

Введение

актуальность темы

Планирование движения - фундаментальная проблема робототехники и искусственного интеллекта. В последние 20 лет в научном сообществе возрос интерес к ней, особенно в задачах связанных с мультиагентными системами.

цель работы

Цель данной работы - исследование одного из способов решения данной проблемы - программирование роботов, основанное на поведении, или на реакциях на окружающую обстановку. Этот подход часто используется в тех ситуациях, где сложное поведение нужно получить при сравнительно несложной реализации отдельных единиц системы. Системы основанные на этом решении получают надежными и легко модернизируемыми.

основные задачи

Исследовать реактивный подход к решению задачи планирования движения.

Спроектировать модель системы которая сможет решить конкретную задачу планирования движения основываясь на данном подходе.

Разработать компьютерную симуляцию подобной системы, используя JavaFX в качестве инструмента для визуализации.

ВКР структура работы

области знаний

Вычислительная геометрия.

Искусственный интеллект.

Мультиагентные системы.

оценка современного состояния

все очень хорошо

новизна

практическая значимость

Глава 1. исследовательский раздел, или обзор

особенности проектируемого ПО в сравнении с аналогичными, как отечественными, так и зарубежными

Планирование движения - фундаментальная проблема робототехники. Она может быть описана как получение непрерывного движения агента из одной точки пространства в другую, с избеганием столкновений как со статичными так и с движущимися препятствиями.

характеристики аналогов и на основе их анализа обоснование необходимости разработки собственного решения

анализ исходных данных

выбор метрик для сравнения решений

To implement such a system we are going to use architecture described by Brooks [1]. This will allow us to solve the foraging task efficiently and at low cost.

A benchmark for efficiency should be the level of intelligence that can be observed in ant and termite swarms. These creatures have simple behaviors (wander around, sense food,

sense pheromone, sense pheromone, return to nest), and on their own they cannot perform the task of finding a short path to the food source efficiently. Only as they gather in big numbers intelligent path-finding emerges as a side-effect of their massive interaction.

Our implementation will avoid some of the fundamental problems that prevent multi-agent systems from being widely used for performing real-life tasks. These problems are the following:

- **Design complexity.** Usually a system of multiple robots is designed up-down way. All of its functions are implemented in different system components, all-system rules are set. The resulted behavior is often very unpredictable and complex, because it is hard to predict how

these components will interact. We will design our system bottom-up way. It means creating identical simple robots, without specifying the behavior of the whole system. The system functionality will emerge as a side-effect of massive parallel interactions.

- **Robustness.** A significant amount of complex robot implementations rely on every single sensor, camera, actuator etc. They are usually modeled to work in perfect conditions of their environment, even though it is hard to imagine a situation where everything goes as planned. Our system will be functional even if some parts of it fail.

- **Flexibility.** Systems that are designed bottom-up way are usually hard to upgrade and they do not adapt to changing environment. Our system is going to be more flexible.

выбор целевых функций

выбор оптимального варианта решения поставленной задачи

Существует два основных подхода к решению данной проблемы - логическо-пространственный и поведенческий.

В основе логического подхода лежит знание агента об окружающей обстановке. Каждый агент, использующий данный подход должен быть оборудован неким механизмом логического вывода. Он так же должен быть способен хранить и обновлять

модель окружающего его мира в терминах логики предикатов. Обновление модели происходит в результате получения информации от сенсоров и последующего рассуждения. В данном контексте под термином “рассуждение” понимается способность робота принимать решения, основываясь на знаниях об окружающем мире, и способность строить планы на будущее. Эти планы впоследствии становятся конкретными действиями, и могут изменяться в зависимости от ситуации. На первый взгляд такой подход кажется естественным, потому, что он основывается на том, как работает человеческое сознание. Но у него есть один большой минус - связанная с ним вычислительная сложность, которая возникает из-за некоторых фундаментальных проблем математики решение которых еще не было найдено. Несмотря на тот факт, что подобные системы разрабатываются и технологии в наше время быстро развиваются, до сих пор представляется фактически невозможным полностью воссоздать механизм работы человеческого мозга.

Поведенческий подход (его так-же можно называть реактивным) обычно ставится в противопоставление логическо-пространственному. Реактивное принятие решений основывается не на накопленном знании об окружающем мире и не на построенных заранее планах. Оно основывается только на информации о текущей обстановке, здесь и сейчас. Множество примеров подобного поведения можно встретить в мире простейших животных или насекомых. Основная задача, которую решают насекомые - передвижение в поисках пищи.

Без сомнения, логический подход необходим при решении сложных задач, в особенности тех где необходимо показать точный путь и причины выбора того или иного шага на этом пути, например доказательство математических теорем. В своей работе мы покажем, что для задачи планирования движения в контексте мультиагентной системы больше подходит реактивный подход.

выбор методов проектирования, обоснование выбора

Представим ситуацию, когда космический корабль (один из примеров автономного агента) отправляется на другую планету с исследовательской миссией. Основная задача миссии - собрать как можно больше разлных образцов породы (объектов интереса) с большой территории. Это хороший пример реальной задачи которую уже способны выполнять роботы. К решению задачи можно подойти с различных позиций.

Одним из таких решений может быть построение сложного робота, способного при помощи логики преобразовывать имеющуюся информацию в конкретные действия. К такой информации может относиться например расстояние до ближайших объектов, текущая температура окружающей среды, наличие или отсутствие освещения, и т.д. Так-же обычно такие роботы должны обладать возможностью посылать и принимать радиосигналы высокой дальности для отправки фотографий или принятия указаний к действиям. Таким образом устроены роботы которые сейчас работают на планете Марс, всего их три: Spirit, Opportunity и Curiosity.

С нашей точки зрения такой подход к решению задачи нельзя назвать надежным и быстрым как минимум по двум причинам. Во-первых робот описанный выше должен быть очень хорошо собран, и оттестирован, так как если хотя бы одна его часть (сенсор, шасси, камера и т.д.) перестанет функционировать, то тогда всю миссию можно считать проваленной. То есть первая проблема это трудность сборки единого целого из множества сложных составных частей. Вторая проблема заключается в том, что даже если подобный робот будет способен бесперебойно осуществлять свою деятельность в экстремальных условиях, то для того чтобы охватить большую территорию, ему понадобится много времени.

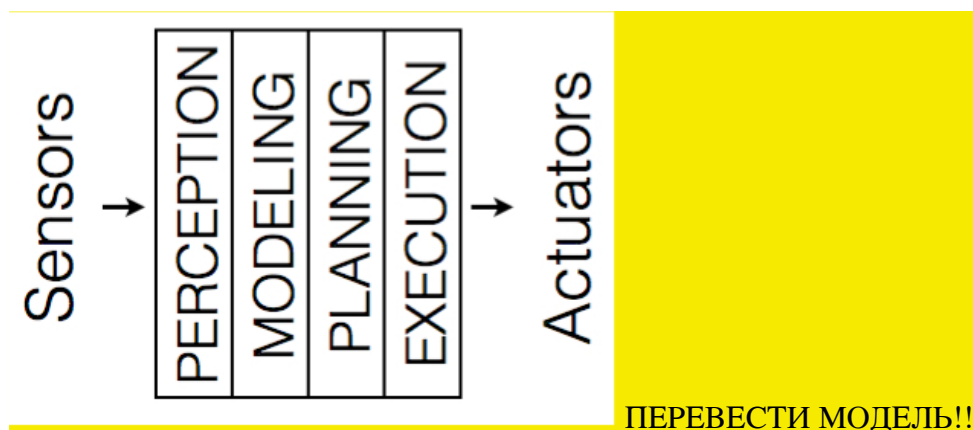
Другой подход, который можно использовать для решения задачи исследования некоторой поверхности - создание децентрализованной системы роботов небольшого размера. Эти роботы должны быть полностью автономными, и им не понадобятся средства для связи на большом расстоянии. Такая система имеет ряд преимуществ:

- **Масса при транспортировке.** Небольших размеров роботов легче транспортировать, их можно так-же для удобства и надежности транспортировать отдельными группами.
- **Взаимозаменяемость.** При отказе нескольких роботов, подобная система сможет гарантированно выполнить задачу, пусть и за большее время.
- **Децентрализованность.** Роботы в подобной системе могут быть абсолютно автономными и у них нет необходимости подчиняться командам центрального сервера.

- **Стоимость проектирования и производства.** Стоимость подобной системы может быть значительно ниже, чем стоимость одного очень сложного робота.

- перепрограммирование made easy
- масштабирование

Теперь опишем архитектуру отдельно взятого робота. Эта архитектура основывается на работе Brooks [1]. Она отличается от традиционной модели построения системы управления роботом. Так выглядит традиционная модель:

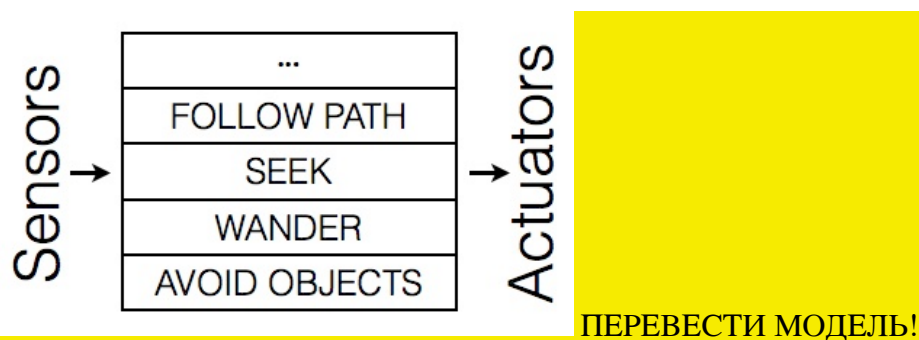


Эта модель состоит из шагов по которым идет любой робот, принимая решения к действию. Сначала происходит срабатывание некоторых сенсоров, затем информация, поступающая с этих сенсоров обрабатывается. После этого робот переводит обработанную информацию в понятную ему модель и проводит на основании этой модели планирование дальнейших действий. Затем эти действия из модели переводятся в конкретные действия

движущихся частей (например напряжение и угол поворота моторов).

Шаги, представленные в традиционной архитектуре построения, неотделимы друг от друга, и должны выполняться строго один за другим. Основная проблема данной архитектуры в том, что если на одном из шагов произойдет ошибка, то она точно отобразится на всех последующих.

Наша модель построения архитектуры выглядит следующим образом:



Она основывается на конкретных поведеньях, которые вместе организованы как слои. Их можно реализовывать отдельно друг от друга, и они могут выполняться параллельно и независимо. Преимущества подобной архитектуры:

- **Несколько целей.** Слои (или поведенья) могут работать одновременно над несколькими задачами, к примеру можно поставить одновременно две задачи: избегать препятствия и достигнуть определенной точки.
- **Надежность.** После того, как один слой разработан и протестирован, к нему можно уже не возвращаться. Можно быть уверенным что при насаивании новых поведений, уже существующие не изменятся.

выбор программных и аппаратных средств решения поставленной задачи

В конце исследовательского раздела приводится постановка задачи на разработку (на исследование).

Глава 2. специальный раздел, или предлагаемые решения

Специальный раздел - основа ВКР, ему должны быть подчинены остальные разделы.

В этом разделе описывается разработка программного средства – от формальных моделей (проектирования структурных, функциональных схем, UML-моделей, структур баз данных и т.п.) до разработки алгоритмов, программ и пользовательских интерфейсов. Подготовка ВКР.

Конкретное содержание специального раздела определяется темой ВКР.

----- CONTENT

Для разработки системы необходимо было выбрать конкретную задачу. Так как эта система основана на поведении простейших живых организмов, мы решили выбрать из тех задач которые преподносит природа этим организмам. Одна из таких задач - поиск пищи. Из нее мы извлекли следующие условия:

- некоторое ограниченное 2D пространство
- некоторая исходная точка
- ...
- препятствия расставленные некоторым образом
- расположенные в случайных местах предметы которые нужно с

В реальности эта задача может быть сформулирована в следующих примерах:

- сбор токсичных отходов с места аварии
- исследование других планет или любого другого пространства, недоступного человеку.

----- ПРОБЛЕМА ПОИСКА