В процессе экспериментов был проведен анализ стратегий управления для выбора наилучшей с точки зрения нейросетевого метода оценки. В экспериментах применялись следующие стратегии управления:

Калибровка скоростей колес. Для уменьшения боковой ошибки был выполнен поиск значений скоростей колес, при которых достигается максимально прямолинейное движение. В результате были найдены пары значений для различных скоростей. Недостатком данной стратегии является сильное влияние неточностей механики на траекторию и результаты.

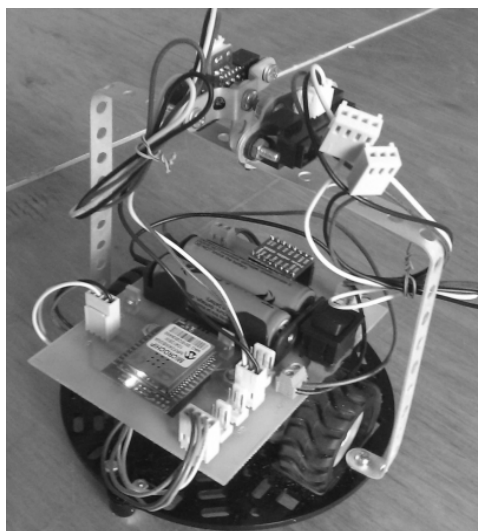


Рисунок 1 – PololuBot: диаметр 12 см, диаметр колеса 4.3 см, одометры, инфракрасные дальномеры, wifi связь

ПИД-регулятор на оба колеса. В этом варианте управление роботом проводилось с помощью алгоритмов пид-регуляции на каждом колесе. Результаты экспериментов показали сложность настройки такого решения, требований к мощности вычислительного процессора, что для малых роботов является существенным недостатком.

ПИД-регулятор на одно колесо. Наилучшая стратегия управления для малого робота с дифференциальной кинематической схемой. При использовании данного подхода были получены наилучшие результаты.

Буксировка робота на холостом ходу. Для оценки влияния электромоторов робота на ошибку одометров и прямолинейность движения был проведен эксперимент с буксированием робота на холостом ходу. В результате были выявлены независимые характеристики одометров и особенности программного обеспечения при работе с ними.

В результате поставленных экспериментов, калибровки и анализа погрешностей механики была установлена величина погрешности одометров, равная примерно 8-10%, зависящая в основном от качества покрытия пола.

3. Математическая модель ошибки одометров

В данной работе эксперименты проводились на роботе с дифференциальной кинематической схемой. Для расчета позиции робота были использованы оптические одометры с разрешающей способностью 12 отсчетов на оборот. Для расчета текущей позиции робота применялась математическая модель [7].

Пусть начальная и конечная позиции робота задаются следующими векторами:

$$P_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix}, \quad P_1 = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix}. \quad (1)$$

В процессе движения одометры позволяют получить мгновенные значения скорости левого и правого колеса v_l, v_r соответственно. Если известен размер колесной базы робота b , то связь между начальной, конечной координатой позиции робота и данными одометров имеет вид:

$$P_1 = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} \frac{v_l + v_r}{2} \cos\left(\theta + \frac{v_l - v_r}{2b}\right) \\ \frac{v_l + v_r}{2} \sin\left(\theta + \frac{v_l - v_r}{2b}\right) \\ \frac{v_l - v_r}{b} \end{bmatrix}. \quad (2)$$

Данную зависимость можно представить через пройденное расстояние и полный угол $\delta_{trans}, \delta_{rot}$ (рис. 2).

Выражения для пройденного расстояния и поворота имеют вид:

$$\delta_{rot} = \frac{v_l - v_r}{b} \quad (3)$$

$$\delta_{trans} = \frac{v_l + v_r}{2}$$

Таким образом, позиция робота, рассчитанная по данным одометров записывается как

$$\delta = (\delta_{rot}, \delta_{trans}).$$

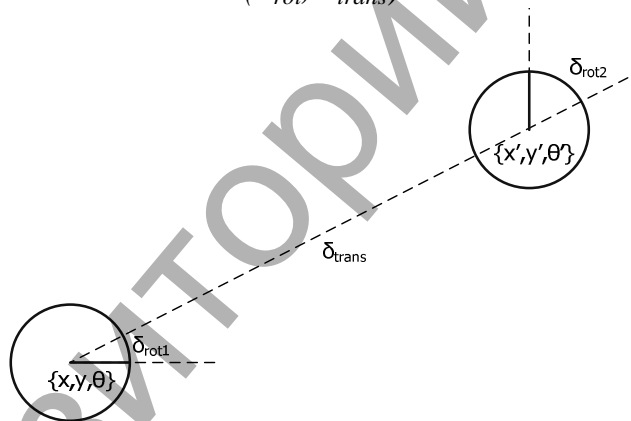


Рисунок 2 – Геометрическая модель движения робота

Погрешность, накапливаемая одометрами, имеет две основных составляющих: систематическую и случайную ошибку расчета. Причины, вызывающие систематическую погрешность позиционирования, – это недочеты при сборке механики и погрешности при проектировании узлов.

Случайные погрешности вызваны проскальзыванием колес и неровностями пола. Исходя из этого, полученные в результате подсчета позиция и угол поворота робота будут содержать в себе эти ошибки:

$$\widehat{\delta}_{trans} = \delta_{trans} + \sigma_{trans}|d| + \varepsilon_{trans}, \quad (4)$$

$$\widehat{\delta}_{rot} = \delta_{rot} + \sigma_{rot}|d| + \varepsilon_{rot} \quad (5)$$

где $\delta_{trans}, \delta_{rot}$ – систематические ошибки при оценке пройденного расстояния и угла поворота робота, нарастающие с количеством пройденного расстояния $|d|$;

ε_{trans} и ε_{rot} – случайные ошибки расстояния и угла поворота; $\hat{\delta}_{rot}, \hat{\delta}_{trans}$ – реальный пройденный путь и реальный угол поворота.

Приведенная выше модель использовалась в программных модулях для расчета координат робота по показаниям одометров. Из представленной модели видно, что ошибка одометров является нелинейной динамической системой.

4. Архитектура нейронной сети

Сложная природа ошибки одометров и выраженное свойство к накоплению с течением времени обуславливает использование искусственных нейронных сетей как средства оценивания величины этой ошибки.

По результатам экспериментов был накоплен временной ряд координат робота, рассчитанных по данным одометров, а также связанные с этим рядом эталонные значения реальной позиции робота. Таким образом, задачей нейронной сети является аппроксимация нелинейной зависимости между данными одометров и реальными данными.

Для решения поставленной задачи был применен двухслойный персептрон (рис. 3) с сигмоидной функцией активации в скрытом слое и линейными выходными нейронами. Количество нейронов входного слоя $l = 18$ было выбрано исходя из данных, полученных в экспериментах. Данные сформировали образы из 18 элементов:

$$\{x_1, y_1, x_2, y_2, \dots, x_8, y_8, t_b, t_e\},$$

где x_1, y_1 – пары координат робота по данным одометров, t_b, t_e – время старта и окончания движения.

Количество нейронов скрытого слоя k варьировалось в процессе экспериментов с нейронной сетью, выходной слой i содержит 2 нейрона, которые определяют координату робота x и y с учетом оценки погрешности нейронной сети.

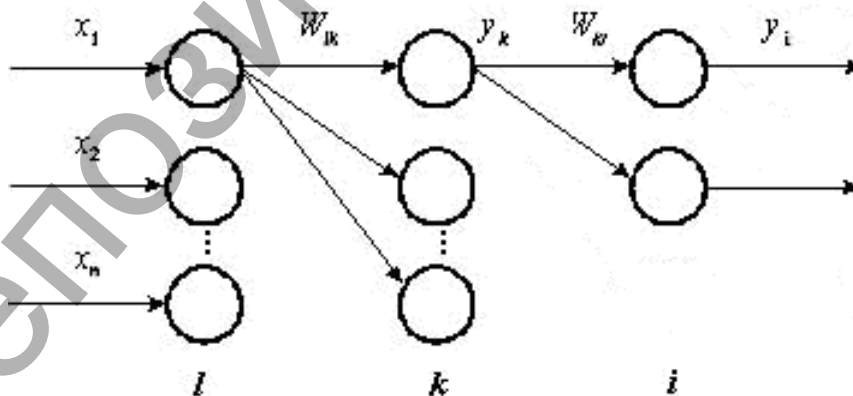


Рисунок 3 – Архитектура нейронной сети

Для обучения использовался метод обратного распространения ошибки. Значение суммарной среднеквадратичной ошибки нейронной сети рассчитывается по формуле:

$$E = \frac{1}{2} \sum_{k=1}^L \sum_j (y_j^k - t_{1,2}^k)^2,$$

где L – размерность обучающей выборки, $t_{1,2}$ – пара эталонных координат реальной позиции робота.

Для обучения нейронной сети в качестве эталонных значений использовались координаты реальной позиции робота. Реальная позиция робота оценивалась с помощью двух подходов – видеорегистрации робота с помощью камеры, закрепленной на потолке, и ручного измерения. Была осуществлена попытка измерения реальной позиции на основе инфракрасных датчиков, но погрешности данных сенсоров не позволили сделать вывод о реальном качестве работы нейронной сети.

Для обучения нейронной сети была подготовлена обучающая выборка размера $L = 131$ образ.

5. Результаты исследований

На основе полученных данных экспериментов было произведено исследование характеристик механики мобильного робота. Малые габаритные размеры робота (12 см X 12 см X 8 см), легкий вес (200 гр) приводят к существенному увеличению погрешностей курсового угла робота. Сильные колебания курсового угла, вызванные небольшой механикой, представлены на рис. 4. Они вызывают существенные погрешности в оценке позиции робота на основе одометров, гораздо больше чем в [6].

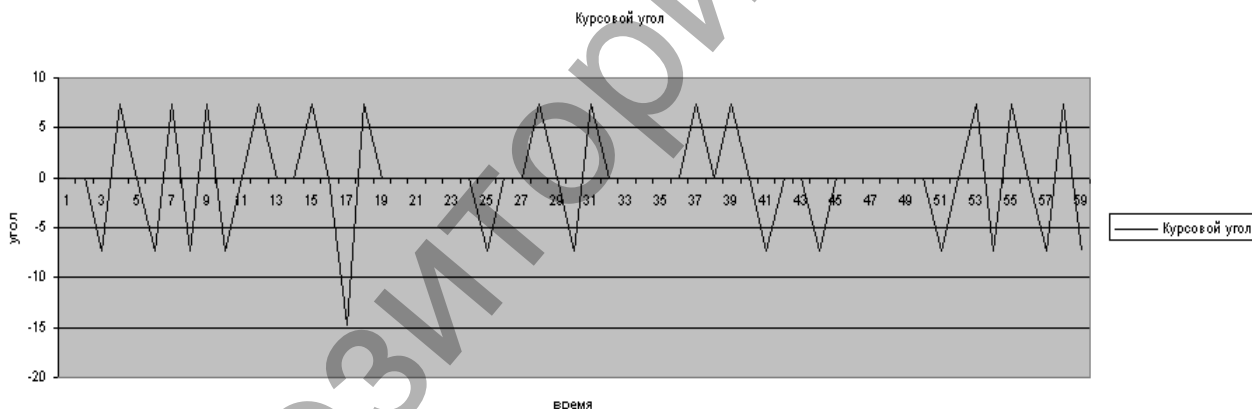
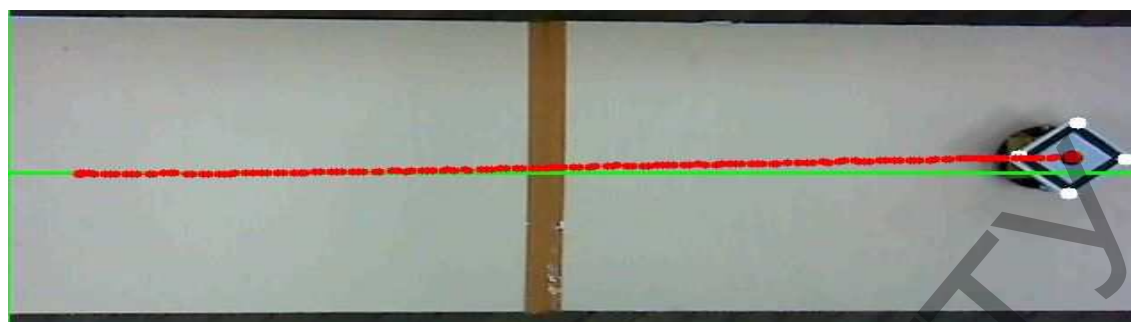


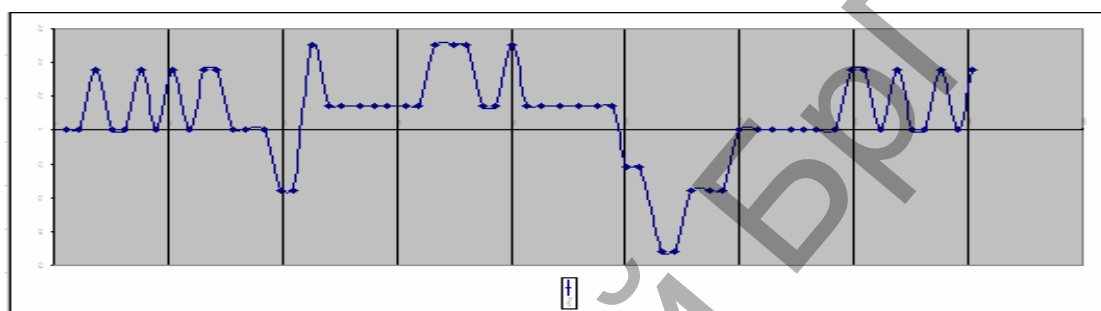
Рисунок 4 – Колебания курсового угла (данные на основе показаний одометров)

Для уменьшения погрешностей в движении мобильного робота, и выделения постоянной составляющей ошибки, в экспериментах для алгоритма управления было введено ПИД-регулирование одного из моторов колес, а также поверхность пола была покрыта специальным материалом. Это позволило существенно уменьшить случайную составляющую ошибки, выделив систематическую. Это позволило более точно готовить обучающую выборку нейронной сети. На рисунке 5б представлен путь робота, который сформирован на основе данных одометров. На рисунке 5а представлен путь робота, который зафиксирован с помощью видеодетектора. Ровная поверхность пола, отсутствие проскальзывания колес, плавное ПИД-регулирование позволяет говорить о том, что случайная составляющая ошибки в данном эксперименте значительно меньше, чем

систематическая. На рисунке 4 наглядно видно, насколько неточны показания одометров в случае систематической ошибки.



a)



b)

a) реальный путь робота, б) путь робота на основе одометров

Рисунок 5

Первоначально в процессе обучения нейронной сети были произведены исследования влияния количества нейронов скрытого слоя на ошибку нейронной сети. Результаты данных исследований приведены в таблице 1.

Таблица 1 – Результаты оценки погрешности одометров

Номер эксперимента	Количество эпох обучения	Среднеквадратичная ошибка обучения НС	Количество нейронов в скрытом слое
1	125	0,08	6
2	21	0,079	6
3	23	0,065	8
4	45	0,021	8
5	20	0,085	10
6	15	0,044	10
7	18	0,081	12
8	32	0,072	12

Исходя из количества входных нейронных элементов и размерности обучающей выборки, количество нейронов в скрытом слое менялось в интервале 6-12 нейронов. При увеличении количества нейронов до 15, нейронная сеть запоминала входные образы и на новых данных показывала неудовлетворительные результаты. Из табл. 1 видно, что на данном интервале количество нейронных элементов не сильно влияет на качество обучения нейронной сети. В итоге для оценки погрешности одометров было выбрано количество нейронов, равное 8.

На рисунках 9 и 10 представлены результаты оценки нейронной сетью ошибки одометров по оси X и Y соответственно относительно реальной позиции робота.

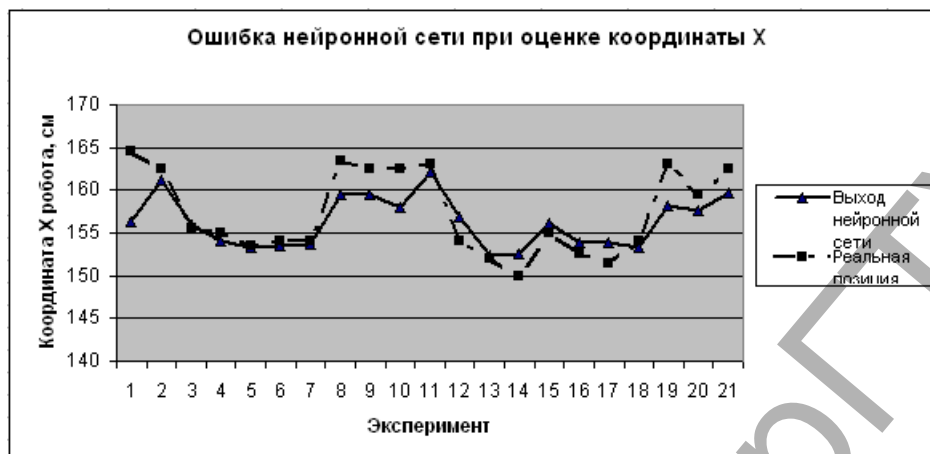


Рисунок 6 – Аппроксимация ошибки одометров нейронной сетью



Рисунок 7 – Аппроксимация ошибки одометров нейронной сетью

С каждым экспериментом систематическая ошибка нарастает и ухудшается оценка нейронной сетью. По координате Y заметное ухудшение происходит после 14 эксперимента, что соответствует примерно 200 метрам пройденного расстояния. Итоговый результат оценки погрешности одометров и погрешности нейронной сети приведен в таблице 2 в сравнении с работой [6], где использовался метод UMBMark и робот больших размеров. При примерно равной ошибке обучения нейронной сети относительная погрешность одометров в данных экспериментах выше, чем в [6], даже с учетом меньшего пройденного расстояния.

Таблица 2 – Сравнение результатов оценки погрешности одометров для различных размеров робота

	Результат	Результат J. Collins
Mean Square Error NN	0,165	0,17
Relative error	3,36%	1,20%
Dataset dimension	161	90
Robot dimension	6 см X 6 см	30 см X 30 см
Пройденное расстояние	1,7 м	5 м

Это согласуется с факторами, сильно влияющими на работа малых размеров. Однако в сравнении с погрешностью одометров (8-10%) относительная погрешность позиции робота по данным нейронной сети (3%) дает более точный результат. Уточнение позиции робота в три раза улучшит точность построения карты и навигации мобильного робота.

Выводы

В результате проведенного исследования были получены характеристики механики для конкретного реального робота. На основании данных характеристик проведена калибровка подсистем управления и позиционирования робота для решения задачи локализации. Для уточнения позиции робота и повышения качества и надежности информации, выдаваемой подсистемой локализации, был предложен нейросетевой модуль. Результаты экспериментов доказали успешность предложенного подхода, особенно в продольном направлении, где ошибка наиболее существенна. Однако в данном подходе имеется ряд недостатков, таких как: необходимость настройки нейросетевого модуля для конкретного робота и окружающей среды, требование к производительности бортового оборудования робота, необходимость использования внешнего видеорежистратора для оценки реальной позиции робота. В дальнейшем для решения данных проблем планируется создание интеллектуальной системы позиционирования, которая смогла бы адаптироваться во время работы к параметрам робота, внешней среды и использовала для оценки позиции, кроме одометров, другие сенсоры робота.

Благодарности

Данная работа выполнена при поддержке гранта Ф11-ЛИТ003 Белорусского республиканского фонда фундаментальных исследований и гранта ГБ 11/117 Министерства образования Республики Беларусь.

Литература

1. D. Fox, W. Burgard, S. Thrun. Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research (JAIR), 1999, pp.11: 391-427.
2. M. Montemerlo, S. Thrun, D. Kollerand, B. Wegbreit. Fast-SLAM2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Con-verges, In Proc. Of the Int. Confs. On Artificial Intelligence (IJCAI), 2003, pp.1151-1156.
3. L. Kleeman. Odometry Error Covariance Estimation for Two Wheel Robot Vehicles, Technical Report MECSE-95-1, Department of Electrical and Computer Systems Engineering, Monash University, 1995.

4. A. Martinelli, R. Siegwart. Estimating the Odometry Error of a Mobile Robot during Navigation, In Procs. Of European Conf. on Mobile Robots, 2003.
5. J. Borenstein, L. Feng. Measurement and correction of systematic odometry errors in mobile robot, IEEE Transactions on Robotics and Automation, 1996, pp.12(6):869-880.
6. Haoming Xu and John James Collins. Estimating the Odometry Error of a Mobile Robot by Neural Networks, In Proc. of International Conference on Machine Learning and Applications, 2009.
7. R. Siegwart, I. Nourbakhsh. Introduction to Autonomous Mobile Robots, The MIT Press, 2004.
8. Z. Zhang. A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

Прокопович Г.А., Сычѳв В.А.

Объединѳнный институт проблем информатики НАН Беларуси

Лаборатория моделирования самоорганизующихся систем

{rprakovich, vsychyov}@robotics.by

ПРЕДОБРАБОТКА СИГНАЛОВ ДЛЯ ИНВАРИАНТНОГО РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Введение

К настоящему времени проведено множество исследований [1], посвящённых применению искусственных нейронных сетей (ИНС) в обработке сложных сигналов различной природы. Однако можно отметить, что в данных работах не уделялось должного внимания проблеме подготовки входных данных для нейросети, предполагая, что данные для обучения и корректного функционирования уже имеются и представлены в виде, доступном для конкретной ИНС. Тем не менее, как показала практика [2], предобработка данных может стать одним из наиболее трудоёмких элементов нейросетевого анализа. Причем иногда знание основных принципов и приемов предобработки исходных данных является более важным, чем знание собственно нейросетевых алгоритмов.

Для решения конкретной задачи недостаточно знать архитектуру и алгоритм работы ИНС. Даже при наличии действующего нейроэмулятора сам процесс решения прикладной задачи, в том числе и подготовка входных данных, целиком ложится на плечи пользователя.

Рассмотрим необходимые этапы нейросетевого анализа:

1) *кодирование входов-выходов*: программно реализованные ИНС могут работать только с числами, поэтому любой сигнал должен подвергаться цифровому преобразованию, т.е. получить количественную оценку;

2) *нормирование данных*: конечный результат работы ИНС не должен зависеть от выбора единиц измерения, для чего входные данные должны быть нормированными и безразмерными;

3) *предобработка данных* заключается в удалении (фильтрации) очевидных регулярностей из входных данных, что облегчает ИНС выявлять нетривиальные закономерности;

4) *обучение нескольких нейросетей* осуществляется для определения оптимальной архитектуры ИНС по производительности, так как конечный результат

обучения зависит не только от её размеров (количества нейронов и самих слов), но и от её начальной конфигурации;

5) *отбор оптимальных сетей*: процесс выбора таких структур ИНС, которые дадут наименьшую ошибку предсказания на известных данных;

б) *оценка значимости предсказаний*: ошибки производимых ИНС-вычислений не должны превышать допустимых задачей ограничений, чтобы не потерять смысл (ценность) предсказанных значений.

Можно заметить, что этапы процесса предобработки входных данных, указанные в пп. 1-3, не связаны непосредственно с ИНС, но, тем не менее, они являются одним из ключевых элементов данной информационной технологии. Результат обучения и функционирования ИНС может решающим образом зависеть от того, в каком виде была представлена информация для ее обучения.

1. Кодирование входов-выходов

По типу входной информации ИНС подразделяются на аналоговые и двоичные. Если первые используют информацию в форме действительных чисел, то вторые оперируют с информацией, представленной в двоичном виде. Аналоговые ИНС, в частности многослойные перцептроны, часто используются для распознавания образов и обработки цветных либо полутонных изображений. Однако современная вычислительная техника основана на цифровой элементной базе, поэтому большинство разработчиков, с целью дальнейшей аппаратной реализации проектируемых ИНС, стремится входные и выходные данные представлять в двоичных кодах. К тому же существуют целые классы ИНС, которые работают только с бинарными либо биполярными кодами. В частности, к ним относятся ИНС типа сетей Хемминга и Хопфилда.

Следующий немаловажный вопрос представления входов-выходов для ИНС заключается в том, что практически все нейросети способны обрабатывать только одномерные массивы, т.е. данные, представленные в виде векторов. Исключением является лишь ограниченное число нейросетей, например, когнитрон или свёрточные ИНС, которые способны обрабатывать двумерные образы как единое целое [1, 2]. Для обработки многомерных данных с помощью остальных архитектур первые требуется преобразовать в одномерные массивы данных. Например, для записи монохромного изображения в ассоциативную сеть Хопфилда его необходимо представить в виде одномерного массива, так как весовую матрицу можно получить лишь с помощью внешнего (матричного) произведения векторов из обучаемой выборки [3].

Для того чтобы монохромное графическое изображение размером $(m \times n)$ привести к необходимому для функционирования ИНС виду, требуется исходный двумерный массив преобразовать в вектор длиной $(m \cdot n)$ при помощи одного из приведённых способов (рисунок 1а и б).

↘	→	→	↘
↘	←	←	↘
↘	→	→	↘
•	←	←	↘

↘	→	→	↘
↘	→	→	↘
↘	→	→	↘
↘	→	→	•

0	1	1	0
1	0	0	1
1	1	1	1
1	0	0	1

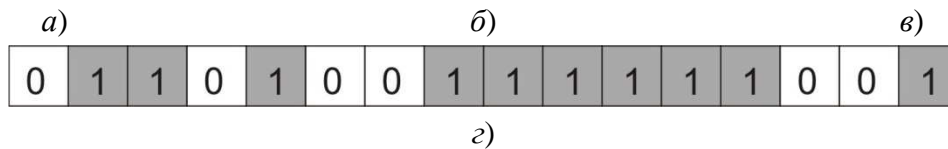


Рисунок 1 – Пример развёртки двумерных массивов в одномерные

На рисунке 1в показан пример, как с помощью двумерного массива, состоящего из 0 и 1, возможно закодировать монохромное изображение буквы «А». На рисунке 1г показан пример развёртки изображённого на рис. 1в двумерного массива указанным в б) способом. Таким образом, полученный вектор отождествляет собой изображение буквы «А», и теперь он может являться входной информацией для ИНС.

2. Нормирование

В связи с тем, что чаще всего ИНС являются промежуточным звеном в АСУ либо системах принятия решений, то входные и выходные данные ИНС могут быть совершенно разнородными величинами. Поэтому становится очевидным, что результаты нейросетевого моделирования не должны зависеть от единиц измерения этих величин, т.е. чтобы сеть трактовала их значения единообразно, все входные и выходные величины должны быть приведены к единому масштабу.

Как правило, нормирование данных часто применяется для того, чтобы уравнять отрицательный вклад, вносимый в модель исследуемого объекта, от различных переменных, а также чтобы обрабатывать совместно разные блоки данных. В результате, нормирование влияет на точность прогнозирования ИНС. Однако неправильный подход к нормированию данных может перечеркнуть все другие меры, принимаемые для увеличения точности работы ИНС.

Нормирование данных необходимо для того, чтобы каждое значение, подаваемое на вход ИНС, находилось в интервале [0, 1]. Расчеты показали, что погрешность прогноза существенно зависит от способа нормирования входных данных. Этому вопросу уделяется внимание во многих работах, например [1, 4], посвященных как прогнозированию нагрузки, так и использованию ИНС в других задачах, где данные по модулю превышают единицу.

Простейшее решение вопроса – это нормирование данных по формуле

$$P_i^H = \frac{P_i}{P_{\max}}, \quad (1)$$

где P_i – значение входного сигнала ИНС, подаваемое на i -й входной нейрон; P_{\max} – максимальное значение входного сигнала в рассматриваемой выборке; P_i^H – нормированное значение. В этом случае для нашей задачи все данные после нормирования находятся, как правило, в интервале $[P_{\min} / P_{\max}; 1]$, где P_{\min} – минимальное значение сигнала в каждой выборке. Однако величина может сильно P_{\min} / P_{\max} ограничивать диапазон снизу.

Следующая формула нормирования лишена этого недостатка и позволяет растянуть данные, подаваемые на вход ИНС, от 0 до 1:

$$P_i^H = \frac{P_i - P_{\min}}{P_{\max} - P_{\min}}. \quad (2)$$

3. Предобработка

Процесс предобработки входных данных для ИНС предполагает использование знаний специалиста из конкретной предметной области для того, чтобы от исходной информации перейти к некоторому минимальному набору признаков. Это позволит принципиально упростить реализацию и уменьшить сложность (и время) обучения ИНС, что, следовательно, приведёт к увеличению производительности и надёжности получаемых решений. Основная задача предобработки входных данных для обучения ИНС заключается в следующем:

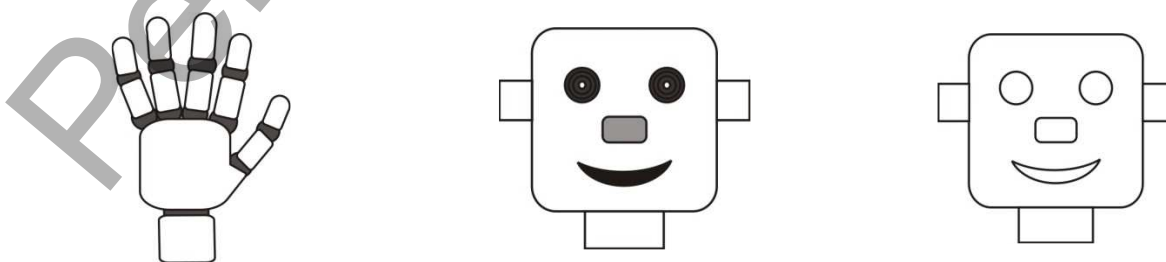
- 1) фильтрация искажений;
- 2) снижение размерности входных данных;
- 3) инвариантность к смещению.

Для решения перечисленных задач обычно используют разработанные и хорошо зарекомендовавшие себя методы из теории обработки сигналов в цифровой и радиотехнике. Однако на практике чаще требуется обрабатывать не только одномерные, но и многомерные сигналы, в частности, такие как графические изображения.

Одним из широко используемых способов предварительной обработки измерительных сигналов является метод преобразования Фурье, который весьма хорошо себя зарекомендовал даже для фильтрации и сжатия цифровых изображений [5]. Например, в [1] предложен метод описания входных образов для перцептронной ИНС с помощью преобразования Фурье, благодаря чему она способна распознавать и классифицировать двумерные образы в независимости от их возможного перемещения, ротации и масштабирования. Однако описанный метод предварительной подготовки использует не всю информацию об образе, а только множество координат (x, y) его контура. Координатное описание контура представляется комплексным числом

$$z(i) = x(i) + j \cdot y(i). \quad (3)$$

Для некоторых классов объектов данный подход значительно упрощает их представление, не уменьшая их информативности (рис. 2а). Однако существуют классы объектов, которые нельзя качественно представить в виде единого замкнутого контура (рис. 2б). В этом случае для распознавания сложных объектов, по-видимому, потребуется использовать лингвистический подход, который будет учитывать иерархию составляющих элементов (рис. 2в). А это приведёт к значительному усложнению архитектуры и увеличению вычислительных затрат самого нейросетевого классификатора.



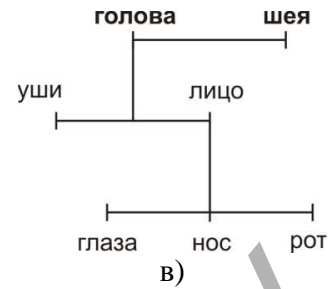
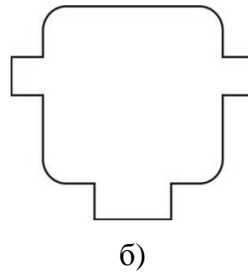


Рисунок 2

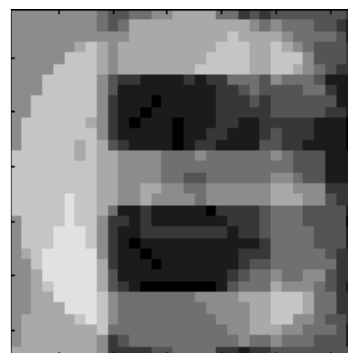
Однако для решения некоторых задач требуются более простые способы для инвариантного распознавания изображений.

4. Декорреляция бинарных изображений

Поскольку изображения содержат большой объём информации, важную роль играют вопросы её представления. Чёрно-белые (бинарные) изображения проще получать, хранить и обрабатывать, чем изображения, в которых имеется много уровней яркости. Несмотря на то, что в бинарных изображениях кодируется информация лишь о силуэте объекта, методы обработки бинарных изображений широко используются при распознавании графических образов, а также в системах технического зрения для пространственного ориентирования роботов. Из проекций бинарных изображений можно извлечь такие простые геометрические характеристики объектов, как площадь, ориентация и положение.

Тем не менее, в явном виде бинарные изображения очень сложно обрабатываются с помощью ИНС. Это связано с тем, что при решении реальных задачах корреляция эталонных векторов существенно снижает надежность идентификации ИНС. Так, например, среди букв латинского алфавита, изображённых на рис. 3а, присутствует корреляция от 36.23 до 85.64% (см. табл. 1). На рис. 3б изображены области пересечения графических образов, причём, чем больше пересечений, тем светлее каждая область.

Для решения описанной проблемы предлагается оригинальный метод декорреляции бинарных векторов, частным случаем которого могут быть чёрно-белые изображения. Данный способ заключается в отображении исходного бинарного вектора $\bar{Y} \in B^N$ в целочисленный вектор $\bar{X} \in Z^n$ меньшей размерности.



а)

б)

Рисунок 3 – Пример пересечения двумерных графических образов

Алгоритм отображения, при котором происходит декорреляция бинарных векторов, заключается в следующем. Пусть имеется некий N -мерный бинарный вектор $\bar{Y} = [y_1, y_2, \dots, y_N]$. Далее разделим его на n равных частей, содержащих

$k+1$ элементов каждый, т.е. $N = n(k+1)$ или $\text{mod}(N, n) = 0$. Таким образом, исходный бинарный вектор $Y \in B^N$ отображается в набор q -мерных векторов, т.е. образ $\bar{X} = [x_1, x_2, \dots, x_n]$. Тогда каждую из n -х частей исходного бинарного вектора можно рассматривать как некоторое целое число $\pm q$, записанное в двоичном коде, причём, первый элемент определяет знак: 0 – «минус», 1 – «плюс», а остальные k элементов – величину q . Например, вектор $Y = [101011110011]$ можно разбить на три фрагмента по четыре элемента в каждом $x_1 = [1010]$, $x_2 = [1111]$ и $x_3 = [0011]$. Первому фрагменту соответствует число 2, второму – 7, а третьему – 3, т.е. при $k=3$ отображение примет вид $\bar{Y} \rightarrow \bar{X} = [2, 7, -3]$, при $k=2$: $\bar{Y} \rightarrow \bar{X} = [1, -3, 2, -3]$.

Одним из главных достоинств предложенного метода является то, что он подавляет имеющиеся корреляции. Рассмотрим два бинарных вектора $\bar{Y}_1 = [01000010]$ и $\bar{Y}_2 = [01011010]$, у которых совпадают 75% значений элементов. Если воспользоваться процедурой отображения с параметром $k=1$ (отображение в пространство $q=2$), при котором каждый вектор следует разбить на четыре фрагмента, получим два целочисленных вектора $\bar{X}_1 = [-1, 0, 0, 0]$ и $\bar{X}_2 = [-1, -1, 0, 0]$. В итоге указанные вектора коррелируют уже на 50%. Если использовать указанную процедуру с параметром $k=3$, то получим два совершенно не коррелирующих отображения $X_1 = [-4, -2]$ и $X_2 = [-5, 2]$.

Таблица 1 – Корреляция изображений букв латинского алфавита

Хемминг %	А	В	С	Д	Е	Ф	Г	Н	К
А		49.61	46.09	36.23	42.19	44.43	43.95	37.01	51.37
В	516		57.03	78.61	75.78	67.87	65.82	61.43	55.47
С	552	440		58.50	66.99	58.11	67.77	45.02	49.41
Д	653	219	425		69.24	63.28	72.36	62.30	51.46
Е	592	248	338	315		85.64	57.03	65.92	65.63
Ф	569	329	429	376	147		52.64	70.51	66.89
Г	574	350	330	283	440	485		44.82	44.92
Н	645	395	563	386	349	302	565		56.93
К	498	456	518	497	352	339	564	441	

Так же немаловажным свойством описанного отображения является то, что оно взаимно однозначно, т.е. по известному \bar{X} всегда можно совершенно точно восстановить его бинарный прообраз \bar{Y} .

Далее рассмотрим, как предложенный метод способен декоррелировать монохромные изображения, приведённые на рис. 3. Для начала с помощью одного из указанных на рис. 1а и 1б способов следует разложить двумерные графические образы размером 32×32 пикселей в вектора. Затем полученные бинарные вектора длиной $N = 1024$ элементов следует отобразить в целочисленные вектора со следующими параметрами: $k = 8$ и $n = N / k = 1024 / 8 = 128$ пикселей. Для наглядности было принято, что $k = q$, т.е. все элементы векторов являются положительными числами в диапазоне от 0 до 255, так как $q_{\text{max}} = 2^8 - 1 = 255$. Ре-

зультат проведённой декорреляции приведён в таблице 2, а на рис. 4 отображены значения элементов декоррелированных векторов трёх первых букв латинского алфавита. Для наглядности на оси абсцисс обозначена только половина точек.

Таблица 2 – Результат декорреляции изображений букв латинского алфавита

Расстояние Эйлера	A	B	C	D	E	F	G	H	K
A		1.0586	0.9696	1.1308	1.2610	1.6088	1.6471	1.0458	1.5328
B	2.9501		0.8136	0.5724	1.0905	1.3165	0.9682	1.0542	0.7220
C	2.7681	2.0128		0.7557	1.1655	0.8976	1.0146	1.0742	0.9898
D	3.6845	4.1347	4.0645		1.1469	1.3253	0.8792	1.1493	0.7892
E	5.5235	6.4267	5.5765	5.2376		1.5305	1.0752	0.4924	1.0487
F	6.5472	7.8325	6.7742	6.1168	2.5134		1.2379	1.4803	1.2208
G	2.5918	2.9243	2.4515	4.3204	5.5242	6.6384		1.1382	0.6568
H	3.0083	3.3943	3.4643	2.9428	5.5020	6.5546	3.7440		1.1394
K	3.8653	5.1055	4.6321	3.6540	5.3526	5.6539	4.5813	2.4406	

Далее необходимо рассмотреть вопрос о том, в какой степени предложенный метод декорреляции будет влиять на схожие образы, относящиеся к одному классу. Главная задача предобработки данных заключается в снижении существующей избыточности во входных данных. Однако необходимо следить за тем, чтобы это не снизило информативность примеров и, тем самым, качество нейропредсказаний.

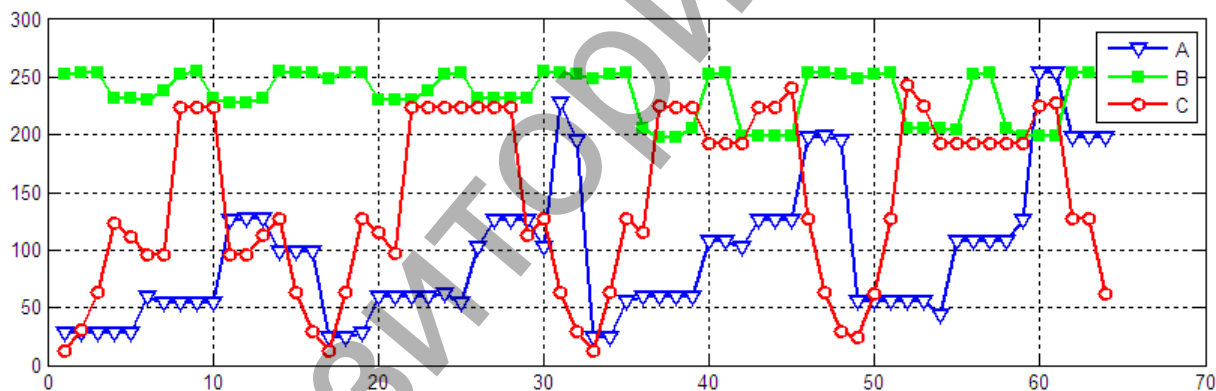


Рисунок 4

На рис. 5 представлены декоррелированные вектора, содержащие информацию о монохромных изображениях латинской буквы "А", выполненной с помощью различных шрифтов. На приведённом графике визуально видна сохранившаяся корреляция между близкими образами. Это происходит благодаря удачно выбранному способу отображения, который при переводе векторов из бинарных в целочисленные увеличивает их евклидовое расстояние по экспоненциальному закону.

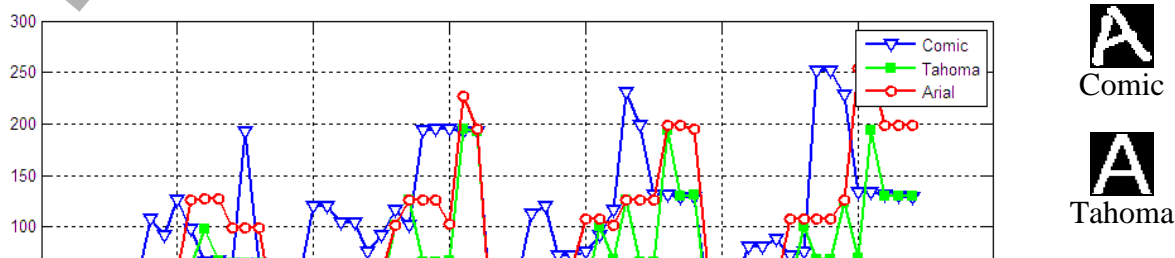


Рисунок 5

На рис. 6 изображён график роста евклидового расстояния по мере увеличения параметра k от 1 до 16. На графике красными и синими линиями отображены евклидовы меры между похожими образами (буква «А» с различным написанием), а зелёными – между буквой «А» и «В». Таким образом, с ростом пространства k растёт и расстояние между образами, причём, чем больше разница между исходными образами была изначально, тем больше будет расстояние между соответствующими отображениями. Это заметно на представленных графиках: на начальном этапе евклидово расстояние между различными образами («А» и «В») было меньше, чем между образами из одного класса. Однако с ростом k расстояние между последними значительно сократилось по отношению к образу «В».

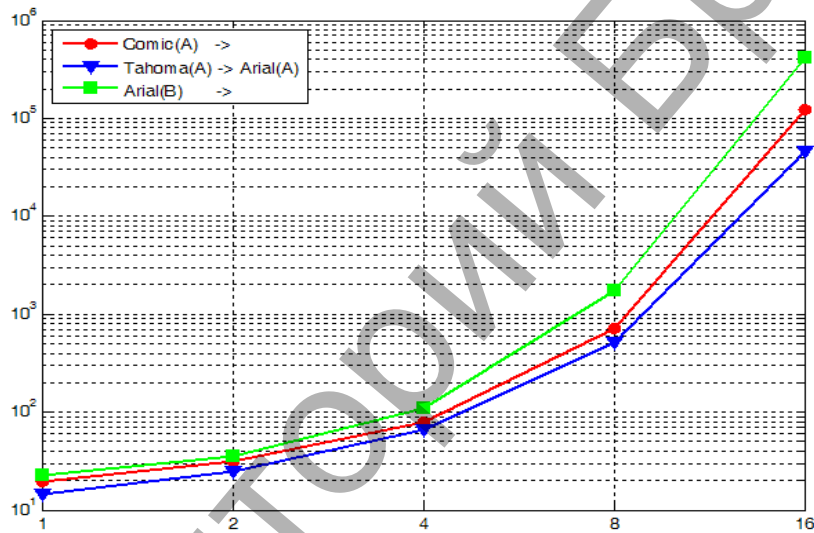


Рисунок 6 – Процесс экспоненциального роста евклидовой меры с ростом параметра k

Чтобы выявить способность сохранять близость между подобными образами в предложенном методе декорреляции, найдём среднюю ошибку распознавания. Допустим, что имеется два близких бинарных образа \bar{Y}_1 и \bar{Y}_2 , отличающихся только на один бит. В качестве примера, рассмотрим случай, когда $N=8$. Если $\bar{Y}_1 = [00000000]$, то по нашему предположению \bar{Y}_2 может принять один из 8 возможных вариантов. Однако так как бинарный код является позиционным, то в зависимости от расположения каждой единицы в бинарном слове меняется и значение всего слова. Поэтому рассмотрим все случаи:

$$\begin{aligned} \bar{Y}_2^1 = [00000001] &\rightarrow X_2^1 = 1; & \bar{Y}_2^5 = [00010000] &\rightarrow X_2^5 = 16; \\ \bar{Y}_2^2 = [00000010] &\rightarrow X_2^2 = 2; & \bar{Y}_2^6 = [00100000] &\rightarrow X_2^6 = 32; \\ \bar{Y}_2^3 = [00000100] &\rightarrow X_2^3 = 4; & \bar{Y}_2^7 = [01000000] &\rightarrow X_2^7 = 64; \\ \bar{Y}_2^4 = [00001000] &\rightarrow X_2^4 = 8; & \bar{Y}_2^8 = [10000000] &\rightarrow X_2^8 = 128. \end{aligned}$$

Тогда среднее арифметическое значение будет равно

$$\langle X_2 \rangle = \frac{1}{N} \sum_{i=1}^N X_2^i = 31.875,$$

а учитывая, что $N_2^{\max} = 128$, средняя относительная погрешность предложенного метода декорреляции будет равна

$$\langle \varepsilon \rangle = \frac{\langle X_2 \rangle}{X_2^{\max}} \cdot 100\% = 24.9\%.$$

Несмотря на то, что ошибка на конкретном участке образа оказывается достаточно большой, на практике образы состоят из значительного числа составных элементов, большая часть из которых для близких образов одинакова. В результате, близкие образы и после декорреляции остаются в одном классе.

5. Инвариантность к искажениям

Несмотря на то, что предложенный метод отображения сохраняет корреляцию между подобными образами, он не способен противостоять таким искажениям, как масштабирование, ротация и смещение. Так, при повороте графических изображений корреляция между образами практически полностью пропадает (рис. 7). Однако, как было показано ранее, после проведения декорреляции бинарных образов они отображаются в новом целочисленном пространстве и могут рассматриваться как последовательности значений неких непрерывных функций. Другими словами, предложенное отображение можно рассматривать, в некотором роде, как цифро-аналоговое преобразование. Поэтому к полученным данным может быть применено большинство из известных методов обработки сигналов.

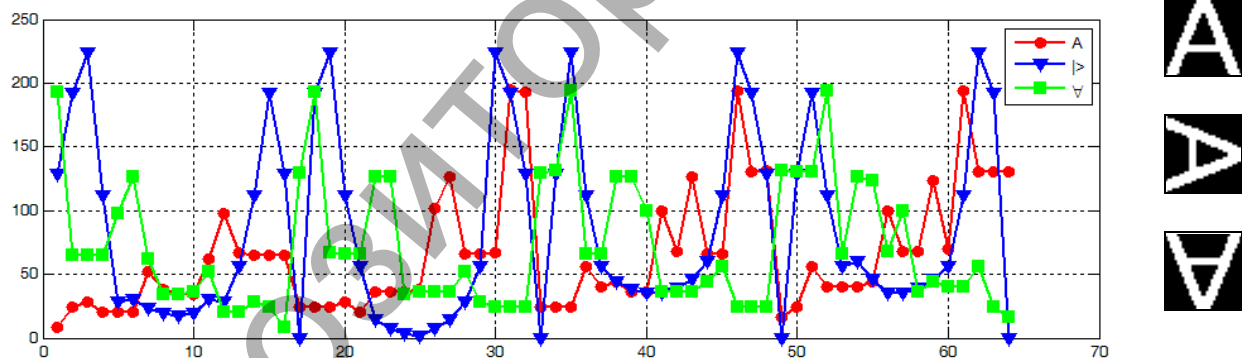


Рисунок 7 – Пример отрицательного влияния искажений типа «поворот»

Например, в работе [6] приведено описание многослойной перцептронной сети, которая производит классификацию сильно искажённых образов. Ключевым элементом описанного классификатора является препроцессор, выходные сигналы которого представляют собой последовательность компонентов дескрипторов Фурье. Однако единственным препятствием для использования указанного метода является различие способов представления начальных данных: поставленные в начале статьи требования не позволяют представлять графические образы в виде их контуров, так как за этим последует потеря семантической информации.

Решением данного противоречия может служить то, что предложенный в п. 4. метод декорреляции производит отображение всего образа, не разделяя его на какие-либо функциональные части. Поэтому с целью создания нового клас-

сификатора, устойчивого к искажениям во входных образах, было предложено в качестве препроцессора использовать предложенный метод декорреляции, а полученные сигналы подвергнуть дальнейшему кодированию с помощью дискретного преобразования Фурье. С этой целью исходный целочисленный вектор \overline{X}_n следует разбить на две равные части $a = [x_1, x_2, \dots, x_{n/2}]$ и $b = [x_{n/2+1}, x_{n/2+2}, \dots, x_n]$, где $\text{mod}(n, 2) = 0$, и представить его в виде следующего комплексного числа:

$$z(i) = a(i) + j \cdot b(i), \quad (4)$$

где i – номер очередной пары измерительных данных, описывающих образ. Далее воспользуемся дискретным преобразованием Фурье:

$$F_i = \sum_{k=0}^{m-1} z(i) \exp\left(-j \frac{2\pi}{m} ik\right), \quad (5)$$

где $m = n/2$ – количество точек описания структуры.

В результате кодирования будет получен вектор $\overline{F} = [F_0, F_1, F_2, \dots, F_{m-1}]$, элементы которого представляют отдельные дескрипторы Фурье-преобразования. Данный вектор будет способен определять уникальность структуры первоначального образа, но в совершенно другом пространстве параметров, в котором путём простых преобразований можно добиться инвариантности к масштабу, ротации и смещению исходных образов. Полученное свойство положительным образом скажется на процессе распознавания образов.

Таблица 3 – Результат декорреляции изображений букв латинского алфавита

Расстояние Эйлера	А	В	С	Д	Е	Ф	Г	Н	К
А		0.8042	0.9312	1.1180	1.4706	1.3469	1.1739	1.0058	0.9428
В	3.7452		0.8288	0.8907	1.0912	1.2994	1.0274	0.5783	0.8841
С	3.7501	3.7983		1.0038	1.3236	0.9328	0.8786	0.9186	0.6975
Д	6.4267	6.1613	6.4399		1.1671	1.3213	1.2368	1.0907	1.0172
Е	5.5215	4.3699	4.6121	3.6919		1.5274	1.2318	0.9642	1.3660
Ф	4.0846	3.9866	3.4590	4.9804	3.5631		0.9991	1.3410	1.0764
Г	3.3943	4.4287	3.4020	5.5020	4.3784	3.0222		1.0530	0.9501
Н	3.2106	3.9793	3.2601	5.4340	4.0729	2.5655	1.5448		0.8988
К	3.8060	3.3751	3.2957	6.6215	4.9729	3.4618	4.0730	3.5903	

Заключение

В работе рассматривается проблема предобработки входных данных для искусственных нейронных сетей, необходимой для эффективного распознавания данных с помощью ИНС. Предлагается метод декорреляции бинарных изображений, позволяющий понизить корреляцию между образами различных классов и уменьшить объём информации, подаваемой на вход ИНС при распознавании данных образов без потери информативности.

Достоинством данного метода является сохранение корреляции между подобными образами и её снижение между образами разных классов. К недостаткам можно отнести недостаточно высокую устойчивость к таким искажениям, как масштабирование, ротация и смещение входных образов.

Полученные в работе результаты могут быть использованы в системах предобработки классификаторов и систем принятия решений.

Литература

1. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М.: Финансы и статистика, 2004. – 342 с.
2. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – 2-е изд. – М.: Издательский дом «Вильямс», 2006. – 1104 с.
3. Головкин, В.А. Нейронные сети: обучение, организация и применение. Кн. 4: учеб. пособие для вузов / Общ. ред. Л.И. Галушкина. – М: ИПРЖР, 2001. – 256 с.
4. Барский, А.Б. Нейронные сети: распознавание, управление, принятие решений / А.Б. Барский. – М.: Финансы и статистика, 2004. – 176 с.
5. Крот, А.М. Быстрые алгоритмы и программы цифровой спектральной обработки сигналов и изображений / А.М. Крот, Е.Б. Минервина. – Минск: Навука і тэхніка, 1995. – 407 с.
6. Dinh Nghia Do, Osowski, S. Shape recognition using FFT preprocessing and neural network // Compel, 1998. – Vol. 17, No.5/6. – P. 658-656.

KESTAS RIMKUS, ARŪNAS LIPNICKAS

Kaunas University of technology

ANTON KABYSH

Brest State Technical University

DIGITAL CAMERA AND 3D LASER RANGE FINDER SYSTEM CALIBRATION FOR ROBOT NAVIGATION

Abstract

Navigation of autonomous mobile robots in 3D needs information about environment they are operating. One of the ways to get that information is by the use of various 3D range finders. Currently, modern 3D laser scanners are very expensive, therefore in the paper we propose the new 3D range scanner composed of the 2D laser scanner with an extra gear supplying the third degree of freedom. Such combination allowed building a non-expensive 3D range scanner, suitable for mobile robots navigation in 3D. Scanning system effectiveness was evaluated between the known and measured object dimensions.

This paper describes the calibration of digital camera with 3D laser range finder system. Then the 2D image matrix elements can be attributed to the depth of the 3D laser scanner data. Such mapping might be also available in the reverse option to 3D laser scanned points assign the real colour of an object from the digital camera data array. 3D laser scanning system can provide spatial information about objects in the environment regardless of the texture. The synergy of these systems, we get not only the position of objects in environment, but also the texture. Composed of 2D image data array depth map can be used for identification of objects in the environment, avoiding obstacles etc.

Objectives. Create and calibrate 3D scanning system from digital camera and 2D laser range finder with an extra gear supply. From the synergy of these systems, get not only the position of objects in environment, but also the texture.

1. Experimental setup. In our work, we use 2D laser range finder UBG-04LX-F01 (Rapid-URG) [3] for area scanning and 3000 – Hitec- HS-422HD servomotor for

supplying the third degree of freedom to the scanning system. The light source of the sensor is infrared laser of wavelength 785nm with laser class 1 safety.

The designed system has digital camera (type FFMV-03MTC-60, mfg. Point Grey, Canada), mounted in parallel with laser range finder UBG-04LX-F01. In our work we have applied Jean-Yves Bouguet's well known and widely used Camera Calibration Toolbox for MATLAB [6] to calibrate the digital camera parameters and to compensate nonlinear radial and tangential distortion of the pinhole camera. Camera calibration was made only to rectify the images and to find the camera model based parameters. The equation of the pinhole camera model is [1, 2]:

$$\begin{bmatrix} u \\ v \\ 0 \end{bmatrix} = \frac{f}{z_c} \cdot \begin{bmatrix} k_u & 0 & \frac{u_0}{f} \\ 0 & k_v & \frac{v_0}{f} \\ 0 & 0 & \frac{1}{f} \end{bmatrix} \cdot \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (1)$$

where u, v – are the retinal coordinates, x_c, y_c, z_c – the world coordinates, k_u, k_v – the scale factors along the axes of pixel coordinates, u_0, v_0 – the pixel coordinates of the principal point (orthogonal projection of the optical center on the image plane), f – the focal length.

Laser range finder data are expressed in spherical coordinates. The sensor model can be written as [4]:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_i c_j & -c_i s_j & s_i & c_i d_x + s_i d_z \\ s_j c_i & c_i & 0 & 0 \\ s_i & s_i s_j & c_i & s_i d_x + c_i d_z \end{bmatrix} \begin{bmatrix} \rho_{ij} \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2)$$

where $c_i = \cos(\varphi_i)$, $c_j = \cos(\theta_j)$, $s_i = \sin(\varphi_i)$, $s_j = \sin(\theta_j)$, ρ_{ij} is the j -th measured distance with corresponding orientation θ_j in the i -th scan plane, which makes the angle φ_i with the horizontal plane. The offset of the external rotation axis from the centre of the mirror in the laser frame has components $d_x = 90 \text{ mm}$ and $d_z = 20 \text{ mm}$. $[x, y, z]^T$ is the coordinates of each measured point relative to the global frame (with its origin at the centre of the rotation axis, the x -axis pointing forward and the z -axis toward the top).

After multiplication with the laser scanner data (ρ_{ij}) (2), the following structure re-

sults, this can be simply implemented and takes up only little computing time:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_i c_j \rho_{ij} + c_i d_x + s_i d_z \\ c_i s_j \rho_{ij} \\ s_i \rho_{ij} + s_i d_x + c_i d_z \end{bmatrix} \quad (3)$$

2. Results. 3D scanned calibration area view in Matlab:

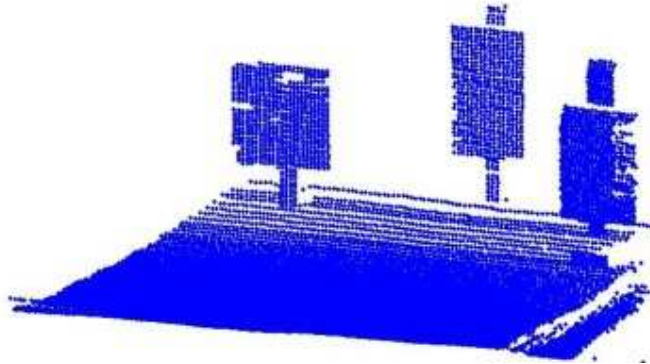


Fig. 1 – 3D scanned calibration area view in Matlab

Base on (1) and (3) using least squares (lsq) and 24 calibration points we fit the model function:

$$f(x) = cx(3) + \frac{cx(2) \cdot (x_c + cx(1))}{z_c - cx(4)} \quad (4)$$

$$f(y) = cy(3) + \frac{cy(2) \cdot (y_c + cy(1))}{z_c - cy(4)} \quad (5)$$

With coefficients cx and cy : $cx = [111.2 \ 479.1 \ 254.7 \ 39.0]$; $cy = [175.8 \ 550.4 \ 235.3 \ 19.8]$.

Calibrated 24 points (corners of boxes) from 3D point cloud (Fig. 1) and undistorted image we can see in Fig. 2.

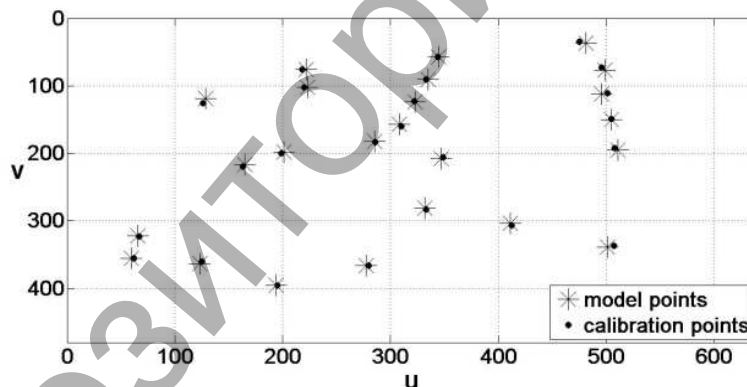


Fig. 2 – Calibration results

In Fig. 2 and Fig. 3 we can see how good model points (Asterisk) correspond to calibration points (dots)

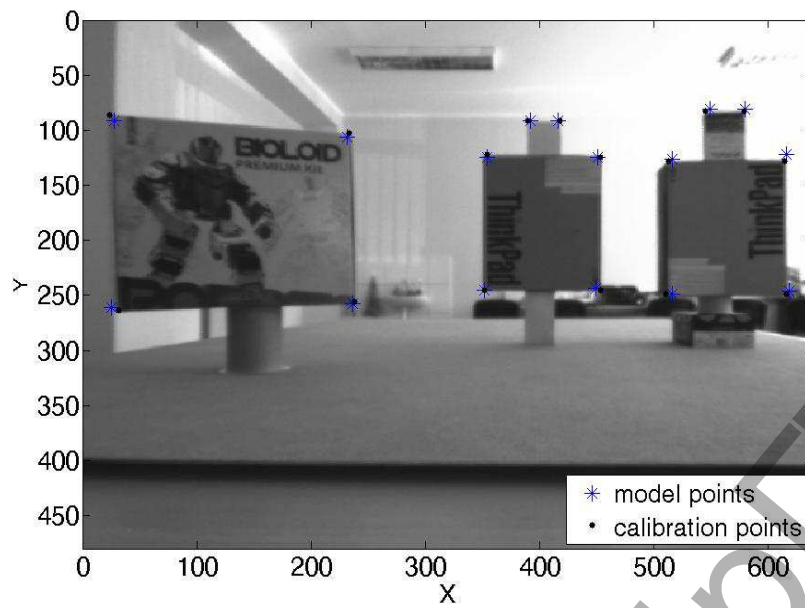


Fig. 3 – Calibration results on (one from two) picture

After carried out calibration, we get the scanned scene Fig. 4.



Fig. 4 – Scene after calibration

Algorithm for CCD camera integration with 3D laser scanner is shown in Fig. 5.

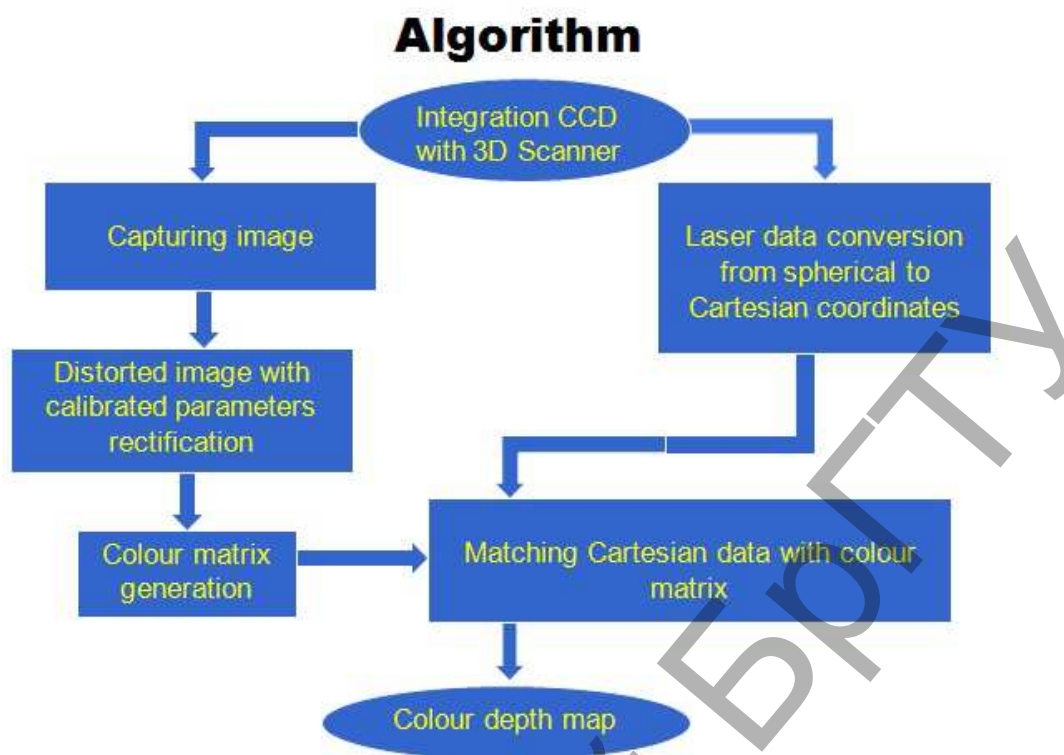


Fig. 5 – Algorithm for CCD camera integration with 3D laser scanner

Conclusions. We created the system which makes the colour depth map using CCD camera integration with 3D Laser range scanner. Calibration point's Euclidian distance absolute mean error was 4.7 pixels. This system can be used for robot navigation, 3D SLAM, object recognition or other tasks.

ACKNOWLEDGMENT

This work has been supported by the Research Council of Lithuanian, project TAP-10078.

References

1. Lipnickas, A.A Stereovision System for 3-D Perception // Electronics and Electrical Engineering / A. Lipnickas, A. Knyš. – Kaunas: Technologija, 2009. – No. 3(91). – P. 99-102.
2. Wulf, O. Fast 3D Scanning Methods for Laser Measurement Systems / O. Wulf, B. Wapner // International Conference on Control Systems and Computer Science, vol 1, Bucharest, Romania, July 2003. – P 312-317.
3. Joochim, C. Development of a 3d mapping using 2d/3d sensors for mobile robot locomotion / C. Joochim, H. Roth // The 2008 IEEE International Conference of Technologies for Practical Robot Applications, TePRA, 2008. – P. 100-105.
4. Scaramuzza, D. Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes / D. Scaramuzza, A. Harati, R. Siegwart // Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 2007.
5. D. Klimentjew, D. 3D Scene Reconstruction Based on a Moving 2D Laser Range Finder for Service-Robots / D. Klimentjew, M. Arli, J. Zhang // Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics December 19 -23, 2009, Guilin, China – P. 1129 – 1134.
6. http://www.vision.caltech.edu/bouguetj/calib_doc. Accessed 2011-03-03.
7. Joochim, C. The Indoor SLAM using Multiple Three Dimension Sensors Integration / C. Joochim, H. Roth // IEEE International Workshop on Intelligent Data Acquisition and Advanced

Дёмин В.В., Дунец И.П.

Брестский государственный технический университет

БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ АВТОНОМНАЯ МОБИЛЬНАЯ ПЛАТФОРМА РОБОТА-ГИДА

Описание

В этой статье описываются алгоритмы навигации автономного мобильного робота, использующего датчики нижнего ценового диапазона. Разработанные алгоритмы были использованы при разработке системы навигации робота-гида.

Введение

В настоящее время в мире проекты робота-гида разрабатывают различные крупнейшие корпорации (Sony, Hitachi, Toshiba, General Motors, Siemens и др. [1-3]). Эти устройства с успехом были применены в кинотеатрах (ED Corporation [4]), в Сеульском национальном историческом музее, в Мадридском офисе Santander's Group City [5], региональном археологическом музее Agrigento, Штутгартском институте мануфактуры инженерии и автоматизации, форумах и корпоративных офисах (Kaikan Exhibition Hall в Toyota City [6]). Главной сложностью применения таких систем является высокая стоимость и закрытые исходные коды. Это делает экономически непривлекательным использование любой из существующих платформ для создания автоматизированных выставок в небольших экспозициях и музеях. В 2014 г. в РБ будет проводиться чемпионат мира по хоккею. Событие такого масштаба будет способствовать многократному увеличению туристов, посещающих нашу страну. Для увеличения числа посетителей выставки будут нуждаться в технических новшествах. Такими ноу-хау могут стать автоматизированные выставки, где экскурсоводами выступают роботизированные гиды. Идея создания робота-гида получила поддержку, и в настоящий момент ведется разработка макета и программной системы управления роботом.

Главной особенностью робота-гида является взаимодействие с людьми на выставках в закрытом пространстве. В помещении располагается предопределённое число экспонатов (рис. 1). У робота есть информация о карте помещения, координатах экспонатов и координатах начала движения. Для управления выбором экскурсии используется сенсорный экран. Каждый посетитель может выбрать интересующий его маршрут. После определения маршрута прокладывается путь к экспонатам. В процессе движения воспроизводится мультимедийная развлекательная информация о выставке и, по прибытию на место, об экспонатах.

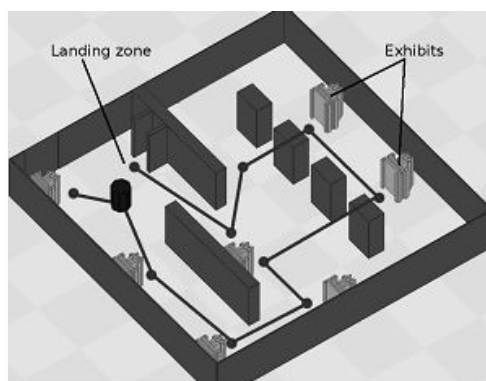


Рисунок 1 – Модель комнаты-выставки для робота-гида

В настоящий момент, изготовлен макет робота-гида и осуществляется разработка программной системы управления роботом. Для разработки и отладки алгоритмов использована система компьютерного моделирования Player/Stage.

Для уменьшения затрат в проекте используются только сенсоры низшего ценового диапазона. Такие сенсоры обладают достаточной точностью, но требуют дополнительной обработки получаемых данных. Для определения препятствий в проекте применяются ультразвуковые и инфракрасные датчики, алгоритмы обработки данных от которых разработаны с опорой на предыдущие исследования, проведенные в СНИЛ «Робототехника» БрГТУ.

1. Платформа робота

Роботизированная платформа разрабатывается в СНИЛ «Робототехника» Брестского государственного технического университета. Внешний вид компьютерной модели робота показан на рисунке 2. На сегодняшний день также разработан макет робота-гида, приведенный на рисунке 3. Макет имеет диаметр 46 см, оснащен одометрами, гироскопом, ультразвуковыми датчиками, инфракрасными датчиками и двумя wi-fi камерами. Для передвижения платформа макета оборудована четырьмя колесами, два из которых свободно вращаются на 360 градусов относительно своей оси, а два оборудованы электроприводами. Каждое из приводных колес оснащено оптическим одометром, позволяющим отсчитывать 64 деления за оборот колеса. Для управления движением робота используется дифференциальная схема управления, при которой поворот платформы осуществляется за счет разницы скоростей вращения колес. Максимальная скорость движения робота – 1,5 м/с.

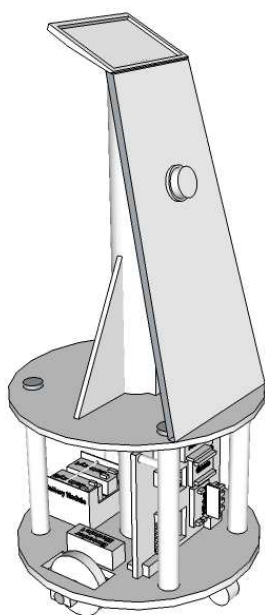


Рисунок 2 – Компьютерная модель робота-гида



Рисунок 3 – Фотография прототипа платформы робота-гида

2. Сенсорная модель робота

Для определения препятствий используются две сенсорные модели: модель инфракрасных датчиков [7] и модель ультразвуковых датчиков [8, 9] (рис. 4а, 4б). Каждая из моделей имеет некоторые свои недостатки. Использование только одной из моделей делает затруднительным навигацию в динамическом пространстве. Комбинирование данных от двух моделей позволяет повысить качество детектирования препятствий. Для повышения надежности работы сенсоров полученные данные подвергаются медианной фильтрации с шириной окна фильтра, равной 5. Значение ширины окна подобрано экспериментально.

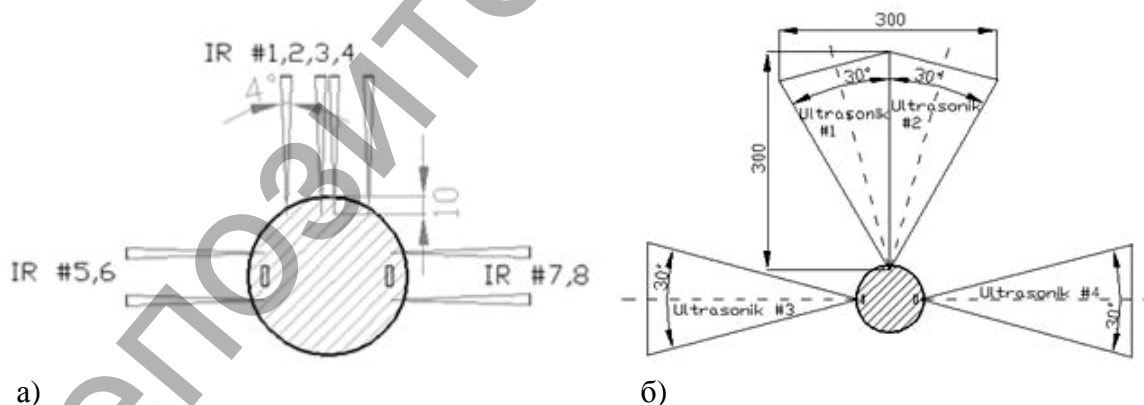


Рисунок 4 – Модель инфракрасных сенсоров (а) и ультразвуковых сенсоров (б)

Для проведения экспериментов моделирования были созданы точные модели сенсоров и актуаторов для системы Player/Stage. Для того чтобы приблизить получаемые данные к реальным, была задана систематическая ошибка всех сенсоров в размере 10%.

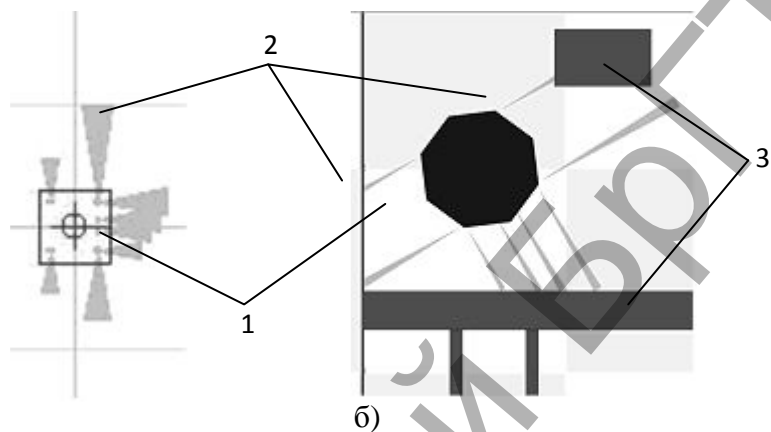
2.1. Обработка данных поступающих от сенсоров

В процессе обработки данных очень важно знать точную ориентацию робота в пространстве. Знание ориентации робота позволяет получить представление о положении препятствий не только относительно робота (рис. 5а, 6а), но также

получить их глобальные координаты. На рисунках 5б и 6б представлена графическая интерпретация показаний инфракрасного и ультразвукового сенсоров соответственно.

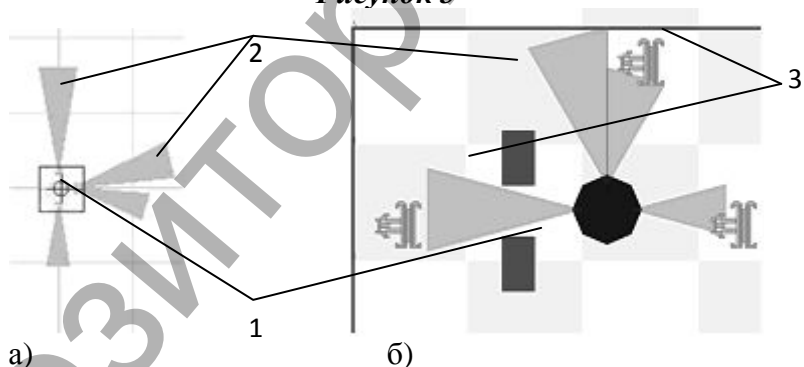
Левое изображение показывает, как выглядит процесс симуляции, а правое показывает данные, поступающие от сенсоров относительно самого робота.

Существует множество точек, где текущие данные от сенсоров будут выглядеть идентично. Это затрудняет локализацию робота при помощи алгоритмов группы SLAM (Simultaneous localization and mapping). Для повышения точности позиционирования робота используются вспомогательные сенсоры, такие как гироскоп и одометры.



а) – Визуализация данных поступающих от инфракрасных сенсоров относительно робота (а) и модели окружающей среды (б); 1 – робот, 2 – графическое отображение покрытия инфракрасных лучей, 3 – стены препятствий

Рисунок 5



а) – Визуализация данных, поступающих от ультразвуковых сенсоров относительно робота (а) и модели окружающей среды (б); 1 – робот, 2 – графическое отображение покрытия ультразвуковых лучей, 3 – стены препятствий

Рисунок 6

2.2. Инфракрасные сенсоры

Одним из типов сенсоров, которые используются при оснащении робота, являются инфракрасные дальномеры. Для оснащения робота были использованы сенсоры SHARP серии GP2Y0A21 с диапазоном измерений 10-80 см. Принцип их работы основан на измерении угла отражения инфракрасного излучения. Полученный отраженный от препятствия импульс фокусируется с помощью линзы и передается на линейную CCD матрицу. На основе засветки площади матрицы определяется угол отражения и рассчитывается расстояние до объекта. Этот метод является более защищенным от случайных помех возникающих из-за разницы цвета и отражающей способности поверхности материалов препятствий.

Возвращаемое сенсором значение лежит в диапазоне от 0 до 3 В [10]. Считывая показания с помощью аналогового входа микроконтроллера, используя формулу (1), можно провести перерасчет вольт в сантиметры.

Формула пересчета показаний сенсора [10]:

$$R = \frac{m}{V+b} - k, \quad (1)$$

где m , b , k – константы, получаемые из спецификации на сенсор [10], V – значение, получаемое от сенсора, R – расстояние до объекта в см.

После подстановки констант формула (1) примет следующий вид:

$$R = \frac{2014}{V+5} - 1 \quad (2)$$

В случае если дистанция от сенсора до препятствия составляет менее 10 см, то происходит сбой в работе сенсора. В случае возникновения такого сбоя, как видно из рисунка 7, препятствие детектируется на более дальней дистанции, чем 10 см. Для устранения этих сбоев сенсоры устанавливаются в глубине корпуса робота.

Данные сенсоры обладают как положительными, так и отрицательными характеристиками.

Положительные:

- 1) высокая точность измерений;
- 2) малый угол обзора, что позволяет получить точную позицию препятствия в пространстве.

Отрицательные:

- 1) сбой при измерениях в зависимости от цвета поверхности (черная поверхность имеет угол отражения на 19% меньше, чем белая);
- 2) при малом угле обзора требуется большое количество сенсоров, чтобы исключить возможность пропуска препятствия;
- 3) если препятствие частично попадает в область обзора, то оно детектируется как далеко отстоящее.

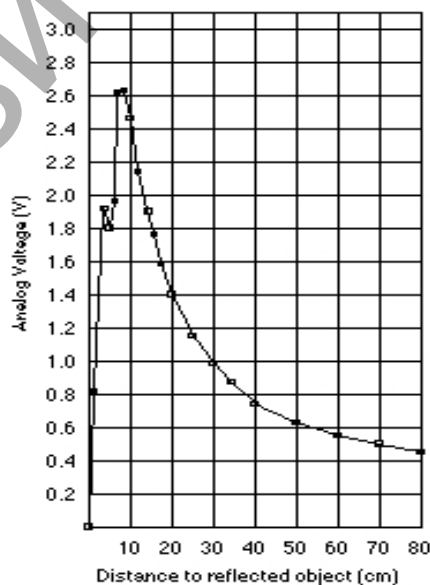


Рисунок 7 – Выходная характеристика для сенсора GP2Y0A21

2.3. Ультразвуковые сенсоры

Второй тип используемых на работе сенсоров – это ультразвуковые сенсоры. Робот оснащен Seeedstudio Ultrasonic SEN136B5B с диапазоном измерений 3-300 см [11]. Принцип работы сенсоров заключается в формировании серии модулированных ультразвуковых импульсов с помощью излучателя и получении отраженного эха при помощи приемника. Подсчет расстояния до препятствия осуществляется на основе исчисления времени получения отраженных импульсов. Формула пересчета показаний сенсора имеет следующий вид [11]:

$$R = \frac{V}{29 * 2}, \quad (3)$$

где V – значение, получаемое от сенсора, 29 мс/см – скорость звука в воздухе, R – расстояние до объекта в см.

Для оценки показаний сенсора используется диаграмма направленности, предоставляемая разработчиками, изображенная на рисунке 8. Угол обзора сенсора составляет порядка 30 градусов.

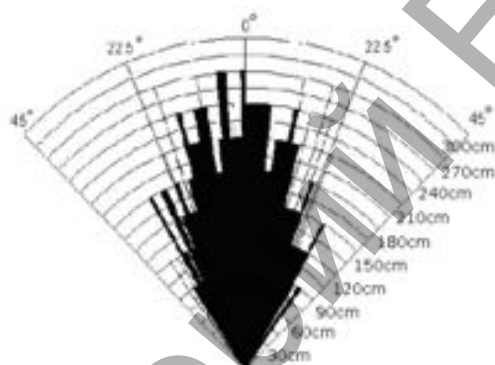


Рисунок 8 – Диаграмма направленности ультразвукового сенсора SEN136B5B

Положительные характеристики:

- 1) высокая точность измерений;
- 2) большая дистанция измерений;
- 3) широкий угол обзора, что позволяет обнаруживать небольшие препятствия.

Отрицательные характеристики:

- 1) широкий угол обзора не позволяет точно определить местоположение препятствия;
- 2) высокая вероятность сбоя в случае, если поверхность препятствия находится под углом к сенсору.

2.4. Одометры и гироскоп

Гироскоп и одометры используются в системе автономного позиционирования робота. Данные, получаемые с гироскопа, интегрируются и применяются при оценке угла поворота робота. На основе одометрических данных производится расчет пути пройденным правым и левым колесом, что позволяет оценивать текущую позицию робота.

Для корректировки траектории на основе гироскопа и одометров и управления двигателями была применена искусственная нейронная сеть [12]. Архитектура сети представлена на рисунке 9.

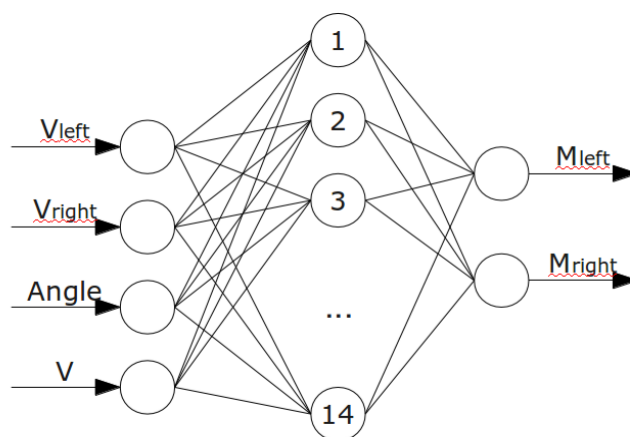


Рисунок 9 – Архитектура нейронной сети регулятора драйверов двигателя

Vleft and Vright скорости левого и правого колеса, которые вычисляются на основе данных, полученных с одометров. Angle вычисляется на основе данных гироскопа. V – это скорость, с которой робот должен ехать. Mleft и Mright – величина управляющего сигнала устанавливаемая на драйверах двигателей левого и правого колеса соответственно, которая задаётся от 0 до 255.

Обучающая выборка составлена на основе компьютерной модели из 60 образцов. В виде функции активации использовалась сигмоидная функция. Сеть обучалась на основе алгоритма обратного распространения ошибки. Среднеквадратичная ошибка обучения задана одна тысячная. Сеть успешно обучается в среднем за 1500 итераций. В результате сеть способна задавать необходимые параметры тока, подаваемого на двигатели для разных скоростей, и корректировать путь робота. Эксперименты моделирования показали, что робот корректирует свой путь и едет ровно по заданной траектории, с минимальной погрешностью. При задании скорости, которая не была учтена в обучающей выборке, робот успешно её выдерживает благодаря обобщающей способности нейронных сетей.

2.5. Объединение сенсорных данных

Основой для объединения данных с сенсоров является вероятностная карта из ячеек. Каждая ячейка представляет собой квадратный сантиметр реального пространства и может принимать значения от 0 до 255. Эти значения показывают степень доверия, что в этой клетке находится препятствие. Основные стадии алгоритма построения карты:

1. Инициализация карты (карта помещения переносится в вероятностную карту робота, задаётся положение и направление робота).
2. Считываются данные ультразвуковых датчиков. Уменьшаются значения ячеек до препятствия, которые покрывает ультразвуковой луч.
3. Считываются инфракрасные сенсоры. Увеличиваются значения ячеек, в которых найдены препятствия.
4. Если произведена калибровка положения робота, обновляется позиция робота и все значения карты уменьшаются на процент погрешности предыдущих координат.

3. Алгоритмы навигации и планирования пути

Программное обеспечение робота состоит из трех связанных подсистем: реактивного поведения, исполнительской и подсистемы планирования движения [13, 14].

Подсистема реактивного поведения отвечает за логику поведения низкого уровня, основанную на обработке данных, поступающих от сенсоров. С помощью этой системы реализуются рефлексыв поведения, позволяющие избегать столкновения робота с препятствиями.

Подсистема исполнения отвечает за логику управления среднего уровня. С помощью этой системы выбранные модели поведения робота превращаются в управляющий набор команд. Также при помощи этой системы осуществляется контроль над исполнением текущей команды, переданной на управляющие механизмы робота, и отвечает за сбор и обработку сенсорных данных (рис. 10).

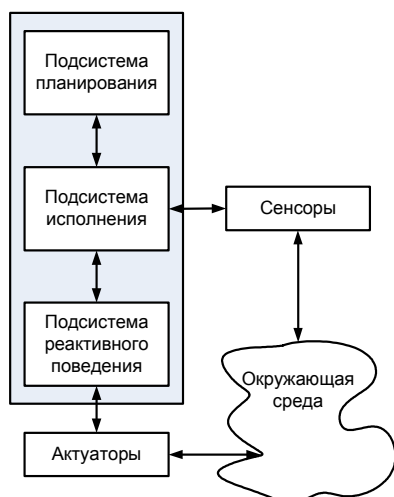


Рисунок 10 – Архитектура программного обеспечения робота-гида

Подсистема планирования отвечает за логику управления высокого уровня. Опираясь на накопленные данные об окружающей среде и анализируя текущее состояние сенсорной системы, подсистема планирования формирует модель поведения робота. Так же один из модулей подсистемы отвечает за коммуникацию с пользователем через графический интерфейс.

Описанная архитектура системы легко разделяется на низкоуровневую и высокоуровневую составляющие. Для реализации низкоуровневой системы используются 8-битные микроконтроллеры, а для высокоуровневой реализации используется портативный ПК. Такое разделение системы позволяет повысить переносимость кода на различные платформы.

Планирование пути роботом производится на основе топологической карты, построенной на основе вероятностной карты ячеек. Такая карта представлена неориентированным графом. Узлы привязаны к вероятностной карте местности и обладают координатами. Таким образом, существует связь между геометрической картой и топологической картой. Рёбра графа показывают возможность перемещения из одного узла в другой. В процессе работы робот способен добавлять и удалять рёбра, в зависимости от изменения данных на вероятностной карте. Узлы графа подразделяются на 4 типа:

- 1) основные;
- 2) вспомогательные;
- 3) калибровки;
- 4) блокирующие.

Основные узлы расположены рядом с важными объектами в помещении: зоной парковки робота, зоной калибровки, экспонатами. Блокирующие узлы не позволяют роботу заехать на заданную территорию. Основные и блокирующие узлы инициализирует оператор с помощью программного обеспечения управления перед запуском робота. Графическое отображение графа показано на рис. 11.

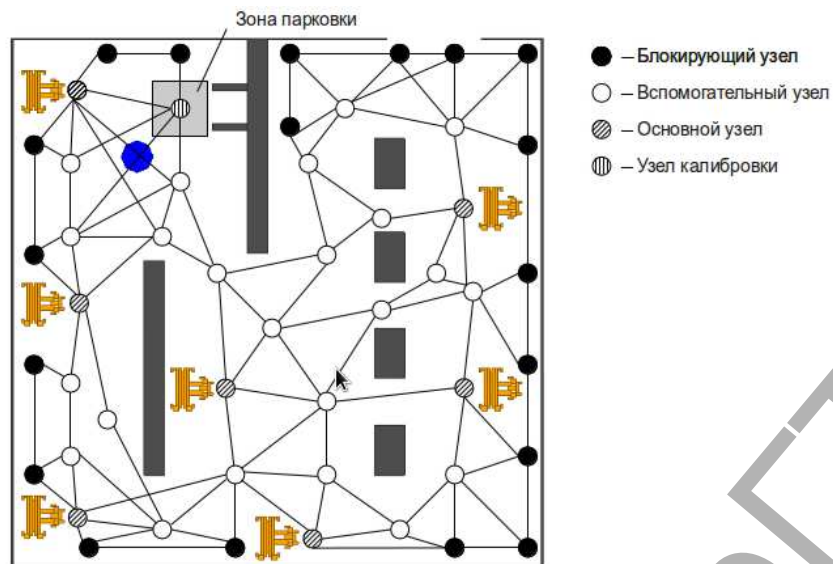


Рисунок 11 – Графическое отображение графа на модели окружающей среды

Вспомогательные узлы и узлы калибровки робот добавляет в процессе работы. Вспомогательные узлы заполняют пустое пространство и позволяют ему выбирать оптимальный путь для достижения цели. Если в узле больше нет необходимости, он удаляется.

Узлы калибровки заложены в систему. Их использование предполагает внедрение SLAM алгоритма, частично описанного в четвертом разделе. Такие узлы обладают дополнительной информацией о необходимом направлении робота и направления калибровки для сенсоров. Таким образом, становится возможным уточнение положения робота относительно стен и больших объектов на карте.

4. Результаты моделирования

Требование к навигации робота – обеспечить точное перемещение в течение 60 с. Для скорости в 1,5 м/с пройденное расстояние будет представлять собой 90 м, что позволяет применить такую систему для навигации в закрытом помещении. Допустимая ошибка задана 40-ка сантиметрами для 60 с. движения.

Задавая погрешности в моделировании для сенсорных данных, мы сравнили результаты передвижения с и без предварительной обработки. Используемые алгоритмы позволили уменьшить погрешности до 1-4%.

5. Заключение и будущие работы

Основной задачей проекта робота-гида при обработке данных сенсоров было подготовить данные для использования высокоуровневых алгоритмов, таких как SLAM. Разработанный алгоритм будет использовать три способа калибровки:

- 1) относительно начального положения робота;
- 2) относительно ключевых точек с известными координатами;
- 3) относительно экспонатов.

Такой подход должен решить проблему длительной навигации робота.

Для использования алгоритмов типа SLAM необходимо решать основные пять проблем локализации:

- 1) систематическая ошибка всех сенсоров;
- 2) выбор размерности и вида карты;

- 3) сопоставление данных сенсоров и данных карты;
- 4) отображение динамических объектов на карте;
- 5) планирование пути на основе карты.

Первые три проблемы решены на данном этапе разработки. Дальнейшая разработка гида планирует внедрение высокоуровневых алгоритмов и интеллектуальной системы управления роботом.

Признательность

Эта работа выполнена при поддержке гранта Министерства образования Республики Беларусь №11/116 2011 года и гранта Белорусского Республиканского фонда фундаментальных исследований F11LIT-003.

Литература

1. <http://www.aeato.com/users-manual/sony-aibo-entertainment-robot-user-manual-guide.html/> Operating instruction ERS-210. Entertainment robot AIBO.
2. http://www.hitachi.com/rd/research/robotics/emiew2_01.html EMIEW2.
3. <http://www.honda-p3.com/robotics-research/shop-robots-toshiba.html>.
4. http://wn.com/FURO_Future_Robot.
5. <http://www.wired.co.uk/magazine/archive/2011/08/start/friendly-bank-bots>.
6. http://www.toyota.co.jp/en/about_toyota/facility/toyota_kaikan/index.html.
7. Chong Seng, Kleeman Lindsay: Mobile-Robot Map Building from an Advanced Sonar Array and Accurate Odometry, the International Journal of Robotics Research, January 1999. – Vol. 18, № 1. – P. 20-36.
8. Kam M., Xiaoxun Zhu, Kalata P. Sensor fusion for mobile robot navigation, proceedings of the IEEE, January 1997.
9. Kleeman Lindsay, Kuc Roman. Mobile Robot Sonar for Target Localization and Classification, the International Journal of Robotics Research, August 1995. – Vol. 14, № 4. – P. 295-318.
10. <http://pdf1.alldatasheet.com/datasheet-pdf/view/412635/SHARP/GP2Y0A21YK0F.html>.
11. <http://www.electan.com/datasheets/GP2Y0A21YK.pdf>.
12. Головкин, В.А. Нейронные сети: обучение, организация и применение: учебное пособие. Кн. 4 / Под общ. ред. А.И. Галушкина. – М.: ИПРЖР, 2001.
13. Arras K.O., Philippsen R., Tomatis N., de Battista M., Schilt M., Siegart R. A navigation framework for multiple mobile robots and its application at the Expo.02 exhibition, proceedings of ICRA '03. IEEE International Conference on Robotics and Automation, 2003.
14. Bonasso R.P., Firby R.J., Gat E., Kortenkamp D., Miller D., Slack M. Experiences with an Architecture for Intelligent, Reactive Agents, journal of Experimental and Theoretical Artificial Intelligence 9(2), 1997.

Кабыш А.С.

*Брестский государственный технический университет
anton.kabysh@gmail.com*

ПРОЕКТИРОВАНИЕ SWARM-СИСТЕМ

Введение

Роевой (или стайный) интеллект (англ. *swarm intelligence*) – это раздел исследований моделей коллективного поведения в системах, состоящих из простых индивидуумов, локально взаимодействующих друг с другом и внешней

средой, поведение которых принимает формы децентрализованного управления и самоорганизации.

Примерами таких систем являются колонии муравьёв и термитов, косяки рыб, стаи птиц, стадное поведение животных, а также уже стаи виртуальных частиц или муравьёв в компьютерных программах для решения задач оптимизации. Во всех примерах итоговая самоорганизация поведения формируется за счет поведения индивидов на основе их локальных данных, не требуя наличия глобальной стратегии поведения системы и окончательного результата.

Исследования аналогов живого мира показывают [1], что самоорганизация строится на основе двух основных противоположных сил – *притяжение и отталкивание*. Эти силы можно трактовать как положительную или отрицательную обратную связь, учитываемую при координации поведения между индивидами в стае. Например, если индивид попадает в скопление агентов с большой плотностью, его поведение диктует ему покинуть их (отталкивание), однако если агент остается один или плотность агентов невысока, они стремятся друг к другу (притяжение). Положительная обратная связь между индивидами приводит к эффекту самоорганизации всей системы и возникновения в ней коллективного поведения; в дальнейшем самоорганизация компенсируется отрицательной обратной связью. Балансируя между циклами положительной и отрицательной обратной связи, стая находится в состоянии динамического равновесия.

В моделях роевого интеллекта, по аналогии с биологическими системами, взаимодействие между индивидами носит ограниченный характер в виде направленного или ненаправленного обмена сигналами.

- *Ограниченное взаимодействие* между индивидами обозначает, что результат поведения каждого зависит лишь от его собственного ограниченного восприятия мира. Например, если индивид определил ведомого, то он следует за ним, если нет – то выполняет случайный поиск, пока не найдет. Другим примером такого поведения является «социальное облегчение» [2] – индивид начинает выполнять те же действия, что и другие индивиды. В крайнем случае, индивиды могут даже и не обмениваться сигналами, целиком и полностью формируя коллективное поведение за счет собственного восприятия мира.

- *Ненаправленный сигнал*, так же называемый следом, индивид сообщает во внешнюю среду, без целеуказания того, кто должен получить этот сигнал. Другой индивид, обнаружив след во внешней среде, либо соглашается следовать ему (положительная обратная связь), либо нет (отталкивание). Частный случай ненаправленного обмена сигналами называется *stigmergy* – обмен сообщениями путем модификации внешней среды. Примером стигметрии являются следы феромона, оставляемые муравьями в процессе поиска оптимального пути. Итоговый найденный путь будет иметь максимум феромона, что для муравья служит сообщением – «этот путь предпочтительнее, чем другие».

- Направленный обмен сигналами обозначает направленное сообщение, способное изменить поведение, в большей или меньшей степени поведение получателей этого сообщения. Например, крик об опасности является направленным сообщением.

Исследователями найдено значительное количество способов взаимодействия между индивидами в биологических социальных системах [2], однако еще не все из них были исследованы применительно к системам искусственного интеллекта.

1. Свойства swarm-систем

Систему, состоящую из простых, автономных агентов, локально взаимодействующих друг с другом и внешней средой, поведение которых принимает формы децентрализованного управления и самоорганизации, назовем *много-агентной swarm-системой* или просто *swarm-системой*. Агента в swarm-системе в дальнейшем будем называть *swarm-агентом*.

Суммируем отличительные характеристики исследований swarm-систем, наблюдаемые в биологических аналогах [1, 3, 4]:

- Исследование должно быть релевантным задаче координации swarm-агентов.
- Swarm-агенты, составляющие стаю, должны быть гомогенными.
- Коммуникация между агентами должна носить только локальный характер.
- Swarm-агент самостоятельно принимает решение о своих действиях, опираясь на доступную ему информацию, формируя распределённое принятие решений агентами (decentralized decision making).
- Механизм координации должен масштабироваться в широком диапазоне размера стаи.
- Если исследование касается стайной робототехники, то swarm-агенты должны быть физическими роботами, способными взаимодействовать друг с другом и с внешней средой. Требования, предъявляемые к swarm-роботам, более подробно описаны ниже.

Swarm-системы, построенные по указанным принципам, обладают *устойчивостью, гибкостью и масштабированием* на системном уровне [3,5]. *Устойчивость* обеспечивается не зависимостью работы системы от конкретных агентов, а только от их плотности; отказ одного агента может быть немедленно компенсирован другим агентом. *Гибкость* означает наличие и использование различных коллективных стратегий в достижении цели. *Масштабирование* означает, что та же задача может быть решена как большим, так и небольшим числом агентов при применении тех же правил; правила не должны зависеть от количества агентов.

Благодаря наличию указанных свойств, swarm-системы являются привлекательной областью исследований. Исследуются как сами алгоритмы формирова-

ния коллективного поведения и координации в swarm-системах, так и анализируются свойства этих алгоритмов по указанным характеристикам.

2. Исследования в области Swarm-Intelligence

Наблюдаемые в природе модели коллективной организации вдохновили исследователей на разработку нового «стадного» подхода к решению существующих проблем. На основе swarm-подхода было разработано несколько алгоритмов оптимизации, получивших широкое распространение в последние годы [4]. Основными семействами алгоритмов являются *муравьиный алгоритм (ant colony optimization, ACO)* и *алгоритм роя частиц (particle swarm optimization, PSO)*. Другой областью swarm-исследований, получившей широкое распространение и признание, является стайная робототехника [1, 3]. Первые предложенные модели стадного поведения [7] нашли свое применение в кинематографе и компьютерных играх для отрисовки коллективного поведения [1]. На основе стадных принципов был спроектирован ряд протоколов маршрутизации следующего поколения [3].

Стадное поведение проявляется при решении следующих задач [1-6]:

1. *Агрегация* – самоорганизующееся поведение, приводящее к формированию кластера агентов и их группированию.

2. *Дисперсия* – противоположность агрегации: необходимо добиться равномерного распределения агентов по внешней среде, максимизируя область покрытия. Развитие этой задачи может включать в себя распределенное исследование и мониторинг внешней среды.

3. *Стадное поведение* – формирование стаи из распределенной группы агентов и демонстрация стадного поведения, при котором наблюдается эффект самоорганизации поведения.

4. *Коллективный поиск и доставка ресурсов (фуражирование)* – богатая область исследований, посвященная разработке оптимальных стратегий поиска и доставки ресурсов на базу стай агентов.

5. *Самоорганизация* – создание коллективом агентов структур посредством «физического» единения и кооперации. Примерами задач, где требуется самоорганизация агентов, являются *коллективное передвижение, кооперативный транспорт и формирование паттернов*.

6. Отдельным классом задач стоят задачи самоконфигурации и реконфигурации. В этих задачах swarm-система представляет собой множество соединенных в одно целое модулей (unit). Исследуются способности данных систем и перестройке собственной структуры, изменению формы и решению проблем посредством изменения собственной формы.

7. В последние годы развитие получили исследования стадного поведения гетерогенных swarm-систем, или нескольких стай (multi-swarm). Например, в работе [10] исследуется коллективное поведение и кооперация наземных и воз-

душных роботов в задаче поиска и добывания «еды» в неизвестной среде, напоминающей лабиринт. Воздушные агенты решают задачу кооперативного исследования и навигации для наземных, а наземные – доставку ресурсов.

Данные задачи решаются как численными способами, моделированием, так и постановкой экспериментов над группами роботов, формирующих стаю.

3. Swarm Robotics

Роевая (Стайная) робототехника или *Swarm Robotics* – это применение идей Swarm Intelligence к физическому коллективу простых роботов. Вместо разработки сложных роботов и сложных алгоритмов поведения, исследования в области swarm robotics фокусируются на решении проблем посредством эмерджентных эффектов коллективного поведения, основанного на применении простых правил взаимодействия к коллективу роботов [1]. Традиционно, в качестве роботов, составляющих стаю, используются простые, недорогие, гомогенные роботы с возможностью к коммуникации друг с другом [5]. Роботы должны быть просты в том смысле, что возможности одного индивида должны быть ограничены и недостаточны для решения поставленной задачи. Коллективное поведение и достижение некоторой цели (например, сохранение формации) может быть тривиальной задачей при моделировании, но с учетом ограниченных возможностей малых роботов задача координации их поведения становится сложным вызовом [8].

При стайном взаимодействии канал связи между роботами либо отсутствует вовсе, либо сильно ограничен [5, 6]. Взаимодействия между роботами (обмен сигналами) должны носить только локальный характер. Роботы принимают решение о своих дальнейших действиях на основе сенсорных данных об окружающей среде. При этом информация о состоянии других роботов группы может не поступать вовсе либо поступать посредством той же сенсорной информации. Робот участвует в информационном обмене с другими роботами в стае, находящимися в пределах зоны видимости робота, ограниченной радиусом L .

Область исследований Swarm Robotics включает как исследования алгоритмов коллективного стайного поведения для решения различных задач, так и вопросы проектирования роботов для таких алгоритмов. Как правило, любые исследования по swarm-робототехнике используют предварительное моделирование для быстрого доказательства работы и оценке свойств алгоритма коллективного поведения.

4. Проектирование swarm-системы роботов

Задача разработки алгоритмов управления swarm-роботами для создания swarm-системы является непростой задачей, которая формируется следующим образом: *как сформировать стаю и создать стайное поведение, имея все ограничения, присущие swarm-роботу*. При проектировании swarm-систем можно выделить два ключевых фактора:

1. **Swarm-коммуникация** определяет конкретный способ взаимодействия между swarm-агентами, с целью получения локальных данных о других аген-

тах, для последующего использования этих данных при принятии решений о конкретной стратегии поведения в swarm-алгоритме. Способ коммуникации должен быть единообразен для всех swarm-агентов в стае. Например, способ коммуникации может определить дистанцию до соседа, его ориентацию и т.д. Примеры различных способов коммуникации рассмотрены ниже.

2. **Swarm-алгоритм** определяет стратегии поведения на каждом swarm-агенте в зависимости от получаемых данных коммуникации. Когда взаимодействие между роботами налажено, требуется задать правила поведения, единообразные для каждого робота, совместное выполнение которых всеми роботами приведет к ожидаемым коллективным эффектам и решению поставленной цели.

Способ коммуникации роботов на всем коллективе должен быть стандартным, единообразным. Роботы могут быть гетерогенными, но на уровне коммуникации их идентичность должна сохраняться. Swarm-алгоритм может быть либо спроектирован вручную и закодирован в робота, либо может быть использована какая-либо обобщенная методология, результат работы которой содержит итоговые стратегии поведения. В качестве такой методологии может выступать: эволюционный алгоритм, нейронная сеть, генетическое программирование и т.д.

Следовательно, стайе можно охарактеризовать группу роботов гомогенных на уровнях коммуникации и используемого алгоритма коллективного поведения.

5. Способы коммуникации swarm-роботов

В простейшем случае, для коммуникации роботы используют свои сенсоры. В зависимости от типа сенсоров и их конфигурации изменяется вид получаемой информации и диапазон восприятия мира роботом. В большинстве случаев робот оснащается дальномерами или камерой. Вне зависимости от вида сенсорных данных, робот должен получать следующие характеристики из внешней среды:

- Наличие соседей в непосредственной близости.
- По каждому соседу необходима следующая информация:
 - Положение соседа относительно робота.
 - Ориентация соседа относительно робота.
 - Ведущий или нет (опционально).
 - Другие данные, специфичные для конкретного алгоритма.

Каждый алгоритм стайного поведения предъявляет свои требования к способу коммуникации роботов. Например, в работе [8] разработан стайный алгоритм роботов только на основе 3-х дальномеров, описанный и предъявляющий самые минимальные требования к роботу. Роботы, оснащенные видеосистемой, могут получить необходимый минимум информации о находящихся поблизости роботах группы, если все роботы группы оборудованы цветовыми индикаторами. Например, в [12] роботы s-bot используют светодиодные паттерны для коммуникации друг с другом, чтобы согласовать направление коллективного движения в задаче кооперативного транспорта. Каждый робот имеет триангулярный цветовой паттерн, составленный из светодиодов, где красный цвет указывает направление движения робота. Каждый робот имеет всенаправленную

камеру с оперативным расстоянием 60 см. На работе используется программное обеспечение, позволяющее определить по «цвету» направление движения соседей в области наблюдения камеры.

Вместо световых индикаторов, могут быть использованы другие паттерны или QR-коды. Наиболее распространенными являются видеокamеры в различных диапазонах спектра, акустические и вибрационные датчики, газовые анализаторы, лазерные дальномеры, широкополосные приемники и т.д.

Для коммуникации роботы также могут использовать внешнюю среду. Для этого и роботы, и внешняя среда должны быть специальным образом подготовлены. В [11] приведены эксперименты, когда стая роботов повторяла классический алгоритм муравьиной колонии. Для этого «передатчик феромона» в виде кругового инфракрасного излучателя и приемника в виде отдельного модуля был установлен прямо на роботе. Чем сильнее был суммарный сигнал, тем сильнее след феромона. Также во внешней среде могут находиться светодиоды, радиометки и другие источники информации для робота.

6. Swarm-алгоритмы

В классической работе К. Рейнольдса [7], исследовавшего поведение стай птиц и косяков рыб, был предложен стайный алгоритм, в котором самоорганизация стайного поведения – результат суперпозиции во времени и пространстве следующих трёх принципов, применимых по каждому агенту [7-9]:

1. **Отгалкивание**, или дисперсия (*collision avoidance, separation*) – соблюдать дистанцию d_{min} до препятствий и других агентов.

2. **Выравнивание** (скорости, ориентации) стаи (*alignment*) – придерживаться средней скорости движения соседей по стае в наблюдаемой области видимости L или выровнять свою ориентацию по ориентации соседних роботов.

3. **Притяжение** (*flock centering, cohesion*) – стремиться занять положение, близкое к центру тяжести, по всем наблюдаемым агентам в окрестности L .

4. **Поведение по умолчанию** выполняется, если наблюдаемых агентов в поле видимости нет. Может принимать самые различные формы.

а. **Блуждание** – агент выполняет случайное блуждание, пока другие агенты не будут найдены.

б. **Ожидание**, пока в области видимости появится какой-либо агент.

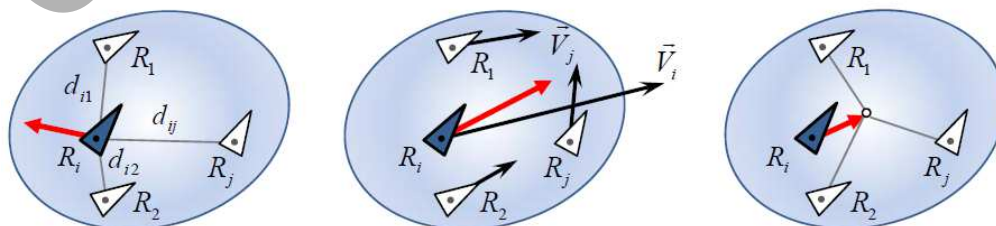


Рисунок 1 – Принципы самоорганизации движения swarm-агентов в модели К. Рейнольдса [7] (адаптировано из [9])

На основе описанных принципов (1) – (3) можно сформировать огромное множество стереотипов стратегий стайного поведения [7-9]:

Индивидуальное поведение	Коллективное поведение
<ul style="list-style-type: none"> • Приближаться к объекту. • Удаляться от объекта. • Преследовать объект. • Убегать от объекта. • Следовать за лидером. • «Блуждать» (бродить) в заданном районе случайным образом. • Преследовать объект с заданным смещением. • Прибыть в заданную точку. • Избегать столкновений (с подвижными или неподвижными объектами). • Следовать по сигналу (следу). • Следовать по траектории. 	<ul style="list-style-type: none"> • Сформировать стаю при заданных параметрах: близости/скорости/притяжения. • Агрегация – собраться в заданной точке. Дисперсия – рассредоточиться в заданном районе. • Блокировать передвижение объекта. • Искать объект. • Охранять территорию или периметр. • Защищать периметр или объект. • Сформировать хаос • Сформировать формацию. • Групповое передвижение. • Другое

Исследования в области алгоритмов стайного поведения включают различные реализации описанных принципов, их взаимосвязь, различные комбинации, частичные реализации и т.д. Например, в работе [13] предложен способ реализации данных принципов на основе физических законов, получивший название «искусственная физика» (artificial physics). В каждый момент времени рассчитывается вектор силы \mathbf{f} , действующий на робота, определяемый тремя составляющими:

- \mathbf{p} – проксимальная сила, действующая на робота. *Проксимальное поведение* [14] (proximal control) [1] означает, что робот воспринимает относительные позиции (дальность и ориентацию) своих соседей в своей области видимости и корректирует свое поведение по принципам агрегации и дисперсии;

- \mathbf{h} – вектор выравнивания ориентации робота вычисляется на основе суммарных показаний ориентаций соседних роботов;

- \mathbf{g} – вектор направления цели. Вектор цели доступен для одних роботов (ведущие) и недоступен для других.

Итоговое значение вектора силы, действующей на робота, определяется как $\mathbf{f} = \alpha \mathbf{p} + \beta \mathbf{h} + \gamma \mathbf{g}$, где α , β , γ – это коэффициенты относительного вклада различных сил, задаваемые эмпирическим путем.

7. Управление swarm-роботами

Конкретный способ управления роботами зависит от имеющихся возможностей роботов и инфраструктуры. Можно выделить два принципиальных подхода – внешнее и внутреннее управление.

Подход на основе внутреннего управления заключается в том, что робот использует только показания своих датчиков для выбора текущего правила поведения. Робот также может опрашивать собственные устройства связи и прини-

мать/отсылать сообщения другим роботам. При этом подходе все вычисления происходят на роботе. Данный вид управления по отношению к swarm-системе называется *микрочитроль*.

Преимуществом данного подхода является то, что робот воплощает классическую модель автономного swarm-агента. К недостаткам можно отнести общую сложность решения, требуемое большое количество сенсоров и электроники, а также только ограниченное количество swarm-алгоритмов и методов коммуникации, которые могут быть реализованы при помощи встроенных, ограниченных возможностей робота.

Альтернативный подход состоит в разработке коммуникационной инфраструктуры для управления стаей роботов, внешней по отношению к самим роботам. В этом случае робот может отсылать показания своих сенсоров на удаленный сервер, а в ответ получает команды для выполнения. При данном подходе swarm-агент управляется внешним источником, но с соблюдением всех правил проектирования и поведения swarm-систем. Другим примером может быть система позиционирования, на основании подвешенной под потолком камеры, которая для каждого робота определяет его местоположение и отношение к другим роботам в стае. В этом случае стайный робот даже может обходиться без датчиков. Подход, при котором управление выполняется на уровне всей swarm-системы, получил название *макрочитроль*. Недостатком данного подхода является необходимость создавать внешнюю инфраструктуру для управления роботами. Стоимость полученной инфраструктуры может быть сопоставима со стоимостью самих роботов.

Заклучение

В данной статье был дан краткий обзор областей исследований Swarm Intelligence и Swarm Robotics с описанием отличительных характеристик исследований в данной области. Адаптивные робототехнические системы, реализующие стайную модель поведения, способны автономно и гибко изменять параметры поведения – в зависимости от выполняемой работы, условий внешней среды, технического состояния роботов – с минимальными затратами ресурсов.

При проектировании swarm-систем стоит уделить особое внимание вопросам коммуникации, алгоритму стайного поведения, способу управления роботами и возможной симуляции стайного поведения до начала экспериментов с физическими роботами. Способ коммуникации определяет, каким образом робот узнает необходимую ему информацию о соседях. В основе стайного поведения лежат принципы этологии, заимствованные из животного мира. Большинство алгоритмов стайного поведения основаны на описании законов притяжения, отталкивания и выравнивания между соседними роботами в стае. Целенаправленное, внешне «интеллектуальное» поведение группы формируется спонтанно, в результате суперпозиции актов местного взаимодействия роботов и их реакции на изменения в локальном пространстве.

Моделирование стайного поведения значительно упрощает задачу и может использоваться для концептуального доказательства формирования стайного поведения алгоритмом. Существует множество разработанных симуляторов

стайного поведения. Если эксперименты ставятся на реальных роботах, то необходимо обеспечить управление стаей роботов. Для этого либо все вычисления остаются на роботе, либо создается коммуникационная инфраструктура, управляющая всей стаей. Один и тот же стайный алгоритм может значительно отличаться в реализации для разных видов систем управления.

Литература

1. Floreano D, Mattiussi C. Bio-inspired artificial intelligence: theories methods and technologies. The MIT Press; 2008
2. Резникова, Ж.И. Сравнительный анализ различных форм социального обучения у животных // Журн. общ. биол. – 2004. – Т. 65. – № 2. С. 136-152.
3. Blum, C., Merkle, D. Swarm Intelligence-Introduction and Applications. Natural Computing Series. Springer. – 2008.
4. Christian Blum, Xiaodong Li. Swarm Intelligence in Optimization. In Christian, B. and Daniel, M. (eds) Swarm Intelligence: Introduction and Applications. Springer, Berlin Heidelberg, 2008. – P. 43-85.
5. Sahin, E., Girgin, S., Bayindir, L. and Turgut, A.E. Swarm Robotics. In Christian, B. and Daniel, M. (eds) Swarm Intelligence: Introduction and Applications. Springer. – Berlin, Heidelberg, 2008. – P. 87-101.
6. Иванов, Д.Я. Информационный обмен в больших группах роботов // Искусственный интеллект. – 2010. – № 4. – С. 513-522.
7. Reynolds, C.W. Flocks, Herds and Schools: a Distributed Behavioral Model // Computer Graphics. – 1987. – Vol. 21, № 4. – P. 25-34.
8. Möslinger, C., Schmickl, T., & Crailsheim, K. A minimalist flocking algorithm for swarm robots. In *Lecture notes in computer science. Proceedings of the 10th European conference on artificial life (ECAL)*. Berlin: Springer. – 2008.
9. Бурдун, И.Е. Метод самоорганизации стайного поведения малых мобильных роботов гражданского и специального назначения для арктических приложений / И.Е. Бурдун, А.Р. Бубин // Доклад на Всероссийской научно-технической конференции «Научное и техническое обеспечение исследований и освоения шельфа Северного Ледовитого океана», СибГУТИ, 9–13 августа 2010 года, г. Новосибирск.
10. F. Ducatelle, G. Di Caro, C. Pinciroli, and L. Gambardella. Self-organized cooperation between robotic swarms. *Swarm Intelligence*. – 2011. – Vol. 5, № 2. – P. 73-96.
11. Payton, D., Estkowski, R., Howard, M. Pheromone robotics and the logic of virtual pheromones. In , Sahin, E. and Spears, W. M., editors, *Proceedings of the Swarm Robotics Workshop*. – Germany, Heidelberg: Springer-Verlag. – 2005.
12. Campo, A., Nouyan, S., Birattari, M., Groß, R., Dorigo, M.: Negotiation of goal direction for cooperative transport. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stutzle, T. (eds.) ANTS 2006. LNCS. Heidelberg: Springer, 2006. – Vol. 4150. – P. 191-202.
13. Spears, W.M., Spears, D.F., Hamann, J.C., Heil, R.: Distributed, physics-based control of swarms of vehicles. *Autonomous Robots* 17, 137–162, 2004.
14. Ferrante, E., Turgut, A.E., Mathews, N., Birattari, M., Dorigo, M. Flocking in stationary and non-stationary environments: A novel communication strategy for heading alignment. In Schaefer, R., Cotta, C., Kolodziej, J., and Rudolph, G., editors, *Parallel Problem Solving from Nature. PPSN XI*, volume 6239 of *Lecture Notes in Computer Science*, pages 331-340. Springer. – 2010.

Андреюк А.А

Научный руководитель: ассистент кафедры ИИТ Кабыш А.С.

ПРОЕКТИРОВАНИЕ СИСТЕМЫ ПОСТРОЕНИЯ ТОПОЛОГИЧЕСКОГО ГРАФА ПО ПЛАНУ ПОМЕЩЕНИЯ

Введение

Для решения задачи навигации в неизвестной динамической среде системе навигации робота необходимо решить такие задачи, как создание глобальной карты помещения, собственное позиционирование в этой карте, разделение статических и динамических препятствий, оценка погрешности перемещения и т.д. В реальном мире исследователи имеют дело с частными случаями навигации и неизвестной динамической среде. Например, если эксперименты проводятся в здании, то неизвестная среда ограничена либо помещением, либо этажом здания, что значительно упрощает задачу навигации.

Если автономному мобильному роботу необходимо передвигаться в помещении, в дополнение к существующей системе навигации может использоваться информация о помещении, полученная из внешнего источника. Если роботу известен план помещения априори, то решение задачи навигации сводится к позиционированию робота на глобальной карте и её последовательное уточнение новыми статическими и динамическими препятствиями. Таким достаточно надежным и доступным источником карты помещения является план эвакуации помещения при пожаре.

План помещения, хоть и служит ценной информацией для робота, не может быть использован «как есть» в задаче навигации. Цель данной работы состоит в разработке системы, подготавливающей план эвакуации помещения для его использования роботом. Система принимает на вход подготовленную фотографию плана помещения, выполняет её обработку, фильтрацию и проводит распознавание паттернов, связанных с опорными контурами помещений, – стен, комнат и т.д., создавая виртуальное представление помещения, пригодное для использования роботом. В следующей фазе программа строит преобразование полученного виртуального пространства в его остов: топологический граф, или граф Вороного. Это преобразование (в дальнейшем) используется роботом для высокоуровневых операций навигации и прокладки пути. Для выполнения алгоритмов роботу должно быть указано его текущее местоположение (позиция) в координатах по карте помещения.

1. План помещения

Планы эвакуации представляют собой схемы, на которые нанесены контуры помещений, коридоров, лестничных маршей в зданиях и сооружениях. Планы эвакуации составляются по ГОСТам, исполнение которых контролируется органами пожарной безопасности. В зданиях планы эвакуации должны быть разработаны и вывешены в местах нахождения сотрудников, охраны и на каждом этаже у входов в лестничные клетки – поэтажные планы. Пример плана эвакуации показан на рисунке 1.

Преимущества использования плана эвакуации из помещения в качестве глобальной карты робота следующие:

- Он является стандартизированным, проконтролированным и оформленным по ГОСТам, т.е. достаточно достоверным планом помещения.

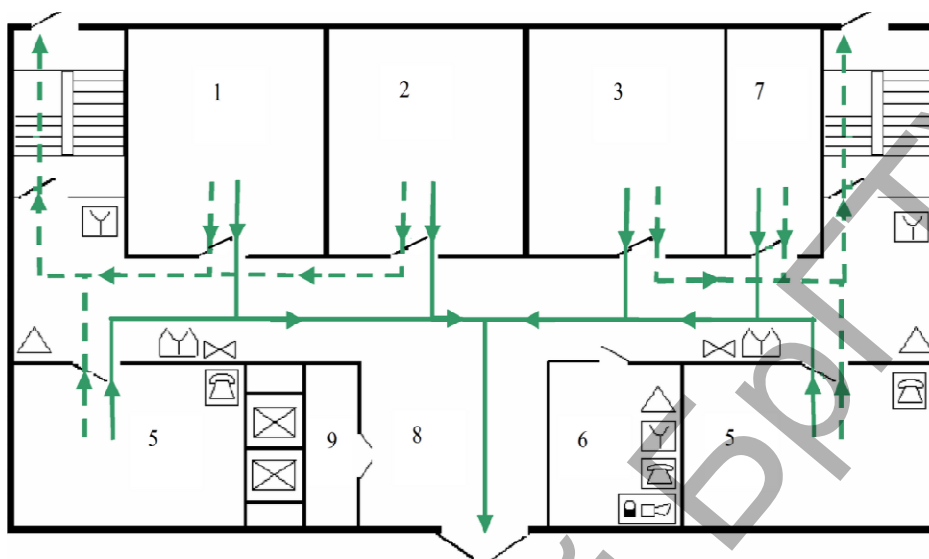


Рисунок 1 – Стандартный план эвакуации из помещения

- Общедоступен. План эвакуации имеется практически в каждом здании, его легко получить путем фотографирования. Если робот должен быстро освоить навигацию помещения в неизвестной среде, то план помещения может служить хорошей основой в качестве предопределенной глобальной карты среды.

- Показаны направления движения людей, которые так же может использовать робот в качестве опорного скелета помещения.

- Высокое качество проработки плана здания.

- Зная карту помещения, несложно указать роботу свою позицию на этой карте.

- Достаточно легко определить лестничные пролеты и прочие участки, опасные для робота.

Недостатки использования:

- Наличие потусторонних знаков и предметов на карте помещения. Проблема решается предварительной фильтрацией изображения.

- Возможное несоблюдение масштабов помещений на плане эвакуации и в реальном здании.

После обработки плана помещения и преобразования его в виртуальную карту программа строит топологический граф по карте.

2. Топологический граф

Граф, размещенный в некотором пространстве, без пересечения ребер называется топологическим. Такие графы применяют для решения различных задач (например, в электротехнике) и для определения оптимальных маршрутов.

После построения ребра графа представляют собой маршруты робота, веса на ребрах определяют расстояния. Существует несколько подходов к построению

графа, в данной работе будем использовать диаграммы Вороного и граф Вороного. Сам алгоритм построения использует важный принцип: путь робота должен пролегать максимально далеко от препятствий. Необходимость этого очевидна: путь перемещения должен учитывать возможные отклонения от идеального пути робота. Возможный пример построенного топологического графа в виде графа Вороного показан на рисунке 2.



Рисунок 2 – Граф Вороного для навигации по помещению

3. Схема работы системы

Предполагается следующая схема работы системы, частично отраженная на рисунке 3.



Рисунок 3 – Схема модулей системы

1. На камеру делается фотография плана помещения.
2. На фотографии отрезаются неинформативные части, такие как название, легенда, и т.д., не относящиеся напрямую к плану помещения.

3. **Начало.** Подготовленное изображение подается на вход системы.

На этом этапе может выполняться бинаризация изображения для его большей эффективности алгоритмов на последующих этапах, а также фильтрация нерелевантных данных, таких как спецметки огнетушителей и т.д.

4. **Модуль преобразования изображения** обрабатывает изображение с целью нахождения на нем контуров помещений и замкнутых пространств. На этом этапе необходимо выделить контуры помещений, коридоров и т.д. Необходимо выделить их геометрические центры, которые станут первыми опорными точками для топологического графа помещения. Цель работы этого модуля – сопоставить изображение и его виртуальное представление, где каждая точка имеет свои координаты относительно изображения; известно, достижима ли она для робота.

5. **Модуль построения графа** выполняет построение топологического графа на помещении на основании данных предыдущего модуля. Топологический

граф служит высокоуровневой проекцией помещения, на которой удобно выполнять операции прокладки пути через траверс графа.

6. **Модуль визуализации** отвечает за интерфейс к системе и представление полученных результатов для администраторов робота.

7. **Модуль работы с данными** позволяет изменять и корректировать данные, полученные из разбора изображения с использованием графического интерфейса. Сюда входят функции задания реальных размеров стен и помещений, масштабирование размеров отдельных помещений относительно друг друга, корректировка топологического графа помещения, задание начальной позиции робота и т.д.

Заключение

Обработанная карта может играть роль глобальной карты помещения, тогда как робот в процессе своего перемещения строит локальную карту на основании показаний своих датчиков и сверяет её с данными глобальной карты. Используя это виртуальное представление, робот в дальнейшем может решать задачу прокладки пути и собственного позиционирования в координатах этой виртуальной карты. Также использование плана помещения для навигации и позиционирования в реальном мире дает базу для использования мощных алгоритмов SLAM, способных решать задачу навигации и прокладки пути в реальном времени с высокой точностью.

Богущ А.С.

Научный руководитель: доцент кафедры ИИТ Дунец А.П.

КОЛЛЕКТИВНОЕ ПОВЕДЕНИЕ РОБОТОВ

Введение

Идея создания сложной системы, состоящей из множества сравнительно простых устройств, всегда была привлекательна. Ей отдавали дань философы и писатели-фантасты, математики и технические специалисты. Решение сложной задачи "простыми" с технической точки зрения средствами может привести к появлению "сверхорганизма", ознаменует явную ступень эволюции технических объектов.

Сложная система, состоящая из простых составляющих, должна обладать следующими особенностями:

- повышенная надежность (утрата части членов коллектива не влияет на работоспособность всей системы в целом);
- гибкость (способность системы к реконфигурации);
- потенциальная возможность развития и усложнения решаемых задач путем наращивания мощности коллектива.

Сферы, в которых могут использоваться механизмы коллективного поведения роботов, весьма многогранны:

- командная работа роботов по диагностике труднодоступных объектов,

- мониторинг окружающей среды,
- коллективное решение задач роботами-спасателями,
- разведка и рекогносцировка,
- охранные функции, патрулирование.

1. Примеры существующих систем

Открытый проект SwarmRobot. Проект посвящен созданию минироботов с линейными размерами до 3 см, которые могут организовывать масштабируемые коллективы из десятков и сотен роботов.

Проект Swarmanoid (Universite Libre de Bruxelles). Основная задача проекта – исследование поведения неоднородных коллективов роботов.

Эволюционирующие роботы. В 2009 г. в швейцарской лаборатории Laboratory of Intelligent Systems (политехническая школа, Лозанна) были проведены исследования в области "эволюции" роботов. Эволюционировал геном робота, определяющий поведение. В экспериментах группа из 10 роботов состязалась за пищу. Роботы – это колесные платформы (т.н. s-боты).

2. Направления в исследованиях коллективного поведения роботов

В названиях большинства существующих проектов фигурирует слово swarm-рой, стая. Это связано с тем, что исследования черпают свое вдохновение из мира насекомых, многие из них постоянно проводят аналогии с колониями муравьев, пчел и прочих стайных животных. Также реализуются и исследуются и другие модели:

- Формальные модели коллективного поведения.
- Многоагентные системы.
- Имитационные модели.
- Роевые алгоритмы.
- Эволюционные методы.

3. Роевые алгоритмы

Роевые алгоритмы и методы зачастую относят к методам т.н. роевого интеллекта. Задачей роевого интеллекта (РИ) является изучение и описание коллективного поведения децентрализованной самоорганизующейся системы, при этом методы РИ рассматриваются прежде всего как некие специфические механизмы поисковой оптимизации. Большинство алгоритмов РИ относится к классу метаэвристик.

Системы РИ состоят из множества агентов (многоагентная система), локально взаимодействующих между собой и с окружающей средой. Сами агенты обычно довольно просты, но все вместе, локально взаимодействуя, создают так называемый роевой интеллект. Ниже представлен перечень некоторых алгоритмов РИ, названия которых хорошо отражают суть подхода к решению задач:

- Муравьиный алгоритм.
- Метод роя частиц.

- Пчелиный алгоритм.
- Оптимизация передвижением бактерий.
- Стохастический диффузионный поиск.
- Алгоритм гравитационного поиска.
- Алгоритм капель воды.
- Светляковый алгоритм.

Характерно, что значительная часть роевых алгоритмов посвящена реализации моделей стайного поведения, прежде всего – стайному движению.

4. Элементная база

Создание системы взаимодействия роботов невозможно без наличия подходящей элементной базы.

Универсальные колесные платформы. Это наиболее распространенный способ реализации члена коллектива роботов. На рис. 2 представлены типичные представители этого класса – робот, выпускаемый фирмой POLOLU и т.н. s-бот.

Такие устройства обычно достаточно дороги и сложны. Например, упомянутые выше s-боты оснащены процессором Xscale с частотой 400 MHz, 64 МВ оперативной памяти и 32 МВ флеш-памяти, а также 12 PIC-микроконтроллерами для низкоуровневой обработки.

Специализированные мини-роботы. В рамках открытого проекта SwarmRobot предполагается создание множества минироботов размером порядка 3 см³. Если будут созданы надежные и простые технологии производства таких роботов, то возникнет реальная возможность появления действительно больших коллективов из нескольких сотен членов.

Микроробот, летающий на магнитных полях (университет Ватерлоо, Канада, 2009 г.). Микроробот MEMS для полета использует магнитное поле Земли. Микроробот весит 0,83 г. Он оснащен крошечными электромагнитами, создающими вокруг него трехмерное параболическое магнитное поле.

Миниатюрные роботы Kilobot (Группа исследования самоорганизующихся систем Гарвардского университета). Вместо колес робот оснащен двумя вибромоторами, позволяющими роботу двигаться вперед и совершать вращение. Отличительной особенностью таких роботов является их дешевизна (порядка \$14).

Заключение

В настоящее время в области коллективного поведения роботов остается очень много нерешенных вопросов как технического, так и теоретического характера. Среди основных открытых вопросов можно выделить следующие:

- Отсутствие единого подхода к проблеме коллективного поведения роботов.
- Применимость языковых (сценарных) подходов.
- Слабая элементная база.

Мешко Е.Э.

Научный руководитель: доцент кафедры ИИТ Дунец А. П.

ПРОСТОЙ ГУСЕНИЧНЫЙ РОБОТ СВОИМИ РУКАМИ

Введение

Одной из проблем при создании своего собственного робота является поиск и выбор необходимых комплектующих. Комплектующие можно разделить на 2 вида:

- Самодельные модули – изготовленные из подручных материалов.
- Готовые модули – базы с моторами, колёса, гусеницы, моторы-редукторы, разработанные специально для робототехники или для сложных и дорогих электронных игрушек на дистанционном управлении.

Эти модули ускоряют сборку робота в разы, более понятны для новичков и создают меньше проблем при отладке и тестировании механики.

Описание поиска и выбора комплектующих

Рассмотрим поиск и выбор готовых модулей для создания робота, которым можно будет управлять с помощью компьютера и смотреть за его перемещениями через видеокамеру. Материал статьи будет наиболее интересен тем, кто хотел бы построить своего робота, но не знает с чего начать.

Робота можно купить и готового, но:

- во-первых это не так интересно: сам процесс создания робота своими руками доставляет огромное удовольствие и дает базовые понятия, что и как устроено;
- во-вторых при покупке готового робота переделать его под свои интересы часто довольно сложно, если вообще возможно;
- в-третьих, самостоятельно сделать робота гораздо дешевле. Для постройки робота будут использоваться максимально дешевые, но готовые узлы, потому что для новичков сборка с нуля будет сложна и рискует надоесть и затянуться на пару недель.

Робот не будет самостоятельно принимать решения (это управляемое с ПК шасси, которое выполняет команды и передает видеосигнал). Однако доработать его до самостоятельного робота не составит никакого труда – нужно лишь добавить сенсоры и дописать логику бортового компьютера. Платформа на это рассчитана.

Для гусеничного шасси понадобится:

1. Двухмоторная коробка с редуктором – Tamiya Twin Motor Gearbox (TAM70097)
2. Гусеницы и катки с осями – Tamiya Track and Wheel Set (TAM70100)
3. Платформы и крепеж – Tamiya Universal Plate Set (2) (TAM70157)

Все эти детали можно заказать в официальном магазине Tamiya. Также можно найти эти запчасти на rololu.com. Вместо платформы с отверстиями вполне можно использовать кусок фанеры, а для крепежа – металлический конструктор, старый, советского образца, или приобрести новые модели. Самое главное – это гусеницы и мотор-редукторы.

Бортовая электроника:

1. Управляющая плата Arduino / Freeduino – мозг нашего робота.

2. Драйвера двигателей для управления моторами.
3. SmartBluetooth модуль. Понадобится для связи с ПК или ноутбуком.
4. Bluetooth dongle – если управлять с ПК и Bluetooth интерфейса в нем нет.
5. Чтобы видеть изображение от первого лица, понадобится миниатюрная видеочка, радиопередатчик (трансмисмиттер) и приемник (ресивер).

Камера

Качество изображения и другие возможности камеры будут зависеть от цены. Нам также понадобится питание для Arduino и моторов. Для этого можно воспользоваться обычными АА батарейками или 9В батарейкой типа «Крона». Но предпочтение было отдано компактным и более удобным LiPoly аккумуляторам: при заказе из Китая все компоненты обойдутся гораздо дешевле, чем их покупка в Белоруссии или других странах, компоненты будут получены примерно через 2-3 недели.

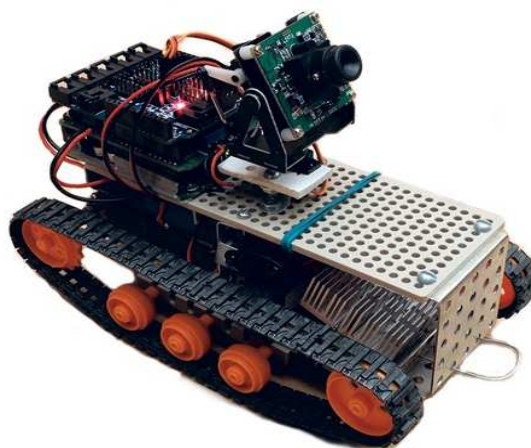
Шасси

Комплект Tamiya 70100 идет с подробной инструкцией. Аккуратно вырезаем кусачками или канцелярским ножом резиновые гусеницы и катки, отделяя от литников, собираем и крепим с помощью винтов к фанерной площадке или к Tamiya universal plate set.

Затем собираем коробку моторов с редукторами TAM70097. У этого редуктора есть два варианта передаточных чисел. 58:1 – робот будет ездить очень быстро, но моторчики на малых оборотах после того, как на платформу будет установлена электроника, могут вообще не поехать. Редуктор будет собираться в варианте 201:1, так решиться проблема с возможной перегрузкой. Для проверки работоспособности достаточно подать питание от двух батареек на моторчики, просто коснувшись проводами контактов мотора. Теперь нужно подпаять по паре проводов к моторчикам. Сантиметров по 10-15 будет достаточно.

После этого с помощью деталей от конструктора можно закрепить вторую площадку для крепления электроники. Это будет делаться для того, чтобы облегчить доступ к электронике и разделить ее с моторами и батареями. На второй этаж установится управляющая плата, Bluetooth – модуль, а также драйвер двигателей. Полярность подключения моторов значения пока не имеет, ее потом можно поменять или задать программно.

Arduino желательно крепить USB портом к краю платформы, чтобы удобнее потом было подключать кабель и программировать, не снимая с робота. Работы всего на пару часов, если делать не торопясь. В итоге получим примерно следующее:



на

Заключение

Собирать все это довольно увлекательно и совсем не так сложно, как может показаться на первый взгляд. В дальнейшем данная база планируется использоваться в проекте. Задача робота будет сводиться к прокладыванию кабелей над натяжными и навесными потолками.

Нефедьев А.Ю.

Научный руководитель: Дунец А.П.

Брестский государственный технический университет

АДАПТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ АВТОТРАНСПОРТНЫМИ СРЕДСТВАМИ НА ПЕРЕКРЁСТКАХ ГОРОДСКИХ УЛИЦ, ОБОРУДОВАННЫХ СВЕТОФОРАМИ

Введение

В последние годы в странах Западной Европы, Японии, США и др. широко используются адаптивные (гибкие) системы управления движением автотранспортных средств (далее АТС) на регулируемых перекрестках городских улиц. Это позволяет своевременно контролировать интенсивность движения АТС на перекрестках и выбирать наиболее подходящий режим работы светофоров (переключение с красного цвета на зеленый и наоборот).

В результате сокращается количество дорожных «пробок» АТС на перекрестках и увеличивается их пропускная способность.

1. Адаптивная система управления движением АТС на городских перекрестках улиц (АСУА «Брест-1»)

АСУА «Брест-1» подключается к дорожному контроллеру «Думка» (с функцией «Табло вызова пешехода»), установленному на перекрестке и жестко регулирующему длительность фаз переключения светофора.

Схема установки четырех шкафов управления АСУА «Брест-1» с радиоантеннами на перекрестке приводится на рисунке 1.

Алгоритм работы основан на поиске разрывов в транспортном потоке АТС с помощью инфракрасных датчиков.

Взаимодействие работы адаптивной системы и дорожного контроллера «Думка» осуществляется посредством канала ТВП (табло вызывное пешеходное), что обеспечивает бесконфликтное, надежное подключение к работающему дорожному контроллеру и не нарушает принципов его работы.

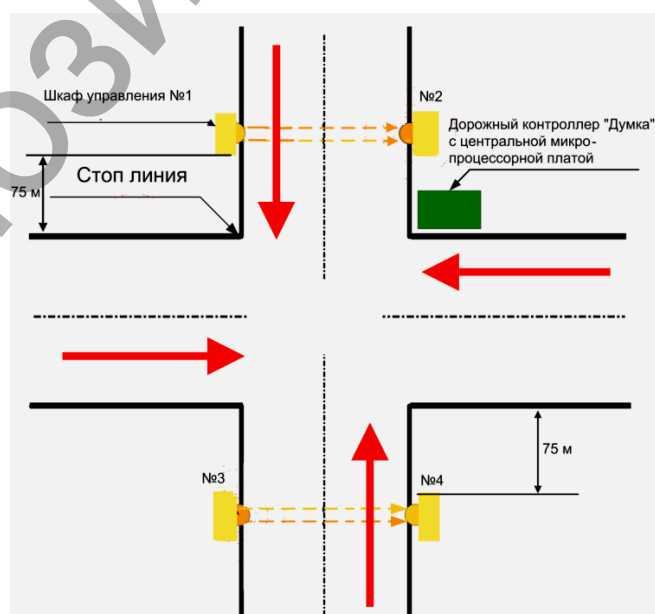


Рисунок 1 – Схема использования АСУА «Брест-1» на перекрестке

Опытный образец системы АСУА «Брест-1» изготовлен, установлен и испытан в рабочем режиме в г. Бресте, на перекрестке проспект Республики – ул. Крушинская (по согласованию с соответствующими республиканскими, областными и городскими службами).

По результатам испытания выяснилось, что пропускная способность вышеуказанного перекрестка в среднем повысилась в 1,6 раза.

При этом:

- сокращается время нахождения АТС перед светофором;
- сокращается количество дорожных «пробок»;
- снижается потребление топлива;
- сохраняется моторесурс АТС;
- сокращается количество вредных выбросов в атмосферу;
- уменьшаются уровни шума и вибрации.

Конструкция опытного образца АСУ «Брест-1» успешно выполняет свою основную функцию – управляет переключением светофоров в гибком наиболее подходящем режиме, с учетом конкретной интенсивности движения АТС.

При доработке конструкции опытного образца АСУА «Брест-1» возможно выполнение дополнительных функций:

- обеспечение беспрепятственного проезда спецавтотранспорта;
- обеспечение предпочтительного проезда общественного транспорта;
- организация движения АТС по адаптивной зеленой волне.

Положительное заключение на возможность использования адаптивной системы АСУА «Брест-1» в г. Бресте и Брестской области получено от Научно-исследовательского центра дорожного движения при Белорусском национальном техническом университете г. Минска.

Хомиченко Д.В.

АНДРОИД МОЖЕТ УПРАВЛЯТЬ РОБОТОМ!

Введение

До начала 2009 года словом «Android» называли человекоподобного робота. Сейчас для большинства людей слово **Android** – это портативная (сетевая) операционная система, основанная на ядре Linux, для коммуникаторов, планшетных компьютеров, электронных книжек, цифровых проигрывателей, нетбуков, смартфонов и даже наручных часов. Изначально разрабатывалась компанией Android Inc., которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который сейчас и занимается поддержкой и дальнейшим развитием платформы.

Смартфоны на базе ОС Android лидируют на рынке, занимая на нём 64,1% на второй квартал 2012 года. Отмечается дальнейший рост популярности – доля Android – устройств в поставках коммуникаторов приближается к 70%. По итогам второго квартала по всему миру было реализовано около 104,8 млн. штук таких смартфонов, или 68,1% от всего объёма. Актуальность выпущенных и планируемых версий ОС отображена на рисунке 1.

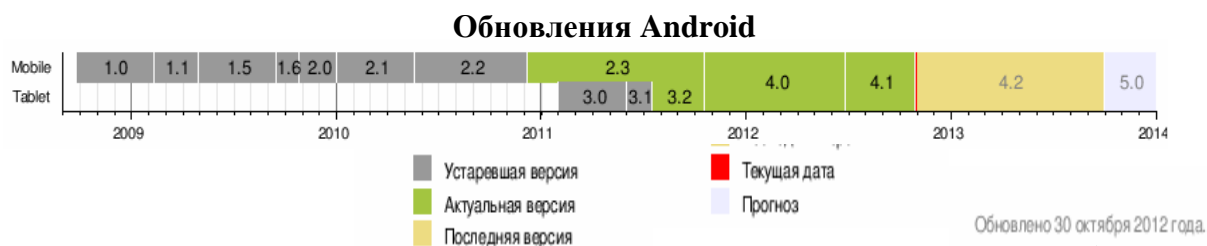


Рисунок 1 – Временная линия обновлений ОС Android

Возможности платформы Android в робототехнике

Использование Android – платформы в робототехнике оправдано по следующим причинам:

- Диапазон устройств на платформе необычайно широк (от мобильных телефонов до планшетов, в будущем и полноценная ОС).
- Платформа получила повсеместное распространение (Android-устройства действительно доступны каждому в самых разных ценовых категориях).
- Платформа легка в покупке, обновлении, поддержке и разработке.
- Платформа является бесплатной и хорошо документированной.
- Android является open-source платформой.
- Устройства на платформе Android обладает широким спектром интегрированных возможностей: мощный процессор, достаточный как для вычислений, так и для отрисовки графического интерфейса; большой и расширяемый объем оперативной памяти; акселерометр, камера высокого разрешения, GPS навигатор, микрофон, динамики, Wi-Fi и Bluetooth, встроенная поддержка сети, богатые библиотеки, удобство разработки и многие другие встроенные и подключаемые возможности.

Недостатком использования платформы является то, что она постоянно развивается и еще несовершенна. Тем не менее, преимущества использования платформы перевешивают этот недостаток.

Можно выделить две основные стратегии использования платформы Android в робототехнике: является или нет Android-устройство частью робота. При первом случае является необходимым компонентом структуры робота. Во втором подходе Android-устройство является расширением робота, формируя к нему удаленный мобильный интерфейс.

Android устройство как часть робота

Использование Android-устройства в качестве интегральной части робота может принимать следующие виды:

- *Использование Android-устройства в качестве «мозгов» автономного робота.* В этом случае Android физически установлен на робота и предоставляет свои вычислительные мощности для управления роботом и обработки сенсорных данных. Открытым остается способ связи между роботом и Android-устройством. Это может быть беспроводная связь (wi-fi/bluetooth) либо жесткое micro-USB соединение, порты для которого практически всегда имеются Android-устройствах. Другим решением данной проблемы является покупка специальной переходной платы. Примером такой платы является IOIO – I/O for Android – микроконтроллер, соединяемый по micro-USB с Android с богатой I/O периферией (48 цифровых пинов, 16 аналоговых, переключатели и т.д.)

- *Использование Android-устройства в качестве «периферии» робота.* Для восприятия мира робот может использовать богатую сенсорную периферию Android-устройства в дополнение к собственным датчикам:

- акселерометр, который можно использовать в качестве датчика движения робота (используется в инерционной навигации);

- камеру высокого разрешения, которую можно использовать в качестве наблюдения за окружающей средой робота (техническое зрение);

- GPS навигацию – для расчета местоположения робота (абсолютное позиционирование);

- компас и ориентация по сторонам света;

- микрофон для получения голосовых команд и динамики для звуковых сигналов (распознавание речи);

- Wi-Fi и Bluetooth телеметрии (телеметрия, удаленное управление, вычислительные мощности в облаке);

- сотовую связь – для беспроводного доступа к интернет сервисам и другие.

- *Android как операционная система для робота.* Это достаточно новое направление, когда робот как изделие основан на платформе Android. В отличие от предыдущих пунктов, где в качестве Android-устройства рассматривается мобильное устройство, установленное на робота, в этом примере железо робота является интегральным целым, на которое установлена ОС Android. Коротким примером данного подхода могут служить робот Android Plen. В 2011 г. компания Google объявила о совместной разработке с Willow Garage проекта ROSjava – open-source java-реализации Robotics Operation System (ROS). ROSjava сможет устанавливаться на Android-устройства и будет служить каркасом и стандартом для построения мощных робототехнических сервисов. По всей видимости, это является показателем того, что Android-платформа становится достаточно зрелой для промышленной робототехники.

Android-устройство как удаленный терминал для робота

Использование Android-устройства в качестве удаленного терминала для управления роботом может проявляться в следующих формах:

- *Использование Android-устройства в качестве удаленного терминала управления роботом.* Самый популярный способ использования Android-устройств в робототехнике ввиду его простоты и удобства для данной цели. Большинство компаний, которые промышленно разрабатывают роботов (Lego Mindstorm или iRobot Create и др.) для исследователей и хобби, предоставляют Robot/Android API для разработчиков. В данном подходе Android предоставляет удаленный интерфейс для управления роботом. Более того, специфические возможности Android (например, акселерометр) также могут быть использованы для управления робота.

- *Использование сервисов Android для эффективного управления мобильным роботом.* Android-устройство, помимо богатой периферии, обладает впечатляющим программным обеспечением и установленными сервисами. Робот может обращаться к Android для выполнения некоторых удаленных операций.

○ Android может предоставлять выход в интернет для робота. Например, опрашивать Android, есть ли на почтовом ящике робота команды для него. Так, некоторые разработчики делают веб-консоль для своего робота.

○ Другим примером является функция распознавания речи, реализованная в Android 4.0. Этот сервис без труда может быть использован для голосового управления роботом.

- *Использование Android для получения телеметрии с робота.*

○ Робот может передавать показания своих сенсоров на Android и формировать картину происходящих процессов для разработчика.

○ Робот может выполнять самодиагностику и отображать этот процесс на Android-устройстве разработчика.

○ Робот может вести лог собственной работы и хранить его в базе данных Android.

○ Робот может передавать на Android записанный звук или видео – возможности памяти на Android, как правило, гораздо больше.

Удаленное использование Android платформы позволяет использовать богатые графические возможности этой платформы для интерфейса, а использование сервисов Android позволяет реализовать концепцию cloud-robotics, когда сложные вычисления делегируются на удаленный, «облачный» сервис.

В этом разделе стоит отметить проект Cellbots, в котором разрабатывается ПО для управления роботом с мобильного телефона. Этот проект позволяет использовать большинство вышеперечисленных способов взаимодействия робот / Android.

Планы на будущее

Основной целью проекта является разработка программы, с которой мы могли связаться с роботом через Bluetooth и заставить его двигаться в соответствии с движениями мобильного телефона, используя встроенный акселерометр. Для начала робот должен уметь анализировать окружающую его среду с помощью ультразвукового датчика для обнаружения препятствий. Робот будет выполнять сканирование по принципу радара, чтобы «увидеть» препятствия.

Как только он обнаружил препятствия – он должен предоставить их для пользователя. Так же рассматривается вариант установки на робота камеры и передачи изображения в Android приложение.

Заключение

Использование мобильных приложений в робототехнике – новый шаг к усовершенствованию процесса разработки, тестированию и, конечно, практическому применению.

СОДЕРЖАНИЕ

Кабыш А.С.

**МОДЕЛЬ КООРДИНАЦИИ ПОВЕДЕНИЯ АГЕНТОВ
НА ОСНОВЕ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ 3**

Касьяник В.В., Дунец А.П.

**ОЦЕНКА ПОГРЕШНОСТИ ОДОМЕТРОВ МОБИЛЬНОГО РОБОТА
С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ..... 13**

Прокопович Г.А., Сычёв В.А.

**ПРЕДОБРАБОТКА СИГНАЛОВ ДЛЯ ИНВАРИАНТНОГО
РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ 22**

*KĘSTAS RIMKUS, ARŪNAS LIPNICKAS
ANTON KABYSH*

**DIGITAL CAMERA AND 3D LASER RANGE FINDER SYSTEM
CALIBRATION FOR ROBOT NAVIGATION..... 32**

Дёмин В.В., Дунец И.П.

**БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ АВТОНОМНАЯ МОБИЛЬНАЯ ПЛАТФОРМА
РОБОТА-ГИДА 36**

Кабыш А.С.

ПРОЕКТИРОВАНИЕ SWARM-СИСТЕМ..... 45

Андреюк А.А

**ПРОЕКТИРОВАНИЕ СИСТЕМЫ ПОСТРОЕНИЯ
ТОПОЛОГИЧЕСКОГО ГРАФА ПО ПЛАНУ ПОМЕЩЕНИЯ 54**

Богуш А.С.

КОЛЛЕКТИВНОЕ ПОВЕДЕНИЕ РОБОТОВ 57

Мешко Е.Э.

ПРОСТОЙ ГУСЕНИЧНЫЙ РОБОТ СВОИМИ РУКАМИ 60

Нефедьев А.Ю.

**АДАПТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ АВТОТРАНСПОРТНЫМИ
СРЕДСТВАМИ НА ПЕРЕКРЁСТКАХ ГОРОДСКИХ УЛИЦ,
ОБОРУДОВАННЫХ СВЕТОФОРАМИ 62**

Научное издание

РОБОТОТЕХНИКА И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ. ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ

Материалы научно-практической конференции
молодых ученых и студентов
25–26 октября 2012 года

Ответственный за выпуск: *Дунец А.П.*

Редактор: *Боровикова Е.А.*

Компьютерная вёрстка: *Кармаш Е.Л.*

Корректор: *Никитчик Е.В.*

ISBN 978-985-493-244-6



9 789854 932446

Лицензия № 02330/0549435 от 8.04.2009 г.

Подписано к печати 27.12.2012 г.

Формат 60x84 ¹/₁₆. Бумага «Снегурочка».

Гарнитура Times New Roman. Усл.-печ. л. 4,0.

Уч.-изд. л. 4,25. Заказ N 1396. Тираж 30 экз.

Отпечатано на ризографе учреждения образования

«Брестский государственный технический университет».

224017, г. Брест, ул. Московская, 267.

Репозиторий БРГТУ

Робототехника и искусственный интеллект Проблемы и перспективы

**Материалы
научно-практической конференции**



Брест 2012