



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО МГТУ «СТАНКИН»)

**Институт
автоматизации
и робототехники**

**Кафедра
робототехники и мехатроники**

Методические указания к лабораторной работе №4
«Управление мобильным роботом
с помощью контроллера на основе принципов нечёткой логики»
Дисциплина: «Современные методы управления в робототехнике и
мехатронике»

Р.В. Колесниченко, Ю.В. Илюхин

Москва 2017 г.

Цель: научиться управлять мобильным роботом с помощью системы управления на основе **нечёткого** регулятора. Изучить структуру контроллера нечёткой логики на основе алгоритма Мамдани.

Задачи: создать систему управления мобильным роботом с помощью контроллера нечёткой логики. Оптимизировать систему управления с целью получения минимального времени переходных процессов. Модифицировать систему управления с целью возвращения робота в начальное положение.

Постановка задачи.

Автономный мобильный робот, следящий за ситуацией на плоскости, находится в точке нулевого положения. На плоскости также находится объект (нарушитель), движущийся по прямой под некоторым углом и с постоянной скоростью (например, катящийся шар). Как только он попадает в зону обнаружения робота, тот стремится сблизиться с объектом и поймать его с помощью имеющегося на борту манипулятора за минимальное время.

Формально, постановка задачи выглядит следующим образом. Робот находится на плоскости в начале системы координат. Его продольная ось совпадает с осью X этой системы координат. В начальный момент времени в некоторой точке на оси Y появляется объект, который движется с постоянной скоростью V под заданным углом к оси X . Робот может развивать скорость V_p больше V . Под действием нечёткого управления он должен двигаться в направлении объекта.

Задача считается решённой, если расстояние от робота до движущегося объекта стало меньше R , например, $R=0,5$ м, и, одновременно с этим, угол между направлениями движения робота и объекта не превышает 20 градусов (0.349 рад). В таком случае манипулятор сможет захватить объект, после чего робот может двигаться в нулевую точку.

Кинематическая модель мобильного робота.

В качестве примера в лабораторной работе рассматривается мобильный робот «Pioneer 3-DX» фирмы ActiveMedia, изображённый на рисунках 1 и 2. Такой мобильный робот имеет три колеса. Два из них являются ведущими колесами и приводятся в движение независимыми электроприводами. Оси этих колёс не меняют своего положения относительно несущей платформы робота. Заднее колесо представляет собой флюгерное колесо. Поворот робота осуществляется при различии значений угловых скоростей вращения передних колёс.



Рисунки 1, 2. Общий вид мобильного робота «Pioneer 3-DX».

В лабораторной работе принято допущение, что робот движется по твёрдой плоской поверхности без проскальзывания, а колёса являются недеформируемыми и находятся в точечном контакте с поверхностью.

Кинематическая схема мобильного робота, находящегося на плоской поверхности в системе координат OXY , изображена на рисунке 3. Будем называть эту систему координат абсолютной системой координат. В точках L и R расположены левое и правое передние колеса. Их оси вращения совпадают с отрезком LR , имеющим длину l . В середине отрезка

LR находится начало системы координат $O_oX_oY_o$, жёстко связанной с роботом. Ось X_o этой системы координат направлена перпендикулярно LR в направлении движения. Ось Y_o вместе с осью X_o образуют правую систему координат. Положение основания робота как твердого тела задается величинами φ , x , y , где φ – угол между осями X и X_o , а x и y – координаты точки O_o в абсолютной системе координат OXY .

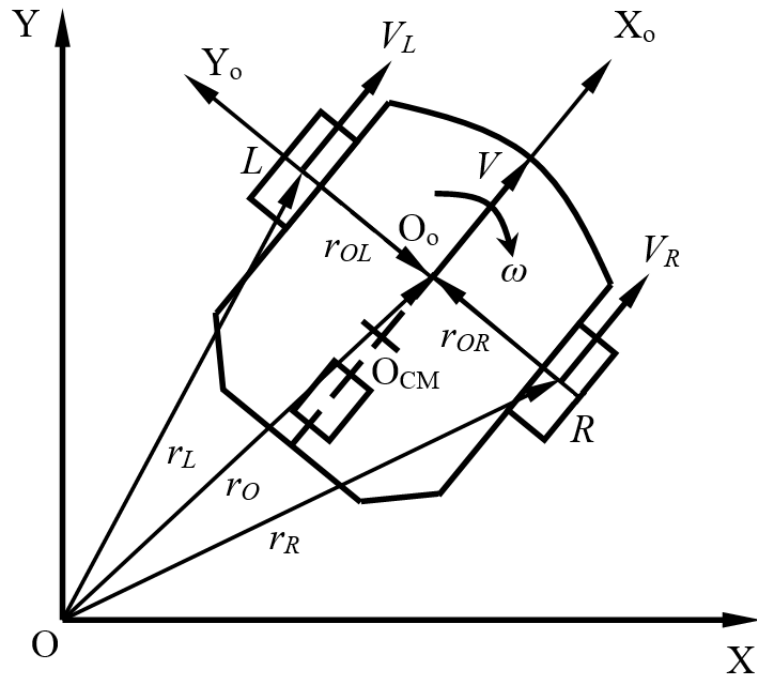


Рисунок 3. Кинематическая схема мобильного робота.

Положение точек робота O_o , L и R в абсолютной системе координат определяется радиус-векторами r_O , r_L и r_R . При этом положение точки O_o может быть получено из уравнений (1).

яя r_{OL} – радиус-векторы, показанные на рисунке 3, причём $r_{OR} = -r_{OL}$.

Связь линейной скорости движения V точки O_o связанной с мобильным роботом системы координат $O_oX_oY_o$ с линейными скоростями V_R и V_L точек R и L и угловой скоростью ω вращения робота относительно точки O_o определяется уравнениями (2).

$$\begin{cases} V = V_R + \omega \times r_{OR}, \\ V = V_L + \omega \times r_{OL}. \end{cases} \quad (2)$$

Выполнив сложение и вычитание этих уравнений, получим систему уравнений (3).

$$\begin{cases} \omega = \frac{1}{l}(V_R - V_L), \\ V = \frac{1}{2}(V_R + V_L). \end{cases} \quad (3)$$

С учётом очевидных связей переменных x , y , φ с переменными V и ω , которые задаются уравнениями

$$\begin{cases} \dot{x} = V \cdot \cos \varphi \\ \dot{y} = V \cdot \sin \varphi \\ \dot{\varphi} = \omega \end{cases}, \quad (4)$$

получим систему нелинейных дифференциальных уравнений (5), описывающих кинематическую модель трехколесного мобильного робота.

$$\begin{cases} \dot{x} = \frac{1}{2}(V_R + V_L)\cos\varphi \\ \dot{y} = \frac{1}{2}(V_R + V_L)\sin\varphi \\ \dot{\varphi} = \frac{1}{l}(V_R - V_L) \end{cases} \quad (5)$$

Скорости V_R и V_L зависят от угловых скоростей вращения выходных валов приводов и представляют собой управляющие воздействия. Линейная V и угловая ω скорости движения робота могут быть найдены из уравнений (3).

Составляющие системы управления роботом в соответствии с постановкой задачи.

1) Система управления робота (файл `robot_kinematic_model.mdl`).

Прежде всего зададим начальные условия движения робота и объекта, представленные в таблице 1.

Таблица 1. Начальные условия модели.

Обозначение	Параметр	Значение
Параметры робота		
l	расстояние между правым и левым колёсами робота	0.381, м
V_{\max}	максимальная скорость (линейная) робота	1.2, м/с
OMEGAmax	максимальная скорость поворота (угловая) робота	0.05, с ⁻¹
Параметры двигателей		
K	постоянная двигателей, м/с*В	0.1, м/с*В
T	постоянная времени двигателей	0.1
Начальные условия движения объекта		
V_0	скорость движения объекта	0.7, м/с
fi0	угол отклонения движения объекта от оси OX	-, рад
x0	точка начала движения объекта по оси x	-, м
y0	точка начала движения объекта по оси y	-, м
Начальные условия движения робота		
V0_rob	скорость робота	1, м/с
fi0_rob	угол отклонения движения робота от оси OX	0, рад
x0_rob	точка начала движения объекта по оси x	0, м
y0_rob	точка начала движения объекта по оси y	0, м
Home	зона начального и конечного положения робота	10, м
Условия решения задачи		
dbound	расстояние между роботом и объектом, при котором задача считается решённой	0.21, м
fibound	угол между роботом и объектом, при котором задача считается решённой	0.349, рад
solve	флаг, показывающий, что задача решена (когда solve=1)	0

С помощью имеющихся датчиков робот измеряет расстояние d до движущегося объекта и азимутальный угол α между продольной осью робота (осью X_0) и направлением на объект. По углу α *нечёткий контроллер* вычисляет желаемые значения угловой скорости $\omega_{\text{ж}}$ движения мобильного робота. В качестве основы для решения данной задачи используется известный метод «собачьей погони», при которой в каждый момент времени робот стремится двигаться в точку, в которой в текущий момент времени находится объект.

Управляющие воздействия, подаваемые на приводы, вычисляются следующим образом:

$$U_R = V_{RЖ} k^{-1},$$

$$U_L = V_{LЖ} k^{-1},$$
(6)

где желаемые значения $V_{RЖ}$ и $V_{LЖ}$ скоростей ведущих колёс мобильного робота вычисляются на основании уравнений

$$V_{RЖ} = V_{Ж} + 0,5\omega_{Ж}l,$$

$$V_{LЖ} = V_{Ж} - 0,5\omega_{Ж}l.$$
(7)

Динамические свойства приводов можно учесть приближённо, используя уравнения

$$V_R(s) = \frac{k}{Ts + 1} U_R(s),$$

$$V_L(s) = \frac{k}{Ts + 1} U_L(s),$$
(8)

где k и T – коэффициент усиления и постоянная времени привода соответственно; $U_R(s)$ и $U_L(s)$ – изображения по Лапласу управляющих воздействий, подаваемых на правый и левый приводы соответственно.

Важно отметить, что скорости V_R и V_L должны быть ограничены. Во-первых, они не должны превосходить некоторой максимальной скорости V_{MAX} , которая, впрочем, больше скорости движения объекта. Следовательно в системе управления должно присутствовать нелинейное звено с насыщением. Во-вторых, угловая скорость ω должна быть ограничена, чтобы робот не перевернулся при повороте. То есть в соответствии с уравнениями (4) и (5), величина $(1/l) * (V_R - V_L)$ должна быть ограничена.

Таким образом, созданная в *Simulink* кинематическая модель мобильного робота, в которой осуществляется расчёт координат робота на плоскости XOY на основе сигналов, выдаваемых с контроллера, $V_{ref}=V_{Ж}$ и $OMEGA_{ref}=\omega$, выглядит следующим образом:

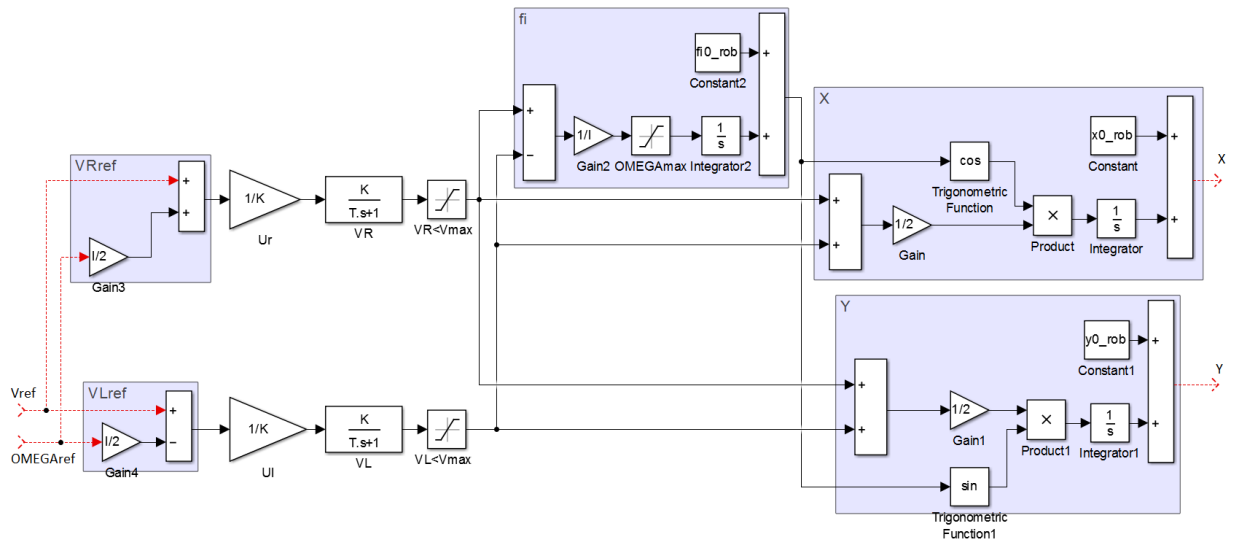


Рисунок 4. Кинематическая модель мобильного робота в *Simulink*.

2) Модель объекта (нарушителя) (файл **object_model.mdl**).

Координаты объекта (нарушителя) могут быть найдены по следующим уравнениям:

$$\dot{x} = V_0 \cos(fi_0)$$

$$\dot{y} = V_0 \sin(fi_0)$$
(9)

На рисунке 5 представлены блоки расчёта координат объекта на основе уравнений (9).

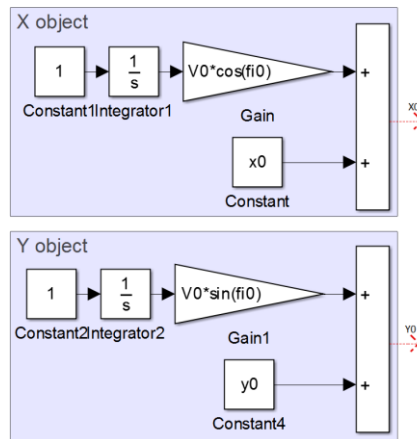


Рисунок 5. Блоки расчёта координат объекта.

3) Обратная связь по углу азимута объекта (нарушителя) относительно продольной оси робота (файл `deltafi_calculate.mdl`).

Как было сказано ранее, робот измеряет **угол азимута объекта относительно собственной системы координат** с помощью имеющихся датчиков (например, СТЗ). Для расчёта этого угла в модели необходимо рассчитать этот угол исходя из знания координат робота и объекта в каждый момент времени.

На рисунке 6 показано представление координат объекта в системах координат XOY и $X_0O_0Y_0$, где x_1 - координата проекции объекта (точки) на ось x_0 системы координат, связанной с роботом. Ось x_0 направлена по продольной оси робота. y_1 - координата проекции объекта на ось y_0 системы координат, связанной с роботом. Ось y_0 с осью x_0 образуют правую декартову систему координат.

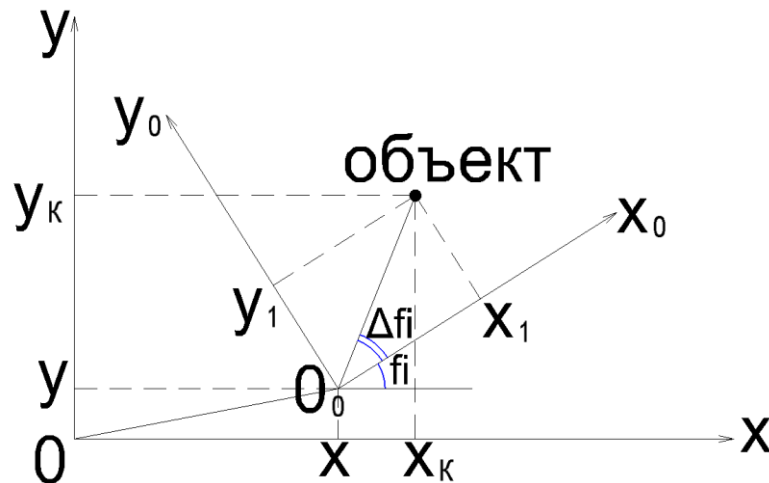


Рисунок 6. Системы координат XOY и $X_0O_0Y_0$.

Расчёт угла азимута объекта (нарушителя) относительно продольной оси робота можно получить по уравнению (10), где Δ (delta) - величина погрешности, равная 0.01, принятая для того, чтобы избежать деление на 0 при определении $\arctg(y_1/x_1)$.

$$\Delta fi = \begin{cases} \frac{\pi}{2} \text{sign} y_1 & \text{npu } |x_1| < \Delta \\ \arctg \frac{y_1}{x_1} & \text{npu } x_1 > \Delta \\ \pi - \arctg \frac{y_1}{|x_1|} & \text{npu } x_1 < -\Delta, y_1 > 0 \\ -\pi + \arctg \frac{|y_1|}{|x_1|} & \text{npu } x_1 < -\Delta, y_1 < 0 \end{cases}, \quad (10)$$

Для получения координат проекции объекта (точки) на оси x_0 и y_0 необходимо записать уравнения переноса (11) и поворота (12) системы координат из XOY в $X_0O_0Y_0$.

$$\begin{cases} x_K = x + x' \\ y_K = y + y' \end{cases} \quad (11)$$

$$\begin{cases} x' = x_1 \cdot \cos fi - y_1 \cdot \sin fi \\ y' = x_1 \cdot \sin fi + y_1 \cdot \cos fi \end{cases} \quad (12)$$

где $x_K = x_0$ и $y_K = y_0$ - координаты объекта (точки (x_1, y_1)) в системе координат XOY . После подстановки уравнения (12) в (11) и преобразований получаем уравнения (13):

$$\begin{cases} x_1 = (x_K - x) \cos fi + (y_K - y) \sin fi \\ y_1 = -(x_K - x) \sin fi + (y_K - y) \cos fi \end{cases} \quad (13)$$

На основании расчёта по системе уравнений (13) вычисляются координаты x_1 и y_1 . Далее по уравнению (10) вычисляется угол Δfi азимута объекта относительно продольной оси робота. На рисунке 7 представлена структурная схема в Simulink, позволяющая рассчитать угол Δfi .

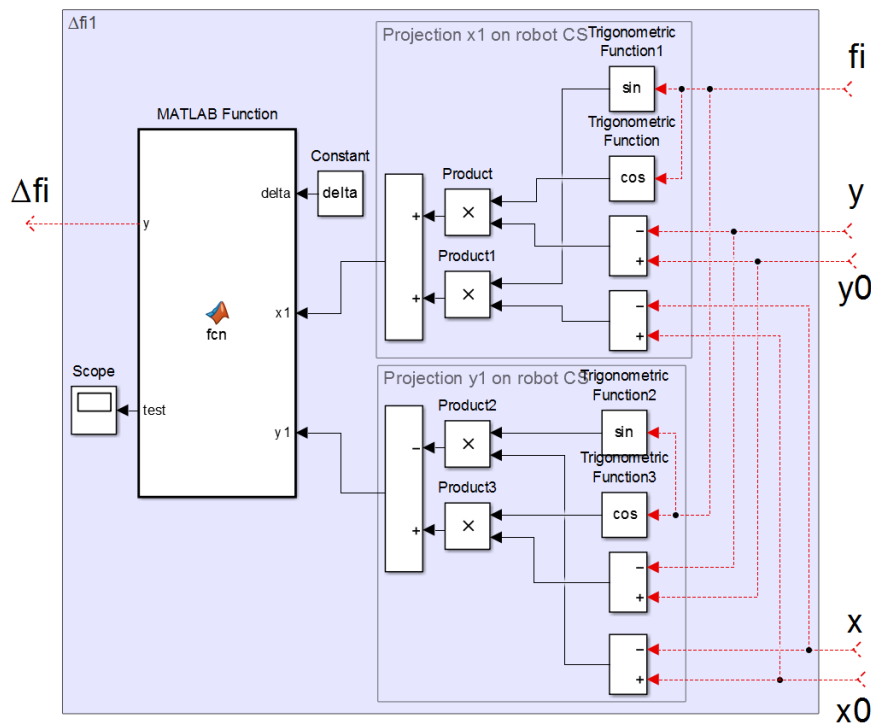


Рисунок 7. Блоки расчёта координат объекта.

Блок `fcn`, представленный на рисунке 7, осуществляет расчёт угла Δfi в соответствии с уравнением (10) по следующему программному коду:

```
function [y,test] = fcn(delta,x1,y1)
%#codegen
```

```

if (abs(x1)<=delta)
    u=(pi/2)*sign(y1);
    test=1;
else if (x1>delta)
    u=atan(y1/x1);
    test=2;
else if (y1>0)
    u=pi-atan(y1/abs(x1));
    test=3;
else
    u=-pi+atan(abs(y1)/abs(x1));
    test=4;
end;
end;
end;
y = u;

```

4) Обратная связь по расстоянию между объектом (нарушителем) и мобильным роботом (файл `d_calculate.mdl`).

Как было сказано ранее, робот измеряет **расстояние до объекта** с помощью имеющихся датчиков (например, СТЗ). Для расчёта этого расстояния необходимо составить следующее уравнение (14).

$$d = \sqrt{(x_0 - x)^2 + (y_0 - y)^2}, \quad (14)$$

где x_0 – координаты объекта по оси ОХ; y_0 – координаты объекта по оси ОУ; x – координаты робота по оси ОХ; y – координаты робота по оси ОУ. На рисунке 8 представлена реализация этого уравнения в *Simulink*.

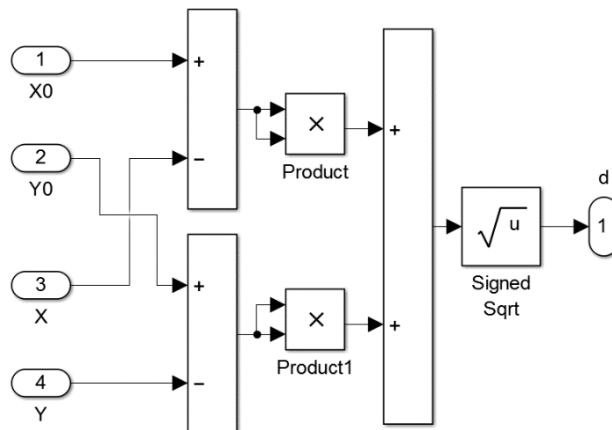


Рисунок 8. Блок расчёта расстояния от робота до объекта.

5) Определение времени выполнения условия решения задачи (файл `solving_condition.mdl`).

Исходя из постановки задачи, она считается решённой, если расстояние d от робота до объекта стало меньше $dbound$, а угол Δf_i азимута объекта относительно продольной оси робота не превышает $fibound$. На рисунке 9 представлены подсистемы *Solving condition* и *terminator & flag*, позволяющие определить время, когда условия задачи выполняются. Блок *if1* проверяет состояния сигналов d и f_i в соответствии с условием:

Если $d < |dbound|$ и $f_i < |fibound|$, то $solve = 1$ и $switch = -1$

Сигнал *switch* можно использовать для включения алгоритма захвата объекта роботом, для переключения робота на другую цель или для остановки робота. Сигнал *solve*

подаётся в рабочее пространство Matlab, где он хранится в виде массива данных, зависящих от времени. Когда solve становится равным нулю, условия решения задачи выполняются.

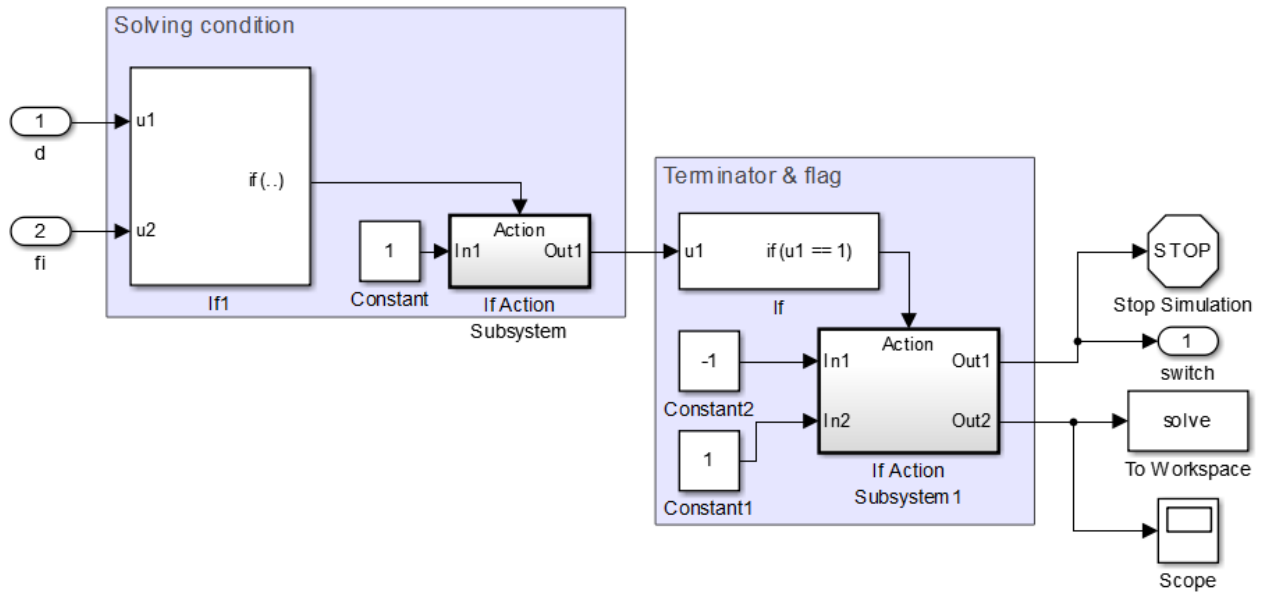


Рисунок 9. Определение времени выполнения условия решения задачи.

6) **Построение графика движения мобильного робота и объекта (файл graph_coordinates.mdl).**

На рисунке 10 представлена подсистема передачи координат движения объекта и робота в рабочее пространство Matlab. Таким образом, во время выполнения процесса моделирования системы строятся отдельные графики движения объекта и робота, а после выполнения процесса моделирования координаты передаются в рабочее пространство Matlab, где можно командой “plot(x_w,y_w,x0_w,y0_w)” построить общий график. С помощью блоков switch реализуется изменение координат объекта таким образом, что после захвата объекта роботом, координаты объекта становятся равными координатам робота.

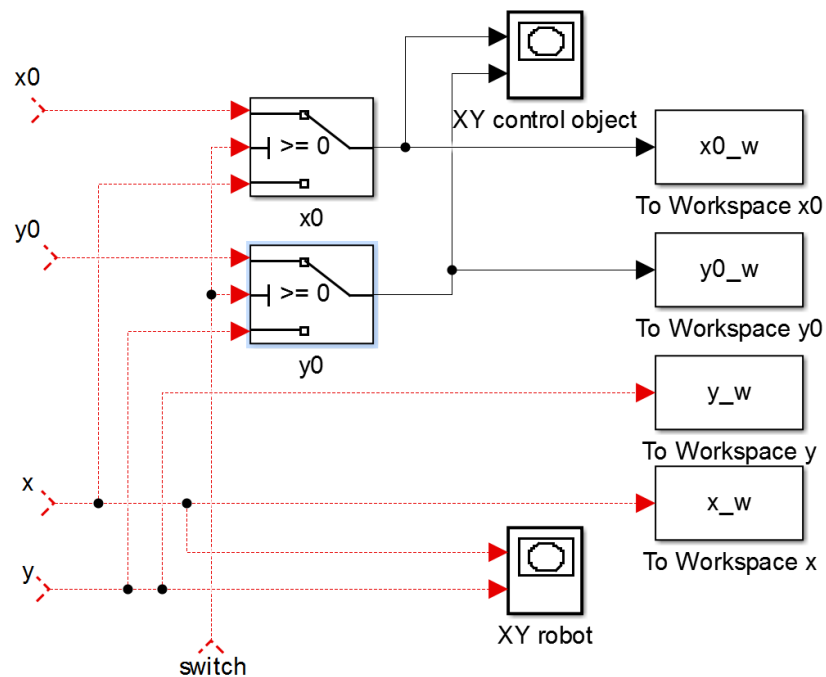
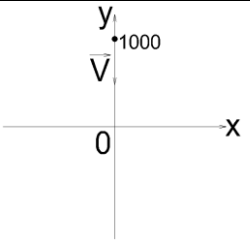
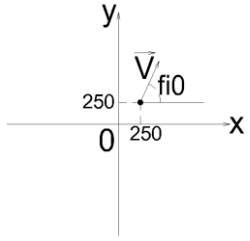
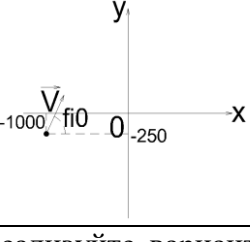


Рисунок 9. Подсистема построения графика движения мобильного робота и объекта. Таким образом, собрав все подсистемы, представленные в пунктах 1-6 в одну, получим модель, представленную на рисунке 10.

	$-\frac{\pi}{2}$	0	1000	
	$\frac{\pi}{3}$	250	250	
	$\frac{\pi}{3}$	-1000	-250	

- 5) Реализуйте вариант системы управления с нечётким контроллером, замкнув систему обратной связью по углу Δfi .

Пояснения: уже готовую модель нечёткого контроллера можно найти в файле “fuzzy_controller.slx”. Модель осуществляет управление только по одной переменной (по углу Δfi) и управляет переменной OMEGA. Модель построена на треугольных функциях принадлежности на основе 5 правил по таблице 3.

Таблица 3. База правил переменной **omega** нечёткого контроллера fuzzy_controller.

	fi				
	tri mf1	tri mf2	tri mf3	tri mf4	tri mf5
OMEGA	tri mf1	tri mf2	tri mf3	tri mf4	tri mf5

Структурная схема контроллера представлена на рисунке 11.

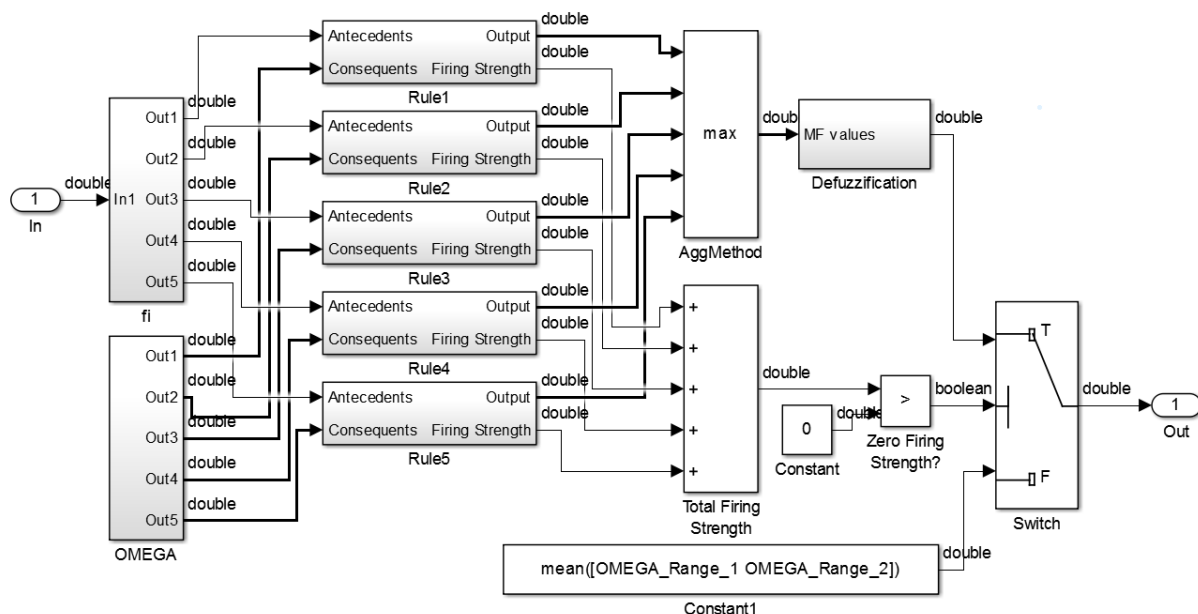


Рисунок 11. Модель контроллера нечёткой логики.

Данный контроллер реализован с помощью алгоритма Mamdani, используются следующие методы: метод логической конъюнкции – min, метод дефаззификации – centroid, метод импликации (активизации) – min, метод агрегирования – max.

В подсистеме φ заданы термы и их треугольные функции принадлежности для входной переменной φ . В подсистеме ω заданы термы и их треугольные функции принадлежности для переменной ω . Так как φ – входная переменная, то на вход каждой функции принадлежности этой переменной подаётся текущее значение φ , а на вход каждой функции принадлежности выходной переменной подаётся вектор, состоящий из $p1=200$ элементов, полученных равномерным разбиением диапазона изменения переменной ω , с помощью команды

`linspace(OMEGA_Range_1,OMEGA_Range_2,p1).`

В блоках, соответствующих правилам таблицы 3, осуществляется расчёт степени выполнения каждого правила для $p1$ элементов в текущий момент времени. В блоке Defuzzification осуществляется расчёт конечного значения выходной переменной. При этом, если ни одно правило не выполняется, то на выход подаётся значение, равное среднему арифметическому диапазона изменения переменной ω (благодаря условию Zero Firing Strength).

Для удобства параметры функций принадлежности задаются с помощью переменных, начальные значения которых представлены в файле “Graph_functions.m”. С помощью команд

`- plot(x,fi_1,x,fi_2,x,fi_3,x,fi_4,x,fi_5);`

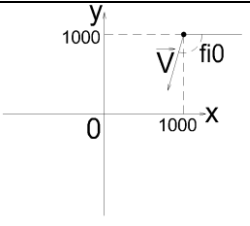
`- plot(x1,omega_1,x1,omega_2,x1,omega_3,x1,omega_4,x1,omega_5);`

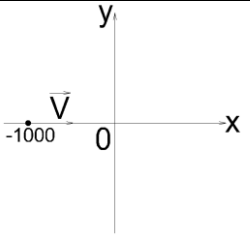
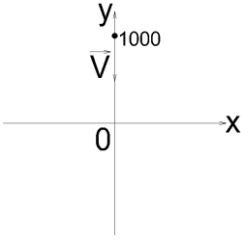
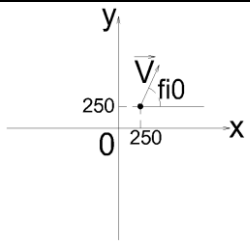
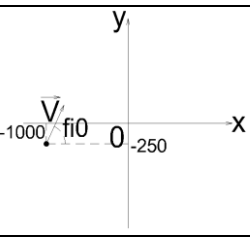
можно построить графики функций принадлежности переменных φ и ω .

Таким образом, для реализации нечёткого управления на основе 5 правил, выполните следующую последовательность действий:

- 1) Загрузите файл “fuzzy_controller.slx” и перенесите нечёткий контроллер в систему управления мобильным роботом вместо звена с коэффициентом усиления в цепи обратной связи по углу $\Delta\varphi$.
- 2) Загрузите из файла “Graph_functions.m” параметры треугольных функций принадлежности. Проверьте вид функций принадлежности с помощью команд `plot(x,fi_1,x,fi_2,x,fi_3,x,fi_4,x,fi_5)` и `plot(x1,omega_1,x1,omega_2,x1,omega_3,x1,omega_4,x1,omega_5);`
- 3) Запустите моделирование в течение 2000 с, и запишите время переходного процесса в соответствующие поля таблицы 4.

Таблица 3. Сравнение времени переходных процессов.

график начала движения объекта	Значения параметров движения объекта			время переходного процесса, с		
	φ_0 , рад	x_0 , м	y_0 , м	Классический вариант управления (из таблицы 2)	Контроллер нечёткой логики	Контроллер нечёткой логики с изменёнными ф-ми прин-ти (п. 5)
	$-\frac{7\pi}{12}$	1000	1000			

	0	-1000	0			
	$-\frac{\pi}{2}$	0	1000			
	$\frac{\pi}{3}$	250	250			
	$\frac{\pi}{3}$	-1000	-250			

- 4) Сравните быстродействие систем с нечёткой логикой и с пропорциональным звеном в обратной связи.
- 5) Измените параметры функций принадлежности, добившись минимального времени переходного процесса. Проведите несколько экспериментов с изменёнными параметрами и занесите время переходных процессов в последний столбец таблицы 3. Сделайте вывод о быстродействии данной системы управления по сравнению с классическим вариантом.
- 6) Добавьте в контроллер нечёткой логики управление скоростью мобильного робота в зависимости от расстояния d , чтобы при подходе к объекту робот замедлялся, что позволило бы осуществить более качественный захват. При этом скорость робота не должна опускаться ниже скорости движения объекта.
- 7) Модифицируйте систему управления, добавив несколько блоков так, чтобы робот, после захвата объекта, двигался в нулевую зону, определяемую переменной Home, и останавливался. Для этого можно добавить несколько блоков switch, переключающихся по сигналу switch с подсистемы terminator & flag.
- 8) Реализуйте систему управления мобильным роботом с помощью нечёткой логики по методу наведения с углом опережения. Идея заключается в том, в любой момент времени вектор скорости робота направлен не прямо на объект, а с некоторым постоянным углом опережения. Сравните времена переходных процессов при данном методе наведения робота с методом «собачьей погони».