Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Томский политехнический Университет»



Инженерная школа информационных технологий и робототехники

Отделение автоматизации и робототехники

15.03.06 «Мехатроника и робототехника»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛАБОРАТОРНАЯ РАБОТА №1,2

Реализация сценария “Охраняемая цель” на платформе Robotino

**Вариант – 3**

по дисциплине:

**Интеллектуальное управление робототехническими комплексами и системами**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Исполнитель:** | |  | | | | |
| студент группы | | 8Е12 |  | Федоренко Д. В. |  | 15.04.2025 |
|  | |  |  |  |  |  |
| **Руководитель:** | |  | | | | |
| преподаватель | |  |  | Андраханов А. А. |  | 15.04.2025 |
|  | |  |  |  |  |  |
|  |

Томск – 2025

# Содержание

[Содержание 2](#_Toc195286024)

[Задание 3](#_Toc195286025)

[Параметры используемого робота 4](#_Toc195286026)

[Предполагаемая логика решения задачи 5](#_Toc195286027)

[Необходимые данные 6](#_Toc195286028)

[Фазификация параметров 7](#_Toc195286029)

[Расстояние до цели 7](#_Toc195286030)

[Расстояние до объектов 8](#_Toc195286031)

[Скорость объекта 9](#_Toc195286032)

[Скорость робота 10](#_Toc195286033)

[Программирование 12](#_Toc195286034)

[Первичная отладка 12](#_Toc195286035)

[Новый подход к задачи 13](#_Toc195286036)

[Новая фазификация и параметры 13](#_Toc195286037)

[Обновлённая система правил 17](#_Toc195286038)

[Программирование 20](#_Toc195286039)

[Вывод по работе 21](#_Toc195286040)

# Задание

Робот находится в стартовой позиции, равноудаленной от левого и правого краев рабочей сцены и на расстоянии 10 см от его борта до нижнего края. Цель (обозначена «\*») является статической, координаты её известны.

Цель окружают 3 динамических объекта диаметром 10 см, вращающихся вокруг неё по окружности некоторого радиуса в пределах от 15 до 50 см (расстояние каждого объекта от цели может быть индивидуальным). Угловое расстояние между объектами может быть любым, а скорость их углового перемещения может быть различной, вплоть до 80 град/с. Вращение объектов всегда происходит синхронно, а центр окружности их вращения всегда совпадает с центром цели. Логика перемещения объектов роботу не известна.

Задача заключается в том, что робот, двигаясь в голономном режиме (с поворотами или без поворотов в сторону результирующего вектора скорости – на усмотрение разработчика), должен достичь координат цели в пределах окрестности радиусом 2 см.

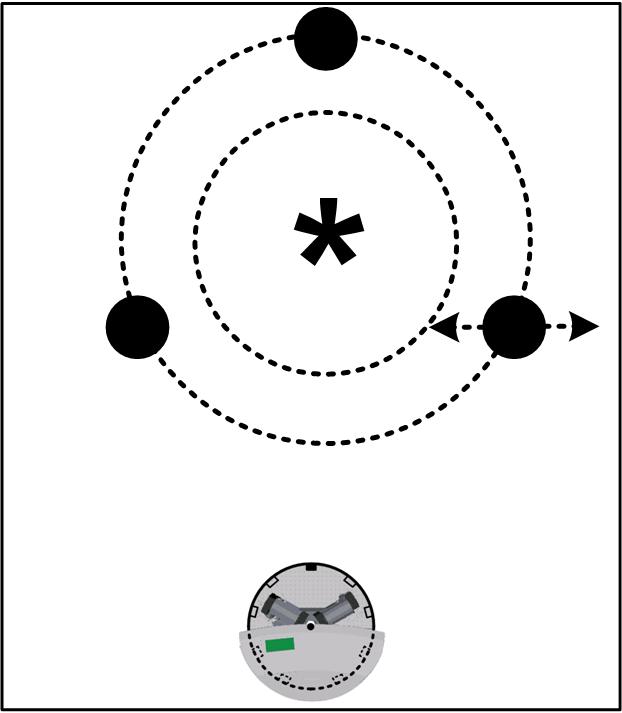


Рисунок 1- Иллюстрация к заданию

# Параметры используемого робота

Одним из аспектов робота является расположение 9 дальномеров по окружности корпуса под углом 40 градусов друг к другу. При этом дальномеры работают на дистанцию около 40-50 см с некоторым рассеиваньем (диаграммой направленности).

Из физических параметров робота наиболее интересным является диаметр робота, так как робот имеет конструкцию корпуса с формой круг, имеющий диаметр около 50 см.

Следящий параметром является скорость. Пиковая скорость робота составляет около 10 км/ч, что равносильно 2.78 м/с или 278 см/с. Однако стоит учесть, что более реально рассчитывать для безопасной работы на скорость в 20 см/с c предельной скоростью в 40 см/с, которая применяется для совершения рывков для ухода от препятствий или финального рывка.

# Предполагаемая логика решения задачи

Согласно задаче, роботу необходимо прорваться через ряд подвижных предпястий, не сталкиваясь с ними, попасть в центр их вращения (цель). Первым шагом будет приближение к зоне прорыва на дистанцию рывка, учитывая окружности движения цели.

Для этого необходимо определить радиус вращения объекта, для этого необходимо приблизиться к зоне на дистанцию детектирования сенсорами (до 40 см), в частности можно приблизиться на дистанцию 20 см к максимально возможному радиусу (его можно знать косвенно относительно центра (цели)). После необходимо детектировать объект на расстоянии.

# Необходимые данные

Для составления свода правил, предварительно необходимо определить необходимые параметры

Таблица 1 – Таблица параметров входных

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение параметра | Наименование параметра | Размерность | Описание |
| L | Дистанция до цели | метры | Оценка расстояние от центра робота до центра области назначения |
| l1, l2, l3, l4, l5,l6 | Дистанция до препятствия | метры | Оценка расстояния до препятствия 1-6 |
| U1, U2, U3, U4, U5, U6 | Скорость цели | м/с | Оценка скорости видимых объектов |

Таблица 2 – Таблица выходных параметров

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение параметра | Наименование параметра | Размерность | Описание |
| Ux | Скорость робота по X | м/с | Скрасть движения по X |
| Uy | Скорость робота по Y | м/с | Скрасть движения по Y |

Таблица 1 и 2 предполагает, что данных наборов параметров будет достаточно для функционирования системы и составления набора правил.

**Действия**

Часть действий робота в рамках задачи решается чёткой логикой, в частности, сам факт движения к точки.

Если в зоне робота оказывается какое-то препятствие, то в дело вступает нечёткая логика. Предполагается, что робот будет ожидать возможности для сближения с целью и после предпринимать соответствующие действия.

# Фазификация параметров

## Расстояние до цели

Данный параметр показывает дистанцию до цели по кратчайшей траектории, то есть прямой. Данные берутся ввиду наличия у робота своей локальной системы координат, внутри которой можно установить точку нахождения объекта.

Вариант распределения представлен на рисунке.

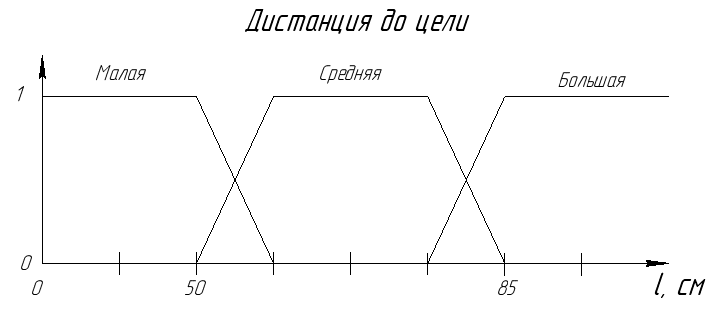


Рисунок 2 - Распределение дистанций

## Расстояние до объектов

Всего есть три охранных объектов, на самом деле, большую часть времени, вероятно, робот будет видеть один или два объекта единовременно. Распределение дистанций до них будет одинаково, так как объективно их идентифицировать будет крайне сложно, если возможно.

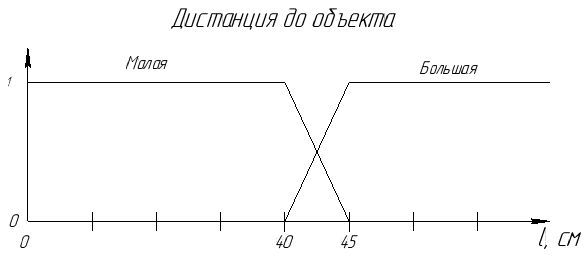


Рисунок 3 - Распределение дистанций до объекта

Данная система распределений имеет всего два класса, так как система фактически должна суметь определить: можно ли проскочить перед объектов или нет.

## Скорость объекта

Данный параметр нужен для определения направления движения объекта. Подобная информация важна для определения степени угрозы объекта для робота. Например, если объект удаляется, значит можно планомерно двигаться к цели, а не делать рывок в обратную сторону ввиду угрозы столкновения.

Учитывать нужно два возможных направления скорости. При этом стоит отметить, что, фактически, нас интересует лишь скорость сближение, а не отдаления, так как объекты, отдаляющиеся от нас, не представляют никакого интереса и его не стоит учитывать в логике изменения маршрутов.

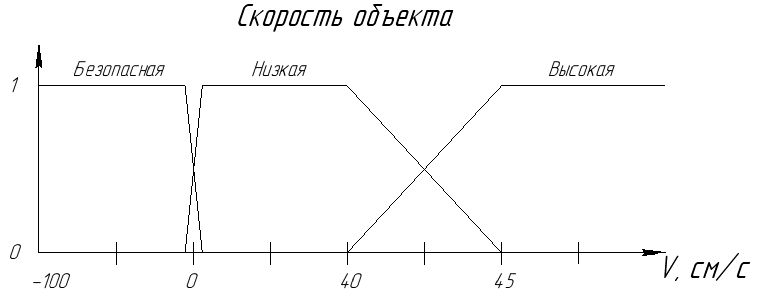


Рисунок 4 - Фазификация скорости объекта

## Скорость робота

Предположительно скорость робота будет разбита как медленная и быстрая, так как система не требует более тонкого регулирования скорости. Скажем системе нужна низкая скорость для плавных и точных движений (приближение к зоне с охранными объектами, смещение в рамках безопасной зоны без охранников), так и высокая, назначение которой будет стремительное движение к объекту в зоне, где охранные модули могут соударяться с роботом.

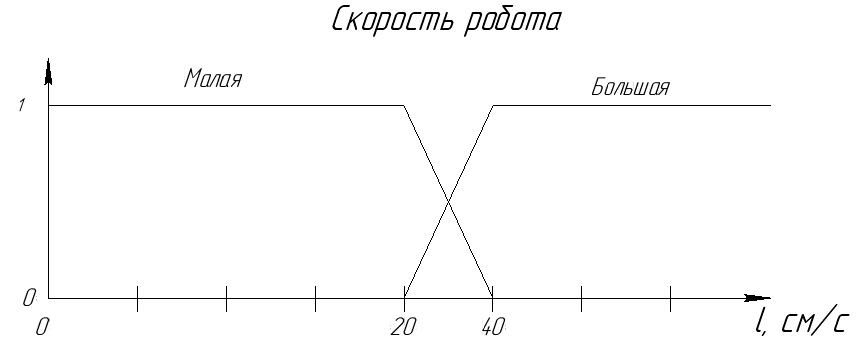


Рисунок 5- Фазификация скорости робота

Как итог, данное распределение параметров позволяет роботу принимать взвешенные решения, сопутствующие выполнению основной задачи робота.

**База правил**

Данный раздел описывает набор правил, на основе которых робот будет принимать какие-то конечные решения в тех или иных ситуациях или скорее их комбинации.

1. Если дистанция до цели большая, то приблизиться медленно на дистанцию средняя.
2. Если скорость объекта справа высокая и дистанция далеко, дистанция до цели средняя – смещение по диагонали влево и назад быстро до большой дистанции к цели.
3. Если дистанция до объекта слева близко и скорость объекта высокая, дистанция до цели средняя – смещение по диагонали влево и вперёд быстро до цели.
4. Если скорость объекта справа низкая и дистанция далеко, дистанция до цели средняя – рывок вперёд.
5. Если скорость объекта слева низкая и дистанция далеко, дистанция до цели средняя – рывок вперёд.
6. Если скорость объекта справа низкая и дистанция близко, дистанция до цели средняя – рывок вперёд.
7. Если скорость объекта слева низкая и дистанция близко, дистанция до цели средняя – рывок вперёд.
8. Если дистанция до цели большая, то сместиться по окружности (текущего радиуса) примерно на 45 градусов окружности и приблизиться на среднюю дистанцию до цели.

Описанная логика должна позволить роботу выполнить поставленную задачу и избежать случайных столкновений. Следующим этапом будет программирование непосредственно робота с подгрузкой базы правил и тестирование, отладка робота.

# Программирование

В рамках данного раздела рассмотрим принцип работы программы и конкретных алгоритмов, в частности отметим, что рабочая программа поделена на два блока: main и fuzzy.

Блок main – является непосредственно главным блоком программы, внутри которого осуществляется подключение к Robotino и основное взаимодействие с ней.

Блок fuzzy выступает в роли управляющего блока, который принимает решение о том, как действовать в текущей ситуации на основе заложенной базы правил с учётом текущих данных с датчиков.

# Первичная отладка

На основе заложенной базы правил, робот вёл себя не корректно, при этом точную причину некорректной работы определить оказалось затруднительно даже при попытке отлаживать правила по отдельности, вероятно, что правила как-то пересекались и могли давать роботу противоречивые команды, не позволяющие ему в полной мере работать.

Так же было обнаружено, что вычисление скорости объектов представляет собой сложную задачу и, чаще всего, ненужную, так как данная информация серьёзно усложняет проект и делает отладку более затруднительной.

При этом сенсоры расстояний работают так же не корректно, ввиду того, что их эффективная дальность составляет около полуметра, но чаще всего ограничена на значении в 40-45 сантиметров, поэтом текущая фазификация данных с датчиков работает не совсем корректно.

На основе этого можно попробовать составить новую логику работы робота, частности, разрабатываемую более детально.

# Новый подход к задаче

Ещё раз рассмотрим задачу и заметим, что, формально, её можно свести к обходу препятствий с стремлением к конкретной точке, указанной в задании – центр окружности. Это задача позволяет абстрагироваться от скоростей.

## Новая фазификация и параметры

Так как с прошлыми параметрами возникли проблемы, было принято решение упростить систему входных и выходных параметров. Результатом стала таблица 3.

Таблица 3 – Таблица параметров входных, новая

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение параметра | Наименование параметра | Размерность | Описание |
| X\_point, Y\_point | Позиция цели относительно робота | метры | Оценка удалённости от центра робота до центра точки |
| l1, l2, l3, l4, l5,l6 | Дистанция до препятствия | метры | Оценка расстояния до препятствия 1-6 |

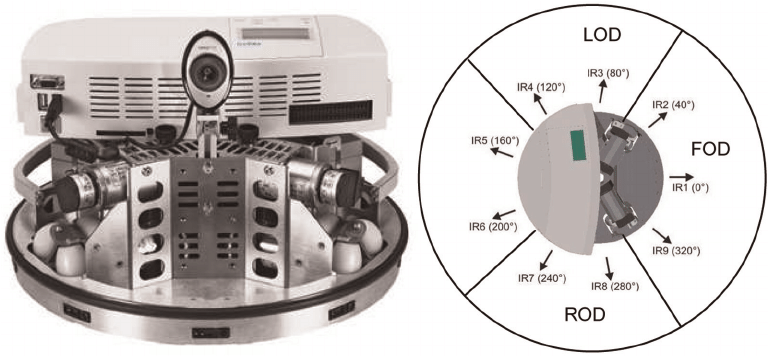


Рисунок 6- Распределение сенсоров на роботе

В проекте задействованы все сенсоры, кроме заднего, так как в рамках текущей задачи информация с него не является актуальной и целевой.

Таблица 4 – Таблица выходных параметров, новая

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение параметра | Наименование параметра | Размерность | Описание |
| Ux | Скорость робота по X | м/с | Скрасть движения по X |
| Uy | Скорость робота по Y | м/с | Скрасть движения по Y |

Рассмотрим тогда новую систему фазификации, учитывая новые данные. Во первых, был совершён отказ от фазификации параметра дистанции до цели, так как данный параметр перестал влиять на работу, фактически, данный параметр просто позволяет нам оценить степень приближения робота, он не будет принимать участие в правилах.

Дистанция до преград будет фазифицироваться следующим образом, показанным на рисунке 5.

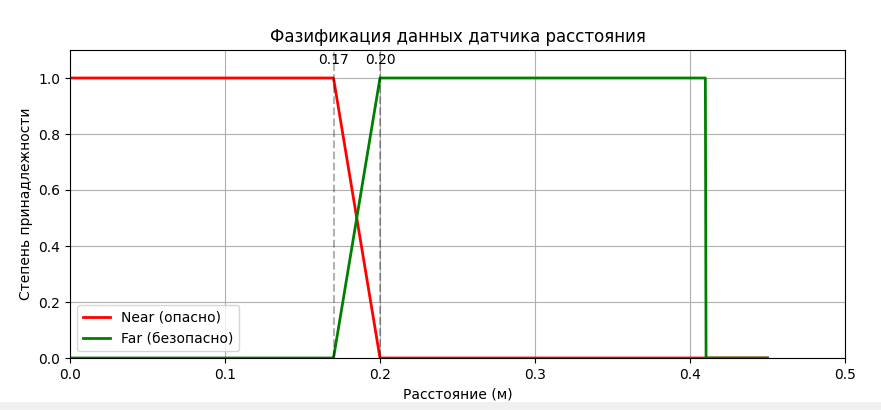


Рисунок 7 - Фазификация дистанции до преграды

Деление так же осталось на два класса, но изменились значения деления. Они представлены на изображении. Опытным путём было выяснено, что это оптимальные параметры для работы системы сенсоров.

Теперь перейдём к фазификации скоростей по оси X. Новая система фазификация представлена на рисунке 7.

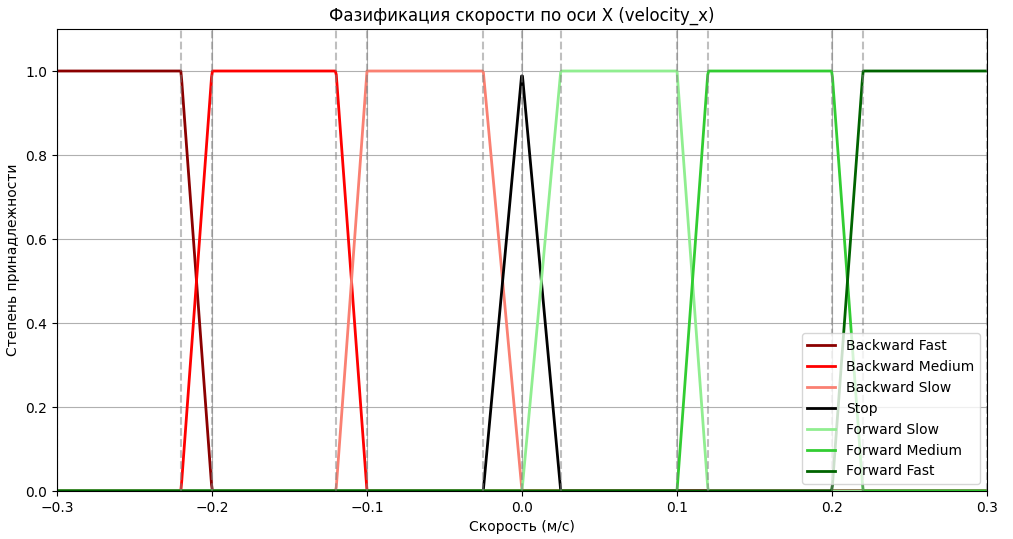


Рисунок 8 - Фазификация скорости по оси X

По аналогии была проведена фазификация скорости по оси Y. Фазификация представлена на рисунке 8.

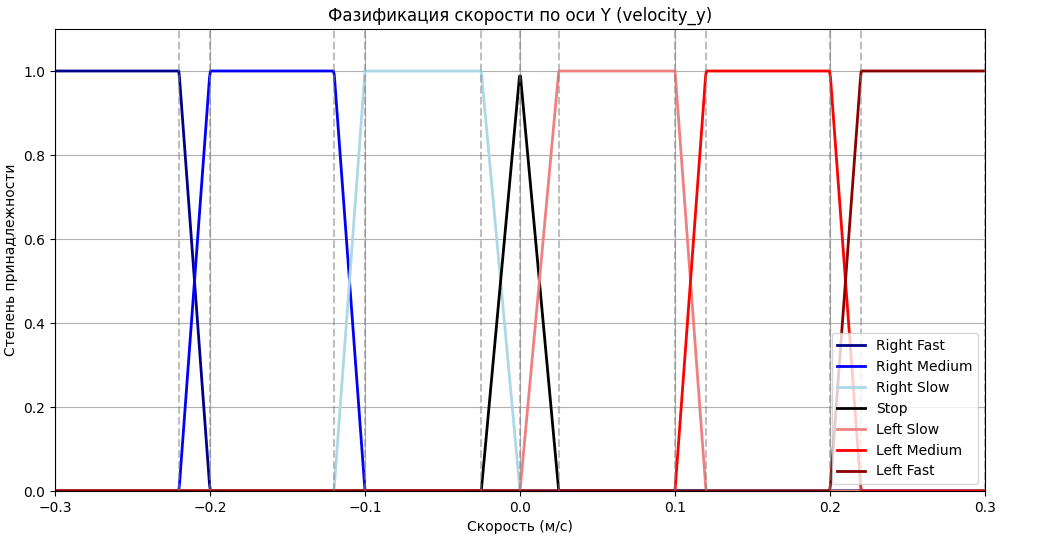


Рисунок 9 - Фазификация скорости по оси Y

Так же были введены новые данные для фазификации положения цели относительно робота, данная физификация представлена на следующем изображении (рисунок 10 и рисунок 11).

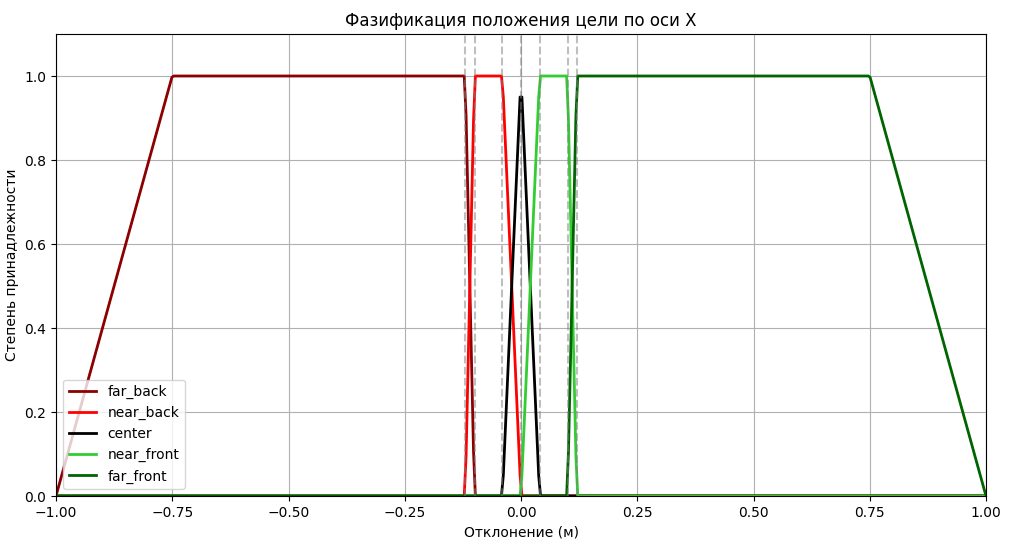


Рисунок 10 – Фазификация положения цели относительно робота (ось X)

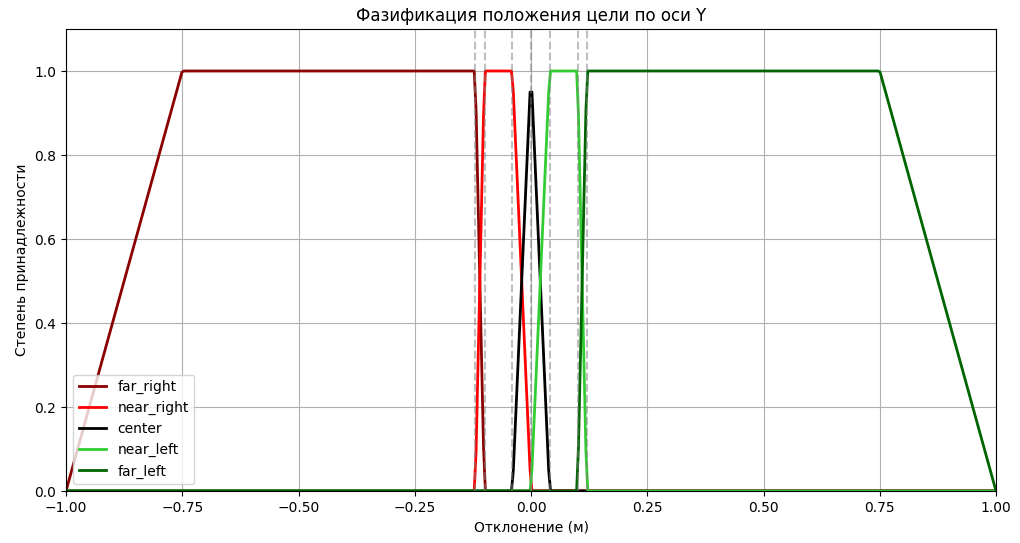


Рисунок 11 – Фазификация положения цели относительно робота (ось Y)

Теперь, рассмотрев новые параметры для фазификации, теперь будем рассматривать и переходить и изучению правил.

## Обновлённая система правил

Всего существует три блока правил, которые активируются в зависимости от обстоятельств.

Первый блок правил – движение к цели. Данный блок состоит из следующего набора правил, описанных в таблице 5.

Таблица 5 – Правила движения к точек

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Условие по X | Условие по Y | Действие по X | Действие по Y |
| 1 | Точка далеко сзади | Точка далеко справа | Быстрое движение назад | Быстрое движение направо |
| 2 | Точка далеко впереди | Точка далеко слева | Быстрое движение вперёд | Быстрое движение налево |
| 3 | Точка близко сзади | Справа близко | Медленное движение назад | Медленное движение направо |
| 4 | Точка близко спереди | Близко слева | Медленное движение вперёд | Медленное движение налево |
| 5 | Точка в центре | Точка в центре | Остановка | Остановка |

Данный набор правил имеет наименьшей приоритет и работает всегда, когда не активные другие наборы правил.

Следующий набор правил отвечает за объезд препятствий и активируется, если хотя бы один из сенсоров уловил наличие рядом объекта на опасной дистанции (состояние близко к роботу).

Таблица 6 – Правила объезда препятствий

|  |  |
| --- | --- |
| **Блок объезда левых препятствий** | |
| Условия | Результат |
| Левый передний сенсор – близко  Левый боковой сенсор – близко  Правый сенсор передний - далеко  Передний сенсор – далеко | Движение – медленно вперёд (x)  Движение – медленно вправо (y) |
| Левый передний сенсор – далеко  Левый боковой сенсор – близко  Передний сенсор – далеко | Движение – быстро вперёд (x)  Движение – быстро вправо (y) |
| **Блок объезда правых препятствий** | |
| Правый передний сенсор – близко  Правый боковой сенсор – близко  Левый сенсор передний - далеко  Передний сенсор – далеко | Движение – медленно вперёд (x)  Движение – медленно влево (y) |
| Правый передний сенсор – далеко  Правый боковой сенсор – близко  Передний сенсор – далеко | Движение – быстро вперёд (x)  Движение – быстро влево (y) |
| **Блок объезда передних препятствий** | |
| Условия | Результат |
| Передний сенсор – близко  Левый передний сенсор – далеко  Первый передний сенсор - далеко | Движение – назад медленно (x)  Движение – стоп (y) |
| Передний сенсор – близко  Левый передний сенсор – близко | Движение – медленно назад (x)  Движение – медленно вправо (y) |
| **Блок объезда задних препятствий** | |
| Левый задний сенсор – близко  Правый задний сенсор - далеко | Движение – быстро вперёд (x)  Движение – медленно вправо (y) |
| Левый задний сенсор – далеко  Правый задний сенсор - близко | Движение – быстро вперёд (x)  Движение – медленно влево (y) |

Последний, третий набор правил – необходим для решения спорных ситуаций, возникающих при работе основного набора правил. Данные правила интегрированы в правила объезда препятствий и, поэтому, работают в связки с ним и имеют одинаковый с ним приоритет.

Фактически, данные правила просто повышают гибкость системы.

Таблица 7 – Динамические правила

|  |  |
| --- | --- |
| Условия | Результат |
| Цель - спереди  Левый передний сенсор – близко | Движение – медленно вперёд (x)  Движение – медленно вправо (y) |
| Цель - слева  Передний сенсор – близко | Движение – назад вперёд (x)  Движение – медленно вправо (y) |
| Цель – далеко слева или далеко справа  Передний сенсор – далеко | Движение – назад вперёд (x)  Движение – медленно вправо (y) |

На это разбор системы правил окончен и можно перейти к обсуждению принципов работы программы.

## Программирование

Программа разбита на два модуля – main и navigation. Первый модуль, main, отвечает за базовые функции, например: подключение, чтение данных с датчиков, чтение данных с внутренней одометрии, отправление управляющих команд, вычисление отклонений от цели. Так же он содержит в себе главный цикл, из которого будут вызываться все остальные функции и процедуры.

Второй модуль, navigation, представляет собой модуль нечёткой логики, который будет принимать решения (принимать входные параметры, анализировать, с помощью нечёткой логики выдавать результирующие воздействия). Внутри модуля содержатся правила нечёткой логики, данные о фазификации и дефазификации всех выходных/выходных параметров. Более подробно про код можно просмотреть на GITHAB, по [ссылке.](https://github.com/GreyPaldin/Robotino_LB)

Или на основе QR кода, представленного ниже на рисунке 12.



Рисунок 12 – QR код с адресом на проект в GITHAB

В репрезитории GITHAB есть модуль READMY, в котором содержится описание работы кода и расписаны некоторые особенности его работы. На этом раздел закончен.

# Корректировка работы программы на основе испытаний (защиты)

В рамках защиты работы было выяснено, что робот абсолютно не приспособлен к ситуации, когда вокруг него находится 4 стенки на 4-рёх базовых направлениях (спереди, сзади, слева, справа).

В них робот зависал и не предпринимал попыток выбраться из данной ситуации, пока не будут убраны любые 2 стенки. В обновлённой версии робот, начав в аналогичной ситуации, начинает колебаться в рамках замкнутого пространства, что абсолютно нормально в рамках нечёткой логики. При это робот, в случай возникновения прохода спереди, слева, справа – способен воспользоваться брешью и выйти из сложившейся ситуации, после чего устремляется к цели.

Проблема некорректного старта в замкнутом пространстве была решена путём изменения алгоритма инициализации переменных дистанций с данными с сенсоров. Достаточно было инициализировать переменные не с типом None (как это было ранее), а с присвоенным каким-либо значением, например 0 в первой итерации цикла. За счёт большой частоты обновлений, сразу же после запуска, робот перезаписывает все эти переменные и, даже если в пришедшем ему пакете не хватает каких-либо данных, у него есть начальные значения, которые обновляется с следующим, более корректным пакетом.

Так же было добавлено два правила, которые представлены в таблице 8.

|  |  |
| --- | --- |
| Условия | Результат |
| Преграда – спереди  Преграда – слева  Свободно - справа | Движение – стоп (x)  Движение – быстро вправо (y) |
| Преграда – спереди  Преграда – справа  Свободно - слева | Движение – стоп (x)  Движение – быстро влево (y) |

Правил для движения вперёд не потребовалось, так как робот корректно справлялся с данной задачей на основе старой системы правил (после изменения алгоритма инициализации переменных.

# Вывод по работе

В рамках первой лабораторной работы были выполнены основные этапы, которые включат в себя разбор задачи, определение измеряемых параметров, установление зависимостей между чёткими значениями параметров (в численном виде) и их нечёткими значениями (различными формулировками).

Так же был составлен свод правил, в рамках которых робот функционировал для достижения цели. Особенностью свода правил является универсальность, позволяющая роботу выполнять свою задачу вне зависимости от условий, если они вписываются в ранее описанные параметры задачи.

Был написан код на языке программирования Python для взаимодействия с Rabotino и проверки работы системы правил на практике с последующей доработкой, описанной в предпоследней главе. В ходе повторных испытаний было установлено, что код работает корректно.