Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2 по дисциплине «Основы искусственного интеллекта»

Тема Нечёткость

Студент Сапожков А.М,

Группа ИУ7-13М

Преподаватель Строганов Ю.В.

Москва, 2024

Содержание

ВВЕДЕНИЕ

В последние два десятилетия резко возрос интерес к различным аспектам проблемы интеллектуального управления. Одно из основных направлений, связанных с решением этой проблемы, состоит в использовании аппарата нечётких систем: нечётких множеств, нечёткой логики, нечёткого моделирования и т. п. Применение этого аппарата приводит к построению нечётких систем управления различных классов, позволяющих решать задачи управления в ситуациях, когда традиционные методы неэффективны или даже вообще неприменимы из-за отсутствия достаточно точного знания об объекте управления. [1]

Целью данной лабораторной работы является создание системы для следования в одномерном пространстве, при отсутствии данных о скорости «лидера», но с известным расстоянием до него.

Задачи данной лабораторной работы:

1) описать общие этапы нечёткого логического вывода;
2) предложить функции принадлежности термам числовых значений признаков, описываемых используемыми лингвистическими переменными;
3) предложить правила для нечёткого логического вывода;
4) описать алгоритм Ларсена для нечёткого логического вывода;
5) описать алгоритмы дефаззификации;
6) реализовать систему следования в одномерном пространстве с использованием нечётких переменных и правил для определения необходимой скорости автопилота;
7) измерить среднеквадратичную ошибку и время вычисления скорости автопилота.

1 Аналитическая часть

1.1 Общие этапы нечёткого логического вывода

На рисунке 1.1 представлена типовая структура нечёткой модели системы с двумя входами и одним выходом. [1]

На входы нечёткой модели поданы два чётких числовых значения x_1^*, x_2^*. Блок «**ФАЗЗИФИКАЦ (FUZZIFICATION)** вычисляет их степени принадлежности входным нечётким множествам A_i, B_j . Для выполнения указанной операции блок фаззификации должен иметь доступ к точно определённым функциям принадлежности $\mu_{A_i}(x1)$, $\mu_{B_j}(x2)$ входов. [1]

Вычисленные и представленные на выходе блока фаззификации степени принадлежности $\mu_{A_i}(x_1^*)$, $\mu_{B_j}(x_2^*)$ дают информацию о том, в какой степени числовые значения x_1^*, x_2^* принадлежат конкретным нечётким множествам, т. е. насколько эти величины являются малыми (A1, B1) или большими (A_2, B_2). [1]

Блок «**ВЫВОД**» **(INFERENCE)** на входе получает степени принадлежности $\mu_{A_i}(x_1^*)$, $\mu_{B_j}(x_2^*)$ и на выходе вычисляет так называемую результирующую функцию принадлежности выходного значения модели (рисунок 1.1). Данная функция обычно имеет сложную форму и определяется посредством вывода, который может быть осуществлён множеством способов. Для выполнения вычислений блок вывода должен включать в себя следующие строго определённые элементы: [1]

— база правил;

— механизм вывода;

— функции принадлежности выходного параметра y.

База правил содержит логические правила, которые задают имеющие место в системе причинно-следственные отношения между нечёткими значениями ее входных и выходных величин. [1]

Решение возложенной на блок вывода задачи, связанной с определением результирующей функции принадлежности $\mu_{res}(y)$, обеспечивается **механизмом вывода**, который состоит из следующих элементов:

IM1: элемент, вычисляющий степень выполнения каждого правила R_i в отдельности;

IM2: элемент, вычисляющий активизированные функции принадлежности заключений каждого правила R_i;

IM3: элемент, вычисляющий результирующую функцию принадлежности $\mu_{res}(y)$ выходного значения на основе активизированных заключений отдельных правил. [1]

Приведём пример механизма вывода для системы с двумя входами:

IM1: агрегация условий правил с использованием оператора PROD для пересечения множеств (И) и оператора MAX для объединения множеств (ИЛИ);

IM2: определение активизированных функций принадлежности заключений правил с использованием оператора импликации Мамдани;

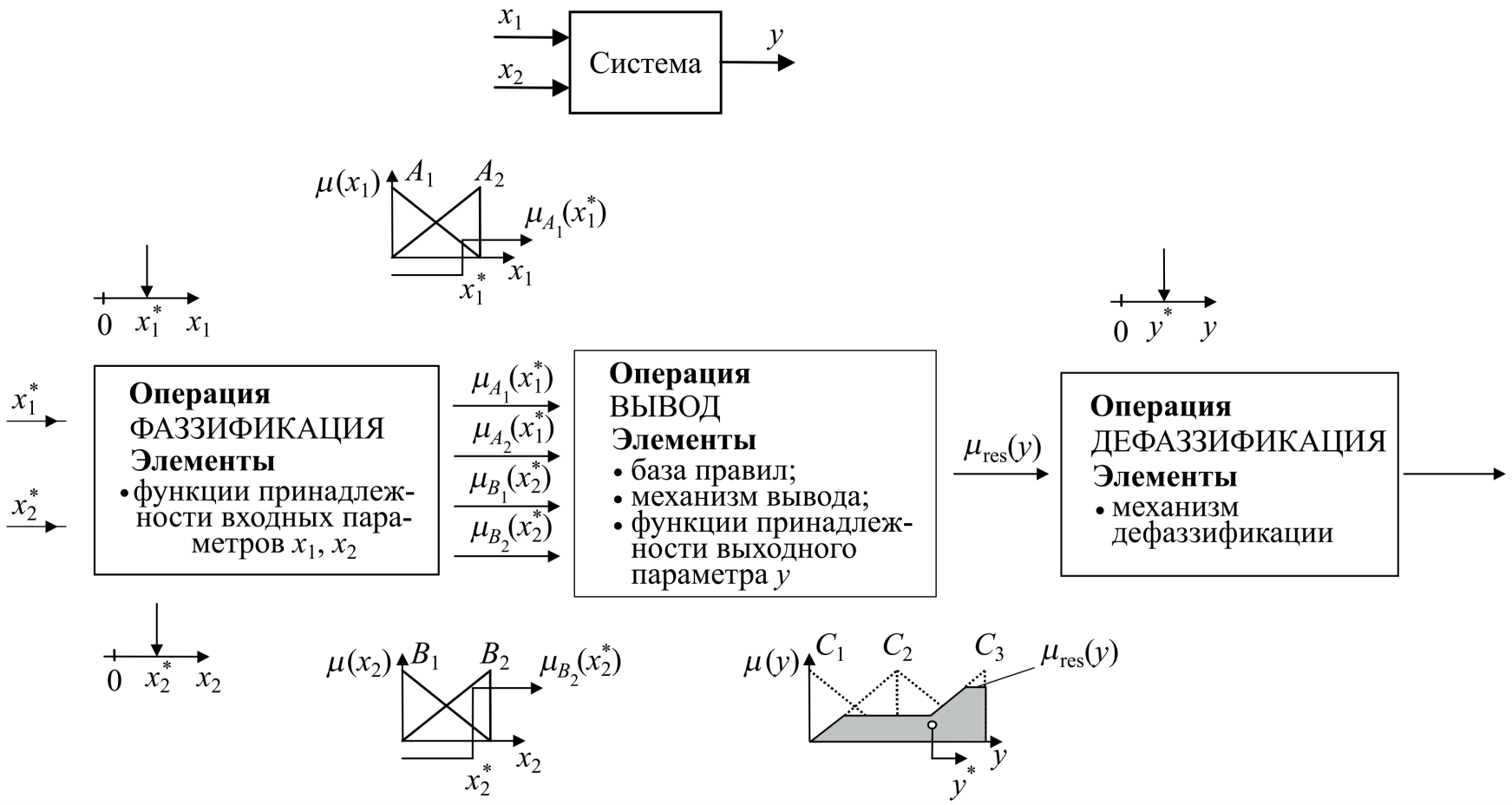

Рисунок 1.1 — Схема венгерского метода решения задачи о назначениях, часть 1

IM3: определение результирующей функции принадлежности $\mu_{res}(y)$ выходного значения (аккумуляция) с использованием оператора MAX. [1]

Блок «**ДЕФАЗЗИФИКАЦИЯ**» **(DEFUZZIFICATION)** на основе результирующей функции принадлежности $\mu_{res}(y)$ вычисляет чёткое числовое значение y^* выходного параметра, являющееся результатом для входных числовых значений x_1^*, x_2^*. Данная операция выполняется посредством **механизма дефаззификации**, который определяет метод вычисления. [1]

1.2 Методы дефаззификации

Наиболее известными методами дефаззификации являются

— метод среднего максимума (Middle of Maxima, MM);

— метод первого максимума (First of Maxima, FM);

— метод последнего максимума (Last of Maxima, LM);

— метод центра тяжести (Center of Gravity, CG);

— метод центра сумм (Center of Sums, CS);

— метод высот (Height, H). [1]

Далее перечисленные будут рассмотрены более подробно метод среднего максимума и метод центра тяжести. [1]

Метод среднего максимума (MM). Функцию принадлежности можно рассматривать как функцию, которая представляет информацию о сходстве между отдельными элементами множества и о наиболее типичном его элементе. С учётом функции принадлежности, соответствующей «среднему» значению роста, человек, имеющий рост 170 см, является типичным представителем данной категории роста (степень принадлежности равна 1), в то время как человека, имеющего рост 175 см, можно со степенью0.5 охарактеризовать как «среднего роста» и со степенью 0.5 — как «высокого». Иными словами, он частично соответствует как людям среднего роста, так и высоким людям. Таким образом, можно положить, что наиболее типичным представителем нечёткого множества B^*, полученного в результате вывода и задаваемого функцией принадлежности $\mu_{B^*}(y) = \mu_{res}(y)$, является значение y^*, имеющее максимальную степень принадлежности. Следует отметить, что множество таких значений часто может содержать более одного элемента и даже бесконечное число элементов. Решением в данной ситуации будет представление результирующего множества средним значением, получаемым по формуле

$$y^* = 0.5 \cdot (y_1^* + y_2^*). \tag{1.1}$$

Именно поэтому рассмотренный метод назван методом среднего максимума. [1]

Достоинством метода MM является простота вычислений, что допускает использование в системах управления более дешёвых микропроцессоров. Вместе с тем, простота вычислений достигается ценой определённых недостатков. [1]

Недостаток метода MM состоит в том, что на результат дефаззификации влияет только нечёткое множество Bj , имеющее наибольшую степень активизации — множества, активизи-

рованные в меньшей степени, никакого влияния на результат не оказывают. В свою очередь, это означает, что на результирующее значение y^* влияет только то правило, которое содержит это множество в своём заключении (часто это может быть только одно правило). Тем самым, дефаззификация становится «недемократичной», поскольку не все правила принимают участие в «голосовании». [1]

Метод центра тяжести (CG). В данном методе предполагается, что в качестве чёткого значения y^* для представления результирующего нечёткого множества B^*, задаваемого функцией принадлежности $\mu_{res}(y) = \mu_{B^*}(y)$, должна выбираться координата y_c центра тяжести фигуры, ограниченной графиком этой функции. Значение координаты центра тяжести C может быть найдено как отношение момента фигуры под кривой $\mu_{res}(y)$ относительно вертикальной оси $\mu(y)$ к площади этой фигуры: [1]

$$y^* = y_c = \frac{\int y\mu_{res}(y)dy}{\int \mu_{res}(y)dy} \tag{1.2}$$

Достоинством метода CG является то, что в дефаззификации участвуют все активизированные функции принадлежности заключений (все активные правила), т. е. метод центра тяжести является «демократичным» и обеспечивает более высокую чувствительность нечёткой модели к изменению входных сигналов, чем методы FM, LM и M. [1]

Недостатком метода CG является высокая стоимость вычислений, связанная с интегрированием поверхностей нерегулярной формы, особенно в случае использования функций принадлежности, не состоящих из прямолинейных участков (например, гауссовых функций). Для интегрирования необходимо определить точки пересечения отдельных составляющих функций принадлежности $\mu_{B_j}(y)$, разбить поверхность на секторы и выполнять интегрирование в пределах каждого из секторов. [1]

1.3 Алгоритм Ларсена

В этом методе в качестве нечёткой импликации для определения активизированных функций принадлежности заключений правил используется оператор PROD, а для аккумуляции — оператор MAX. В методе Ларсена правило задаётся следующим образом. [2]

$$R_i = if\ x_1\ is\ A_i, and\ x_2\ is\ B_i\ then\ z\ is\ C_i, i = 1, 2, \ldots, n. \tag{1.3}$$

Для R_i значение функции принадлежности будет определяться как

$$\mu_{R_i} = \mu_{A_i and B_i \to C_i}(x_1, x_2, z). \tag{1.4}$$

Использование оператора PROD в качестве нечёткой импликации означает, что функции принадлежности для заключений будут определяться следующим образом. [2]

$$\mu_{C_i} = \alpha_i \cdot \mu_{C_i}(z), \alpha_i = \mu_{A_i}(z) \wedge \mu_{B_i}(z). \tag{1.5}$$

Для случая нескольких правил

$$\alpha_i = \min \Big[\max_{x_1} \Big(\mu_{A_i}(z) \wedge \mu_{B_i}(z) \Big), \max_{x_2} \Big(\mu_{A_i}(z) \wedge \mu_{B_i}(z) \Big) \Big]. \tag{1.6}$$

Вывод

В данном разделе были описаны общие этапы нечёткого логического вывода, алгоритмы дефаззификации и алгоритм Ларсена для нечёткого логического вывода.

2 Конструкторская часть

2.1 Функции принадлежности

Для введения нечёткости предлагается рассматривать нечёткие переменные «скорость» (автопилота) и «расстояние» (до лидера) и задавать их с помощью функций принадлежности термам «очень низкая(ое)», «низкая(ое)», «средняя(ее)», «высокая(ое)» и «очень высокая(ое)». Графики предлагаемых функций принадлежности представлены на рисунках 2.1 и 2.2.

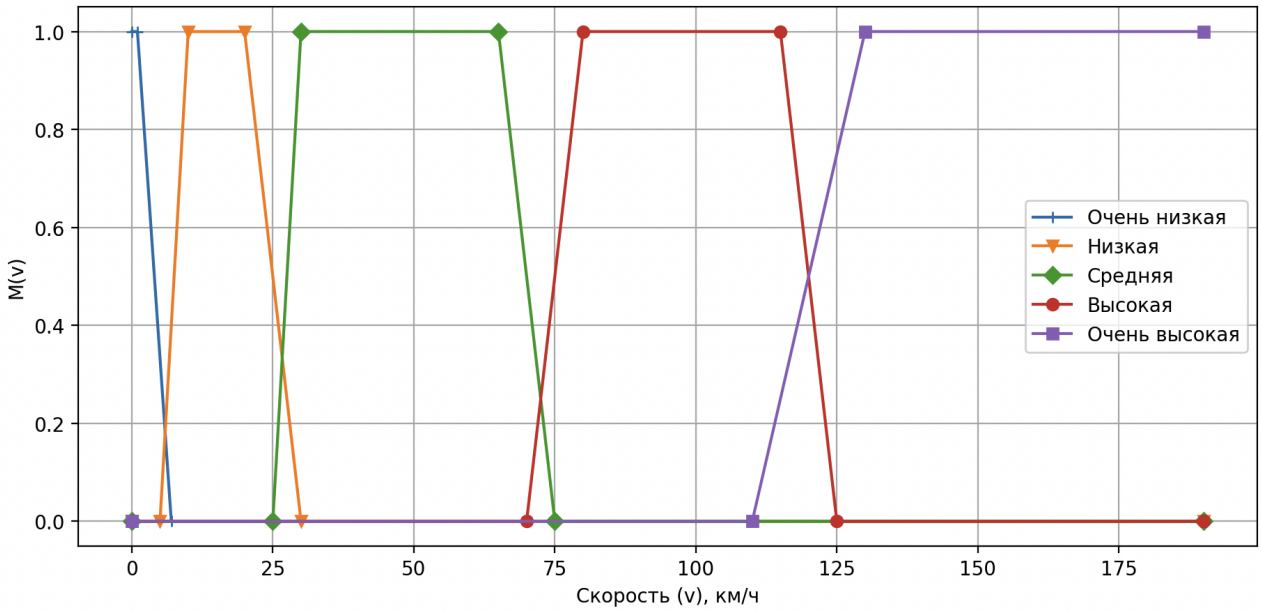

Рисунок 2.1 — Графики функций принадлежности термам значений скорости

Функция принадлежности расстояния терму «среднее» имеет единственный максимум, достигаемый при расстоянии равным 15 метров. Таким образом задаётся фиксированное расстояние до лидера, которого должен строго придерживаться автопилот.

2.2 База правил

Для определения рекомендуемой скорости автопилота была разработана база правил вида 1.3. Предлагаемые правила представлены в таблице 2.1

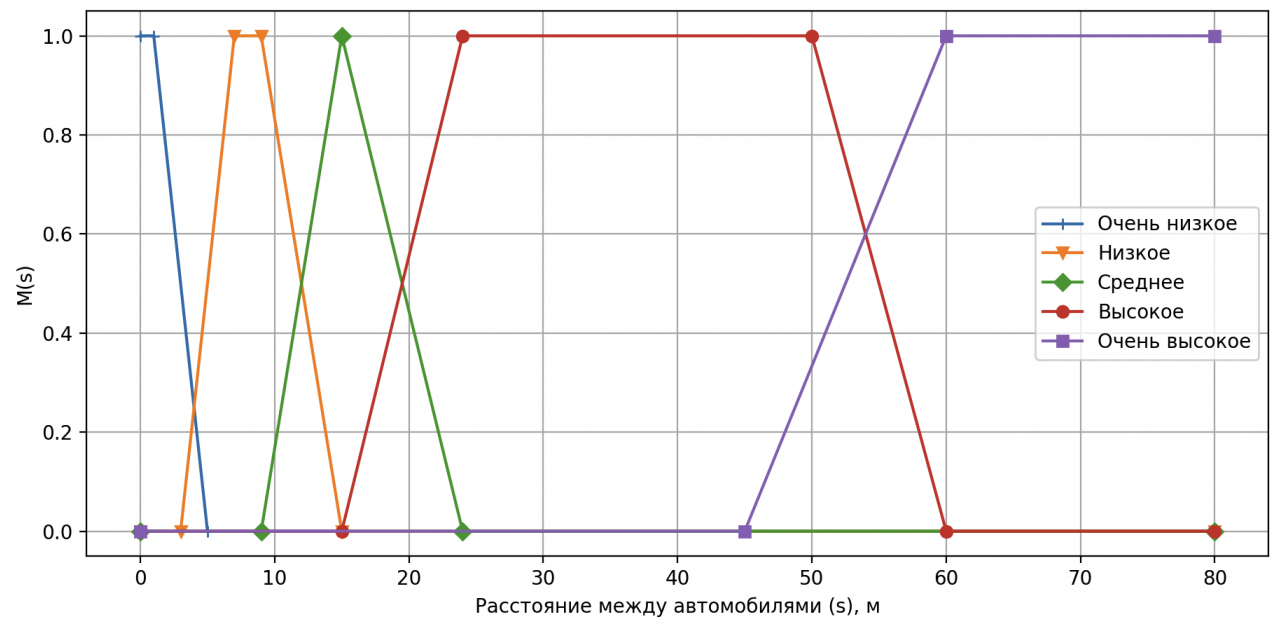

Рисунок 2.2 — Графики функций принадлежности термам значений расстояния между лидером и автопилотом

Таблица 2.1 — Зависимость рекомендуемой скорости автопилота от его текущей скорости и его расстояния до лидера

		Текущая скорость автопилота				
		Очень низкая	Низкая	Средняя	Высокая	Очень высокая
Расстояние до лидера	Очень низкое	Очень низкая	Очень низкая	Очень низкая	Очень низкая	Очень низкая
	Низкое	Очень низкая	Очень низкая	Низкая	Низкая	Низкая
	Среднее	Очень низкая	Низкая	Средняя	Средняя	Высокая
	Высокое	Низкая	Средняя	Высокая	Высокая	Очень высокая
	Очень высокое	Средняя	Средняя	Высокая	Очень высокая	Очень высокая

Вывод

В данном разделе были предложены функции принадлежности термам числовых значений скорости автопилота и его расстояния до лидера, а также правила для нечёткого логического вывода.

3 Технологическая часть

3.1 Требования к ПО

К программе предъявляются следующие требования:

— на вход подаётся текущая скорость автопилота и расстояние до лидера;

— запрещено определять ускорение, следует описать нечёткие переменные и правила для определения необходимой скорости автопилота;

— на выходе должна быть определена скорость автопилота, необходимая для поддержания заданного расстояния между лидером и автопилотом.

3.2 Средства реализации

В качестве языка программирования для реализации лабораторной работы был выбран язык Golang [3]. Выбор этого языка обусловлен наличием функциональности, требуемой для замеров процессорного времени и профилирования программ, а также наличием библиотек для организации нечётких вычислений. Для реализации системы следования использовалась библиотека fugologic [4], реализующая алгоритмы нечёткой логики.

3.3 Реализация алгоритмов

На листинге 3.1 представлена инициализация нечётких переменных и правил логического вывода.

Листинг 3.1 — Инициализация нечётких переменных и правил логического вывода

```
var (
  fvSpeed, fvDistance, fvNewSpeed *fuzzy.IDVal
  engine                          fuzzy.Engine
)

func init() {
  crispSpeed, _ := crisp.NewSetN(0, 180, 1/eps)
  fsSpeed, _ := fuzzy.NewIDSets(map[id.ID]fuzzy.SetBuilder{
    "Very low":  fuzzy.StepDown{A: 1, B: 7},
    "Low":       fuzzy.Trapezoid{A: 5, B: 10, C: 20, D: 30},
    "Medium":    fuzzy.Trapezoid{A: 25, B: 30, C: 65, D: 75},
    "High":      fuzzy.Trapezoid{A: 70, B: 80, C: 115, D: 125},
    "Very high": fuzzy.StepUp{A: 110, B: 130},
  })
  fvSpeed, _ = fuzzy.NewIDVal("Speed", crispSpeed, fsSpeed)

  crispDistance, _ := crisp.NewSetN(0, 100, 1/eps)
```

```
fsDistance, _ := fuzzy.NewIDSets(map[id.ID]fuzzy.SetBuilder{
  "Very low":  fuzzy.StepDown{A: 1, B: 5},
  "Low":       fuzzy.Trapezoid{A: 3, B: 7, C: 9, D: 15},
  "Medium":    fuzzy.Triangular{A: 9, B: 15, C: 24},
  "High":      fuzzy.Trapezoid{A: 15, B: 24, C: 50, D: 60},
  "Very high": fuzzy.StepUp{A: 45, B: 60},
})
fvDistance, _ = fuzzy.NewIDVal("Distance", crispDistance, fsDistance)

crispNewSpeed, _ := crisp.NewSetN(0, 180, 1/eps)
fsNewSpeed, _ := fuzzy.NewIDSets(map[id.ID]fuzzy.SetBuilder{
  "Very low":  fuzzy.StepDown{A: 1, B: 7},
  "Low":       fuzzy.Trapezoid{A: 5, B: 10, C: 20, D: 30},
  "Medium":    fuzzy.Trapezoid{A: 25, B: 30, C: 65, D: 75},
  "High":      fuzzy.Trapezoid{A: 70, B: 80, C: 115, D: 125},
  "Very high": fuzzy.StepUp{A: 110, B: 130},
})
fvNewSpeed, _ = fuzzy.NewIDVal("New speed", crispNewSpeed, fsNewSpeed
   )

bld := builder.Config{
  Optr:   fuzzy.OperatorZadeh{},
  Impl:   fuzzy.ImplicationProd,
  Agg:    fuzzy.AggregationUnion,
  Defuzz: fuzzy.DefuzzificationCentroid,
}.FuzzyAssoMatrix()
_ = bld.
Asso(fvSpeed, fvDistance, fvNewSpeed).
Matrix(
[]id.ID{"Very low", "Low", "Medium", "High", "Very high"}, // Speed
map[id.ID][]id.ID{ // Distance
  "Very low":  {"Very low", "Very low", "Very low", "Very low", "Very
      low"},
  "Low":       {"Very low", "Very low", "Low", "Low", "Low"},
  "Medium":    {"Very low", "Low", "Medium", "Medium", "High"},
  "High":      {"Low", "Medium", "High", "High", "Very high"},
  "Very high": {"Medium", "Medium", "High", "Very high", "Very high"
     },
},
)
```

```
  engine, _ = bld.Engine()
}
```

На листинге 3.2 представлена функция для расчёта скорости автопилота.

Листинг 3.2 — Функция для расчёта скорости автопилота

```
func Inference(speed, distance float64) float64 {
  res, _ := engine.Evaluate(fuzzy.DataInput{
    fvSpeed:    speed,
    fvDistance: distance,
  })

  return res[fvNewSpeed]
}
```

Вывод

В данном разделе были приведены детали реализации системы следования.

4 Исследовательская часть

4.1 Технические характеристики

Ниже приведены технические характеристики устройства, на котором выполнялось тестирование.

— Операционная система: macOS 14.6.1.

— Объём оперативной памяти: 18 Гб.

— Процессор: Apple M3 Pro.

Тестирование проводилось на ноутбуке, включённом в сеть электропитания. Во время тестирования ноутбук был нагружен только встроенными приложениями окружения, а также непосредственно системой тестирования.

4.2 Время выполнения реализаций алгоритмов

Алгоритмы тестировались с помощью запуска утилит тестирования программ на языке Golang [5] [6] [7]. Данные утилиты запускают бенчмарк для функции, реализующей нечёткий логический вывод, и замеряют процессорное время её выполнения [8]. Среднее время выполнения одного вывода составило 19.971 мс. На рисунке 4.1 приведены график зависимости среднеквадратичного отклонения расстояния (СКО) между лидером и автопилотом от скорости лидера. Эталонное расстояние между лидером и автопилотом было равно 15 метрам.

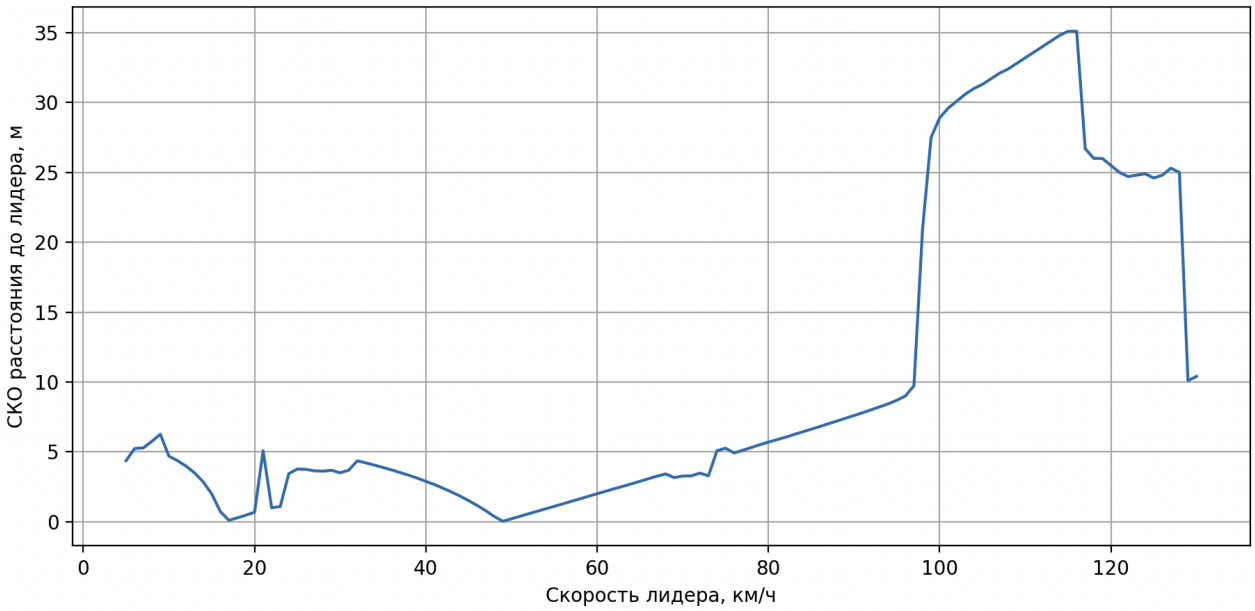

Рисунок 4.1 — График зависимости СКО расстояния между лидером и автопилотом от скорости лидера

Вывод

В данном разделе были приведены результаты замеров времени вычисления скорости автопилота и среднеквадратичной ошибки расстояния между лидером и автопилотом. Время выполнения одного нечёткого логического вывода в среднем составило 19.971 мс. Минимальное значение среднеквадратичного отклонения расстояния между лидером и автопилотом составило 0.1 м, максимальное — 35.1 м. Для повышения точности реализованной нечёткой модели необходимо либо переработать базу правил, дополнив её новыми правилами или изменив существующие, либо увеличить число нечётких переменных.

ЗАКЛЮЧЕНИЕ

В рамках лабораторной работы была создана система для следования в одномерном пространстве, при отсутствии данных о скорости «лидера», но с известным расстоянием до него. Все поставленные задачи были выполнены.

1) Описаны общие этапы нечёткого логического вывода;
2) Предложены функции принадлежности термам числовых значений признаков, описываемых используемыми лингвистическими переменными;
3) Предложены правила для нечёткого логического вывода;
4) Описан алгоритм Ларсена для нечёткого логического вывода;
5) Описаны алгоритмы дефаззификации;
6) Реализована система следования в одномерном пространстве с использованием нечётких переменных и правил для определения необходимой скорости автопилота;
7) Измерена среднеквадратичная ошибка время вычисления скорости автопилота.

Минимальное значение среднеквадратичного отклонения расстояния между лидером и автопилотом составило 0.1 м, максимальное — 35.1 м. Для повышения точности реализованной нечёткой модели необходимо либо переработать базу правил, дополнив её новыми правилами или изменив существующие, либо увеличить число нечётких переменных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Нечеткое моделирование и управление / А. Пегат ; пер. с англ. — 4-е изд., электрон. — М. : Лаборатория знаний, 2020. — 801 с.

2. Tejash U. Chaudhari, Vimal B. Patel, Rahul G. Thakkar, Chetanpal Singh Comparative analysis of Mamdani, Larsen and Tsukamoto methods of fuzzy inference system for students' academic performance evaluation. International Journal of Science and Research Archive, 2023, 09(01), c.s517–523.

3. The Go Programming Language documentation [Электронный ресурс]. — Режим доступа, URL: https://go.dev/ (дата обращения: 03.11.2024).

4. fuzzy logic inference system (in golang) [Электронный ресурс]. — Режим доступа, URL: https://github.com/sbiemont/fugologic (дата обращения: 03.11.2024).

5. Testing functions [Электронный ресурс]. — Режим доступа, URL: https://pkg.go.dev/cmd/go#hdr-Testing_functions (дата обращения: 03.11.2024).

6. Testing flags [Электронный ресурс]. — Режим доступа, URL: https://pkg.go.dev/cmd/go#hdr-Testing_flags (дата обращения: 03.11.2024).

7. pprof [Электронный ресурс]. — Режим доступа, URL: https://pkg.go.dev/runtime/pprof (дата обращения: 03.11.2024).

8. Diagnostics [Электронный ресурс]. — Режим доступа, URL: https://go.dev/doc/diagnostics (дата обращения: 03.11.2024).