

# ОПЫТ РАЗРАБОТКИ СИСТЕМЫ УПРАВЛЕНИЯ СОРТИРОВОЧНЫМ КОНВЕЙЕРОМ НА БАЗЕ СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Д.В. Бажутин, ст.преп., Denys.bazh@gmail.com  
Р.Б. Мирзалиев, студент rmirzaliyev1998@gmail.com

ГОУ ВПО «Донецкий национальный технический университет», г. Донецк.

Кафедра «Электропривод и автоматизация промышленных установок»

*Мирзалиев Р.Б. Статья посвящена вопросам разработки и реализации системы управления сортировочным конвейером на базе системы компьютерного зрения, основанной на распознавании объектов с помощью нейронных сетей. В ходе исследования были изучены вопросы реализации управления электроприводами с использованием микроконтроллеров, так же была создана собственная конфигурация нейронной сети для распознавания объектов в реальном времени посредством камеры.*

*Ключевые слова: одноконтурная система позиционирования, микроконтроллер stm32, система регулирования скорости, YOLO v3, нейронная сеть, система распознавания объектов, ШИМ, ПИ-РС, ПД-РП, датчик положения.*

*Mirzaliyev R.B. The article is devoted to the issues of development and implementation of the sorting conveyor control system based on the computer vision system based on object recognition with the help of neural networks. In the course of the research the questions of implementation of control of electric drives with the use of microcontrollers were studied, as well as an own configuration of the neural network for real-time recognition of objects by means of a camera was created.*

*Keywords: single-circuit positioning system, stm32 microcontroller, speed control system, YOLO v3, neural network, object recognition system, PWM, PI-RS, PD-RP, position transmitter.*

**Введение.** За последние годы все больше предприятий переходят от ручной сортировки к более эффективному автоматизированному решению, сортировочным системам. В статье будет рассмотрен опыт применения сортировочного конвейера. Применение конвейеров для сортировки позволяет существенно увеличить как рентабельность, так и производительность многих операций с грузами и представляют собой составную часть множества

распределительных центров и производств. Автоматическая сортировка груза, применяющаяся в процессе движения конвейера, значительно уменьшает затраты ручного труда для их дальнейшей обработки: упаковку на паллеты, к дальнейшей транспортировке и т. д.

А применение машинного зрения, а именно системы распознавания объектов, создает эффективную конкуренцию системам, основанным только на датчиках. Так как в таких системах анализируется сразу несколько характеристик объекта: цвет, форма, размеры, текстура, рисунок, узор и т.д. Так же такие системы легко дополняются и улучшаются теми же датчиками, для увеличения эффективности.

### ***Низкоскоростная сортировка***

Поперек конвейера, у пункта съема, располагают стационарный или подвижной рычаг-дефлектор либо пластину. Идущий по конвейеру ящик натывается на рычаг и сталкивается с конвейера в сторону на другой конвейер или в желоб.

Другой образец низкоскоростной сортировочной машины – конвейер, оснащенный сталкивателем. Подвижной рычаг с вертикальной пластиной на конце устанавливают сбоку, напротив места съема изделия. Когда продукт подходит к месту съема, механизм сталкивает его с конвейера в поперечном направлении. Сталкиватели работают со скоростью 20...100 ед./мин.

### ***Сортировка со средней скоростью***

Устройства с поднимающимися рабочими органами сдвигают грузы с ленточных конвейеров и обеспечивают точную ориентацию груза. Когда ящик подходит к месту съема, под его дном поднимаются колеса или ролики, приподнимают ящик над конвейером и транспортируют его в сторону. Максимальная скорость сортировки – 60...150 ед./мин.

Роликовые сортировочные машины работают аналогично, обеспечивая скорость сортировки до 150 ед./мин, вместо роликов могут использоваться ленты.

### ***Сортировка с высокой скоростью***

**\*\***В сортировочной машине с наклоняющимися лотками каждый транспортирующий элемент несет лоток, на который и укладывается поступающий груз. Лотки с грузами движутся по замкнутой траектории, пока груз не достигнет соответствующего места съема. После этого лоток наклоняется, и груз соскальзывает с него обычно на желоб или боковой конвейер. Скорость сортировки – до 365 ед./мин.

Сортировочная машина с поперечным ленточным конвейером-съемщиком работает по такому же принципу, но вместо лотка каждый транспортирующий элемент несет секцию ленточного конвейера, которая

приводится в движение небольшим электродвигателем. Скорость сортировки – свыше 670 ед./мин.

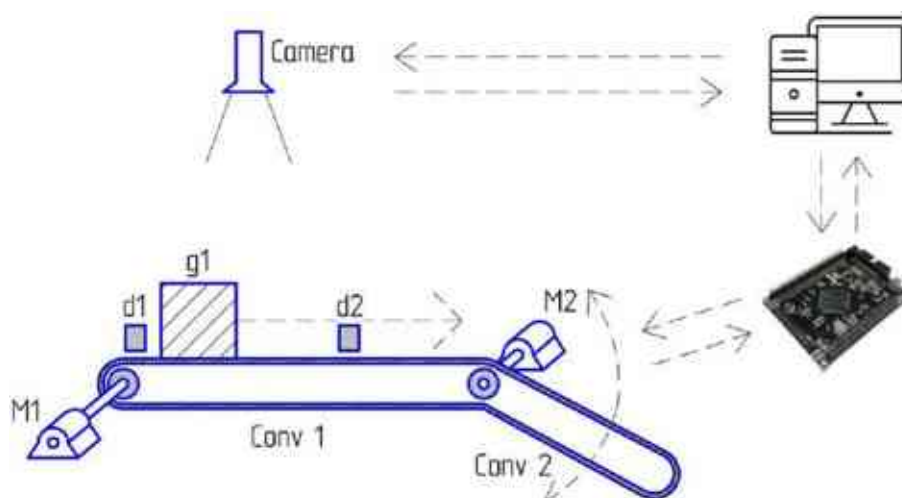
В сортировочной машине с раздвижными лотками каждый транспортирующий элемент несет плоский лоток, разделенный на две равные части. Когда груз достигает места съема, обе половины лотка раздвигаются, и груз падает в желоб или контейнер, расположенный под сортировочной машиной.

Машины с поперечными ленточными конвейерами-съемщиками надежней и точней других выполняют сортировку, так как их рабочие органы приводятся электродвигателями, а не работают под действием тяжести.

В машинах с отклоняющими кулачками основу сортировочной машины составляет длинный металлический пластинчатый конвейер. На каждой пластине установлена небольшая резиновая деталь, которая называется кулачком. Конвейер транспортирует коробки, и когда очередная коробка достигнет места съема, несколько кулачков приводятся в действие. Кулачки перемещаются по пластинам конвейера и сталкивают коробку с конвейера вбок. Скорость сортировки от 200 - 300 ед./мин.

### **Описание проекта лабораторной установки.**

Система сортировки, рассмотренная в данной статье, это сортировочный конвейер с позиционной направляющей. Управление положением направляющей, как и всего алгоритма движения конвейера происходит с помощью контроллера stm32f407vzt6. Контроллер собирает информацию с датчиков и системы распознавания, основываясь на этой информации формирует сигналы, управляющие приводами конвейера.



*Рисунок 1 – Упрощенная конструкция лабораторной установки*

### **Алгоритм управления приводом ленты**

Алгоритм управления строится на основе одноконтурной системы регулирования скорости (ПИ-РС) с ограничением рывка. Ограничение рывка реализуется с помощью (ЗИ) задатчика интенсивности управляющий сигнал

которого подается на (ПИ-РС) пропорционально интегральный регулятор скорости, оба эти элемента реализованы на уровне программы и вшиты в контроллер. Который в свою очередь поддерживает скорость путем подачи управляющих сигналов (ШИМ) на драйвер. Драйвер основанный на мостовой схеме формирует сигнал управляющий двигателем. Так же на двигателе присутствует датчик Холла, сигнализирующий о скорости оборотов, он создает обратную связь обеспечивающую работу ПИ регулятора скорости.

**Синтез одноконтурной системы управления с дискретным ПИ-РС**

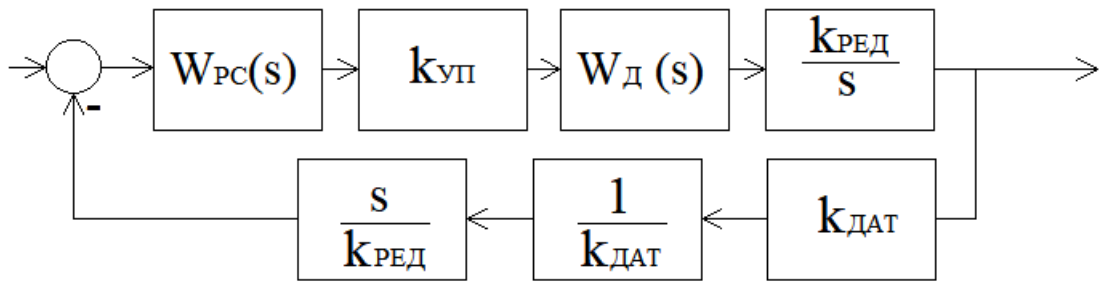


Рисунок 2 – Обобщенная структурная схема РКС

ПФ – разомкнутой системы РС

$$W_{PKC} = W_{PC}(s)W_{УП}(s)W_{Д}(s)W_{РЕД}(s)W_{ДАТ}(s)W_{БВУ}(s)W_{БВС}(s) = \frac{k_{ТП}k_{\delta}}{(T_{ТП}s + 1)(T_M s + 1)}$$

Примем постоянную якорной цепи и тиристорного преобразователя как незначительные, тогда:

$$T_C = T_{ТП} + T_Я$$

Тогда РКС по модульному оптимуму

$$W_{PKC}(s) = \frac{1}{2T_C s(T_{ТП}s + 1)}$$

В итоге ПФ-РС имеет вид

$$W_{PC}(s) = \frac{T_M s + 1}{2T_C k_{ТР} k_{\delta} s} = \frac{T_M}{2T_C k_{ТР} k_{\delta}} + \frac{1}{2T_C k_{ТР} k_{\delta} s}$$

Пропорциональный коэффициент  $k_p$  и интегральный  $k_i$  имеют вид:

$$k_p = \frac{T_M}{2T_C k_{ТР} k_{\delta}}; \quad k_i = \frac{1}{2T_C k_{ТР} k_{\delta}};$$

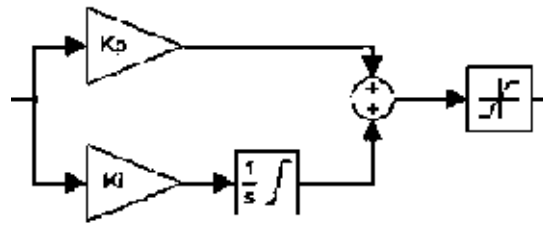


Рисунок 3 – Непрерывный ПИ Регулятор скорости

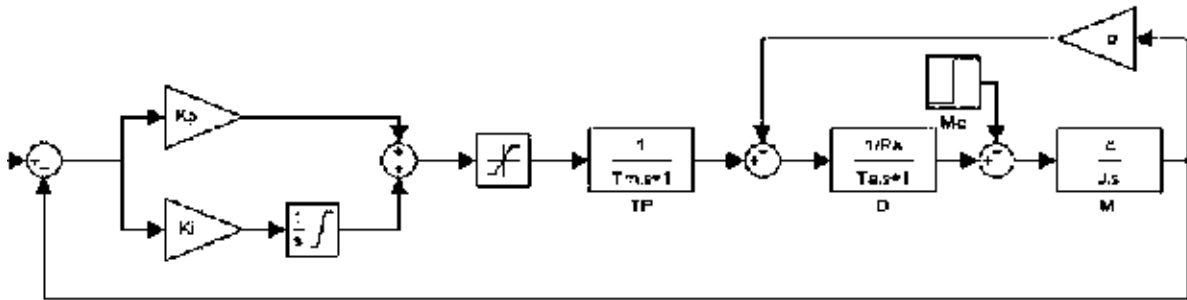
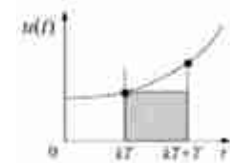


Рисунок 4 - Полная модель одноконтурной САУ скорости, составленная в пакете MATLAB Simulink

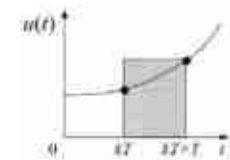
Приведенный регулятор необходимо привести к дискретной форме так как он описывает непрерывные преобразования.

Существует несколько основных подстановочных методов дискретизации непрерывных объектов.

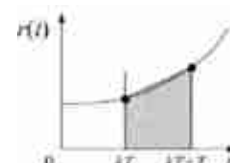
Метод Эйлера  $s = \frac{z-1}{T} = \frac{1-z^{-1}}{Tz^{-1}}$



Метод обратных разностей  $s = \frac{z-1}{Tz} = \frac{1-z^{-1}}{T}$



Метод Тастина  $s = \frac{2}{T} \cdot \frac{z-1}{z+1} = \frac{2}{T} \cdot \frac{1-z^{-1}}{1+z^{-1}}$



Используя метод Тастина, или как его еще называют методом трапеций дискретный ПИ регулятор будет иметь вид:

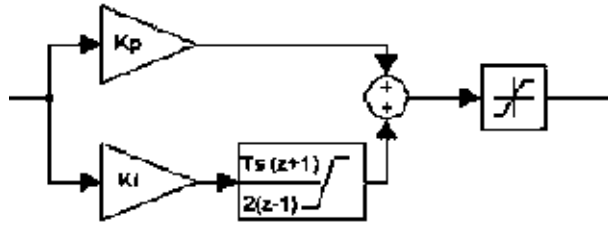


Рисунок 5 – Дискретный ПИ-РС

$$W_p(s) = \frac{k_p s + k_i}{s}$$

$$W(z) = \frac{k_p s + k_i}{s} \Big|_{s = \frac{2z-1}{Tz+1}} = \frac{k_p \frac{2z-1}{Tz+1} + k_i}{\frac{2z-1}{Tz+1}} = \frac{2k_p z - k_p + k_i Tz + k_i}{2z-1} = \frac{U_{вых}}{U_{вх}}$$

$$U_{вых}(z) = 2k_p U_{вх}(z) + k_p z^{-1} U_{вх}(z) - k_i T U_{вх}(z) - k_i z^{-1} U_{вх}(z) - 2z^{-1} U_{вых}(z) =$$

$$= U_{вх}(z)(2k_p - k_i T) + z^{-1} U_{вх}(z)(k_p - k_i) - 2z^{-1} U_{вых}(z)$$

Разложим выражение на простые множители и воспользуемся правилами z-преобразования, чтобы перейти от операторной формы к виду решетчатой функции:

$$U_{вых}(nT) = U_{вх}(nT)(2k_p - k_i T) + U_{вх}(nT - T)(k_p - k_i) - 2U_{вых}(nT)$$

### Управление приводом позиционной направляющей

Основной регулируемой координатой в таких системах является положение (перемещение) исполнительного органа (ИО) рабочей машины. В качестве датчиков положения (ДП) принимают устройства двух типов, устанавливаемые на валу двигателя или ИО – аналоговые или дискретные.

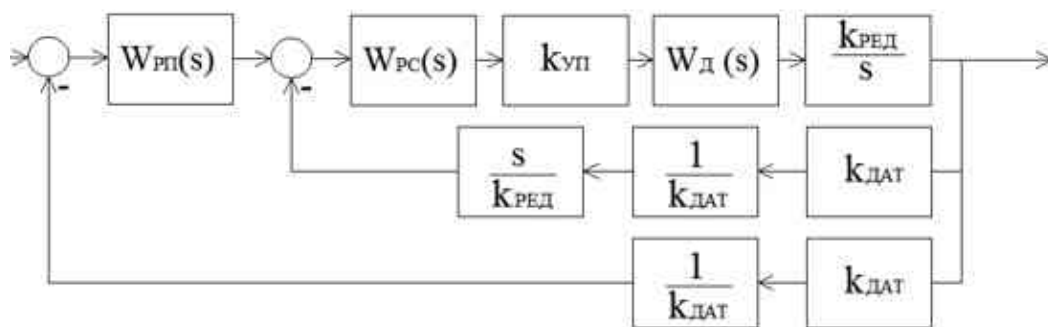


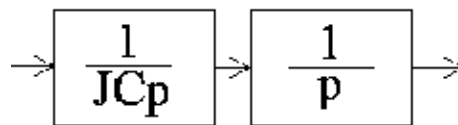
Рисунок 6 – Обобщенная структурная схема системы регулирования положения (системы слежения)

Построение системы регулирования положения заключается в организации внешнего по отношению к контуру скорости контура регулирования, замкнутого по перемещению вала двигателя или ИО. В качестве системы регулирования скорости (СРС) принципиально может использоваться любая из возможных, в нашем случае используем вышеописанную систему для управления лентой конвейера. Выход регулятора положения (РП) ограничивается на уровне, соответствующем максимально допустимому значению скорости  $\omega_{\max}$  (с учетом запаса на перерегулирование в динамике, который следует выбирать с учетом величины статической нагрузки, что обычно составляет 1...5%).

Наибольшее распространение имеют систем регулирования положения, построенные на базе однозонных СРС, поэтому максимально допустимым значением скорости принимаем значение скорости идеального холостого хода с учетом запаса на перерегулирование в динамике.

### ***Синтез регулирования положения следящей системы***

Примем допущение, что на начальном интервале времени переходная функция апериодического звена изначально отличается от таковой интегрирующего звена. Исходя из этого структурную схему объекта регулирования можно представить в таком виде:

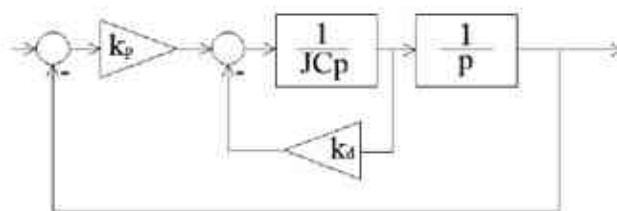


*Рисунок 7 – Объект управления*

Используя ПД регулятор положения можно считать эталонной моделью объекта регулирования колебательное звено с постоянной времени  $T$ , отвечающей требованиям к быстродействию электропривода, и коэффициентом демпфирования  $\xi = \frac{\sqrt{2}}{2}$ .

$$K(s) = \frac{1}{T^2 s^2 + 2\xi Ts + 1}$$

Тогда получим структурную схему с принятыми допущениями:



*Рисунок 8 – Следящая система позиционирования с ПД-регулятором*

ПФ такой системы имеет вид:

$$K(s) = \frac{1}{\frac{CJ}{k_p} s^2 + \frac{k_d}{k_p} s + 1}$$

Исходя из требования совпадения в численном виде коэффициентов передаточных функций, получим следующие значения коэффициентов ПД-регулятора:

$$k_p = \frac{CJ}{T^2}; \quad k_d = 2\xi T k_p;$$

Исходя из того, что ПД регулятор положения фактически задается коэффициентами не повышая порядок системы, то и в дискретизации данный регулятор не нуждается. В этом случае в дискретной системе будут использоваться те же коэффициенты из непрерывной системы.

### Задатчик положения

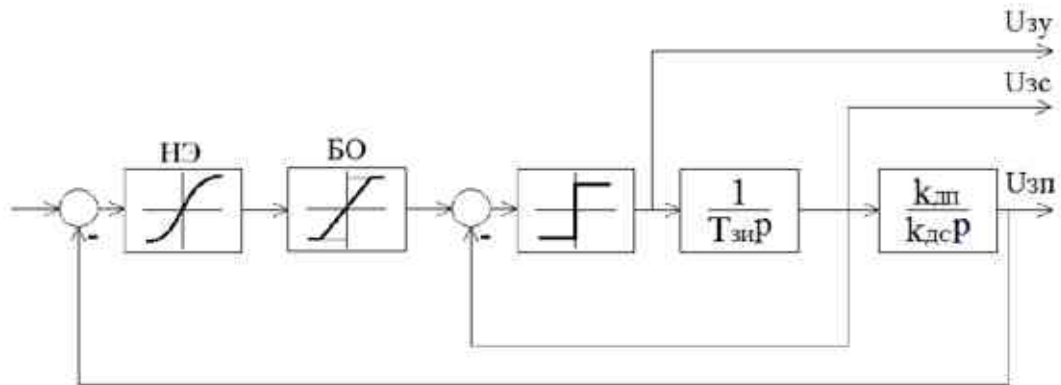


Рисунок 9 – Структурная схема задатчика положения

Учитывая, что параметры ЗИ

$$\frac{U_0}{T_{ЗИ}} = \frac{k_{д.с} \omega_3}{t_{р.ж}}$$

Выходной сигнал ЗП:

$$u_{з.н}(t) = k_{д.с} \frac{k_{д.н}}{k_{д.с}} \int \omega_3 dt = k_{д.н} \varphi_3(t)$$

Это позволяет задать идеальную диаграмму изменения регулируемой координаты (положения) при безынерционной подчиненной системе регулирования положения.

При этом отпадает необходимость использования ЗИ на входе контура регулирования скорости.



Важным достоинством применения ЗП является возможность получения производных от задающего воздействия без применения операции дифференцирования:

$$u_{3.c}(t) = \frac{k_{\partial.c}}{k_{\partial.n}} \cdot \frac{du_{3.n}}{dt} = k_{\partial.c} \omega_3(t) \text{ – задание на скорость}$$

$$u_{3.y}(t) = \frac{k_{\partial.c} T_{3И}}{k_{\partial.n}} \cdot \frac{d^2 u_{3.n}}{dt^2} = k_{\partial.c} T_{3И} \varepsilon_3(t) \text{ – задание на ускорение}$$

С помощью этих сигналов реализуют принцип комбинированного управления, позволяющий обеспечить достаточно высокое быстродействие контура регулирования положения (КРП). Данный принцип при регулировании положения оказывается очень важным (так как эквивалентная инерционность системы регулирования положения уже довольно существенна).

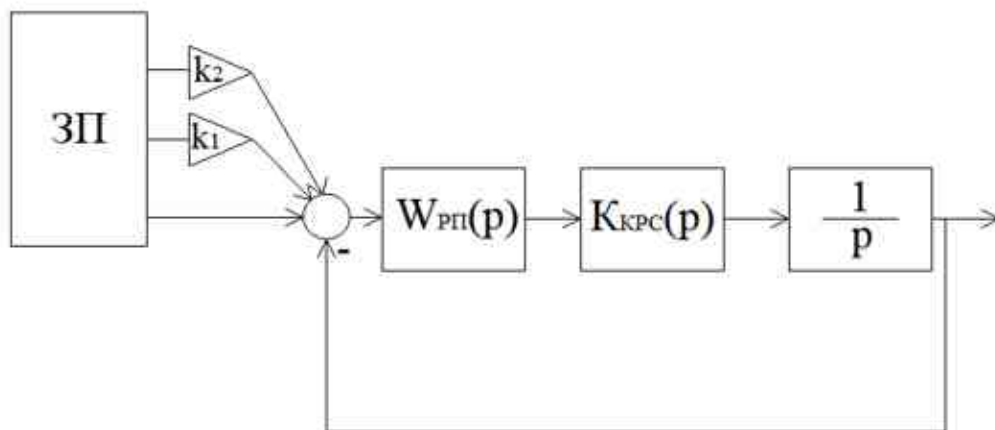


Рисунок 10 – Структурная схема СРП с ЗП

При этом все регуляторы системы работают в линейном режиме, а ограничение скорости и ускорения осуществляется за счет специального формирования задающего сигнала на КРП:

$$u_{3.П} = u_{3.n} + k_1 u_{3.c} + k_2 u_{3.y}$$

Используя метод Эйлера структурная схема дискретного ЗП будет иметь вид:

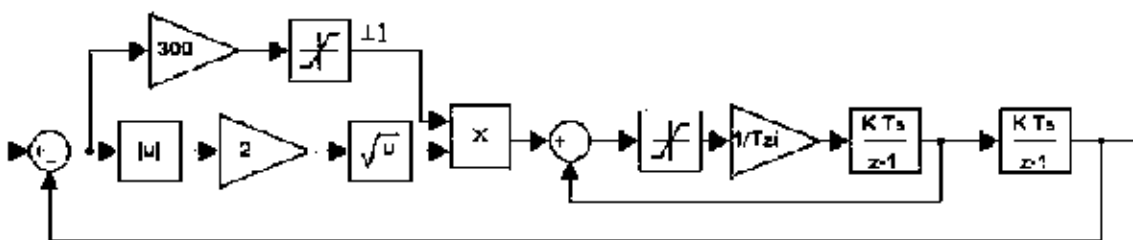


Рисунок 11 – Структурная схема дискретного ЗП

## **Описание работы алгоритма распознавания образов.**

Система сортировки строится на основе распознавания объектов алгоритм которой основан на регрессии. Вместо выбора интересных частей изображения они определяют классы за один прогон алгоритма. Двумя наиболее известными примерами из этой группы являются алгоритмы семейства YOLO (You Only Look Once) и SSD (Single Shot Multibox Detector). Обычно используются для обнаружения объектов в реальном времени, поскольку, как правило, они жертвуют точностью ради значительного повышения скорости.

YOLO – первая нейронная сеть, которая распознавала объекты в реальном времени на мобильных устройствах. Отличительная особенность: различение объектов за один прогон (достаточно один раз посмотреть). То есть в архитектуре YOLO нет явных циклов «for», из-за чего сеть работает быстро. Например, такая аналогия: в NumPy при операциях с матрицами тоже нет явных циклов «for», которые в NumPy реализуются на более низких уровнях архитектуры через язык программирования C. YOLO использует сетку из заранее заданных окон. Чтобы один и тот же объект не определялся многократно, используется коэффициент перекрытия окон (IoU, Intersection over Union). Данная архитектура работает в широком диапазоне и обладает высокой робастностью: модель может быть обучена на фотографиях, но при этом хорошо работать на рисованных картинах.

**SSD – Single Shot MultiBox Detector:**

Single Shot – означает, что задачи локализации и классификации объектов выполняются за один проход сети.

MultiBox – так называется методика поиска ограничивающего прямоугольника.

Detector – нейронная сеть работает как классификатор (детектор) объектов.

Подход SSD был опубликован в конце ноября 2016 года и привел к новому рекорду с точки зрения производительности и точности для задач обнаружения объектов. С её помощью мы можем классифицировать за 1 проход несколько разных объектов и на одном кадре определять и людей, и животных и транспорт и т.п. Программа подсчета объектов на видео, например, может считать разные типы объектов и учитывать их нахождение в определенной зоне.

Нейронная сеть также способна “сопровождать” объекты, таким образом определив на одном из кадров машину как Car\_001 она будет Car\_001 и на других кадрах, тем самым вы будете понимать, что это одна и та же машина, а не каждый раз новая.

YOLO – как уже было сказано это алгоритм, основанный на регрессии, вместо выбора интересных частей изображения и классификации их он определяет классы путем сканирования всего изображения за один прогон алгоритма. Процесс сканирования всего изображения начинается с предварительно определенного окна  $n*n$  пикселей, которое выдает логический результат, который принимает значение true, если указанный объект присутствует в отсканированной части изображения, и значение false, если это не так. После сканирования всего изображения алгоритм увеличивает размер окна, которое используется для повторного сканирования изображения. Методы на основе деформируемых моделей частей для обнаружения объектов (DPM), используют эту технику, которая называется скользящим окном. Модель YOLO была разработана для нейронной сети с открытым кодом на основе DarkNet. Но одним DarkNet реализация не ограничивается, YOLO можно реализовать и на Keras и на том же Tensorflow. Но в рамках поставленной задачи DarkNet полностью отвечает требованиям, так как полностью написан на C, а значит быстродействие в рамках реального железа гарантировано.

### *Подготовка набора данных*

Для тренировки сети на определенные объекты необходимо создать набор данных в котором будут храниться фотографии на которых находятся необходимые объекты в формате .jpg и соответствующие им по названию .txt файлы в которых будут храниться координаты областей на которых изображены интересующие нас объекты. Для этого используем открытый git-репозиторий LabelImg (<https://github.com/tzutalin/labelImg.git>), эта программа, специально написанная для решения подобного рода задач.

Файл .txt хранит в себе данные для каждого объекта в новой строке: `<object-class> <x> <y> <width> <height>`, где:

`<object-class>` - целое число объектов для распознавания от 0 до (N-1)

`<x> <y> <width> <height>` - значения с плавающей запятой относительно ширины и высоты изображения, могут быть равны от 0,0 до 1,0

`<x> <y>` - это центр прямоугольника.

Пример такого файла:

```
2 0.703750 0.208437 0.430833 0.353125
3 0.276667 0.330937 0.395000 0.291875
0 0.572917 0.513437 0.222500 0.096875
1 0.479583 0.687187 0.165833 0.233125
```

Рекомендую сделать как можно больше фотографий объектов с разных ракурсов, при разном освещении, разном фоне, положительно отразится и наличие похожих вещей на заднем плане для того чтоб более наглядно показать сети детали и отличительные особенности заданных объектов.

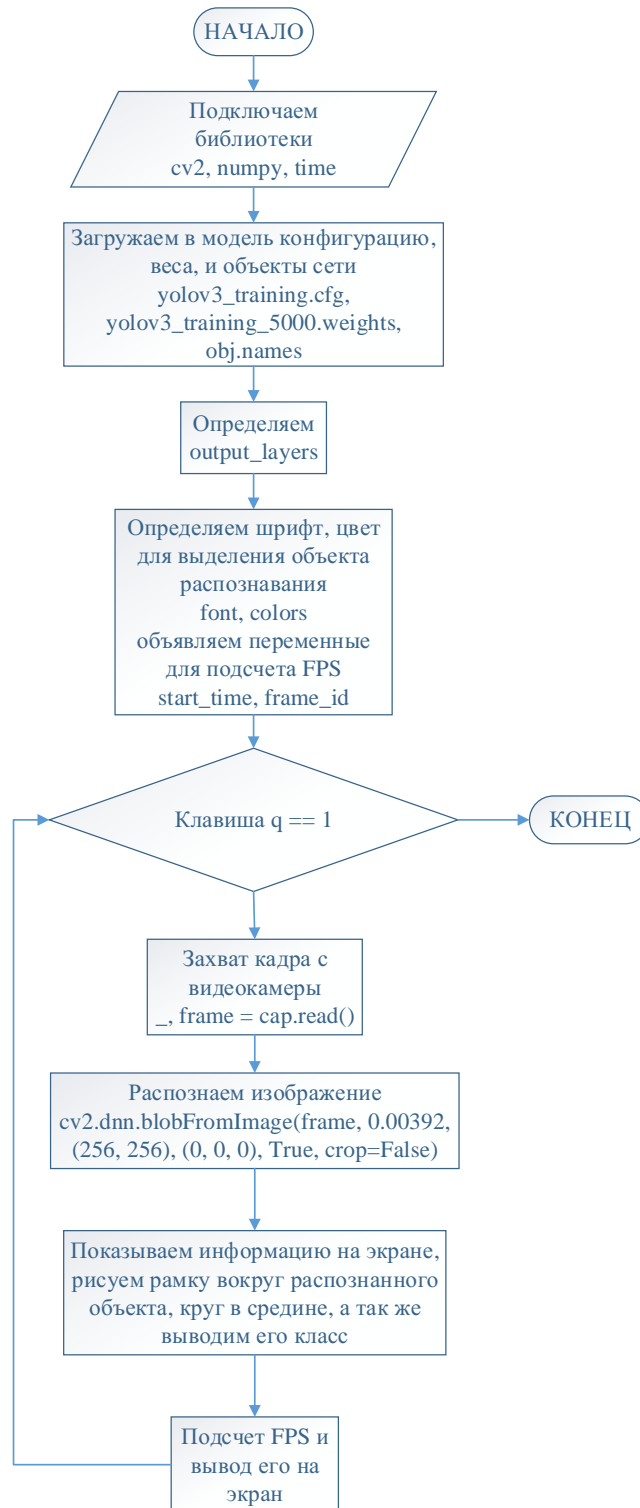


Рисунок 12 – Упрощенная блок-схема работы программы распознавания объектов с видеокамеры

### Общий алгоритм управления конвейером

Как уже было сказано конвейер начинает свою работу с сигнала датчика, который в свою очередь проверяет наличие объекта на ленте. Как только сигнал поступает на микроконтроллер формируются плавно нарастающий сигнал ШИМ с ограничением рывка, задание на 50% от номинальной скорости.

По достижению определенной зоны на которую направлена камера, объект опознается и, если он соответствует одному из классов в базе тогда система формирует определенный сигнал задания, передаваемый на микроконтроллер, который в свою очередь должен сформировать управляющий сигнал на постановку направляющей в определенную позицию.

После того как объект был отсортирован в соответствии своему классу конвейер увеличивает скорость до номинальной. И как только датчик сигнализирующий прохождение объектом конечной точки дает положительный сигнал, микроконтроллер начинает отсчет времени, по истечению которого конвейер прекращает свою работу, отсчет может быть прерван если датчик сигнализирующий подачу нового объекта сработал, и цикл повторяется.

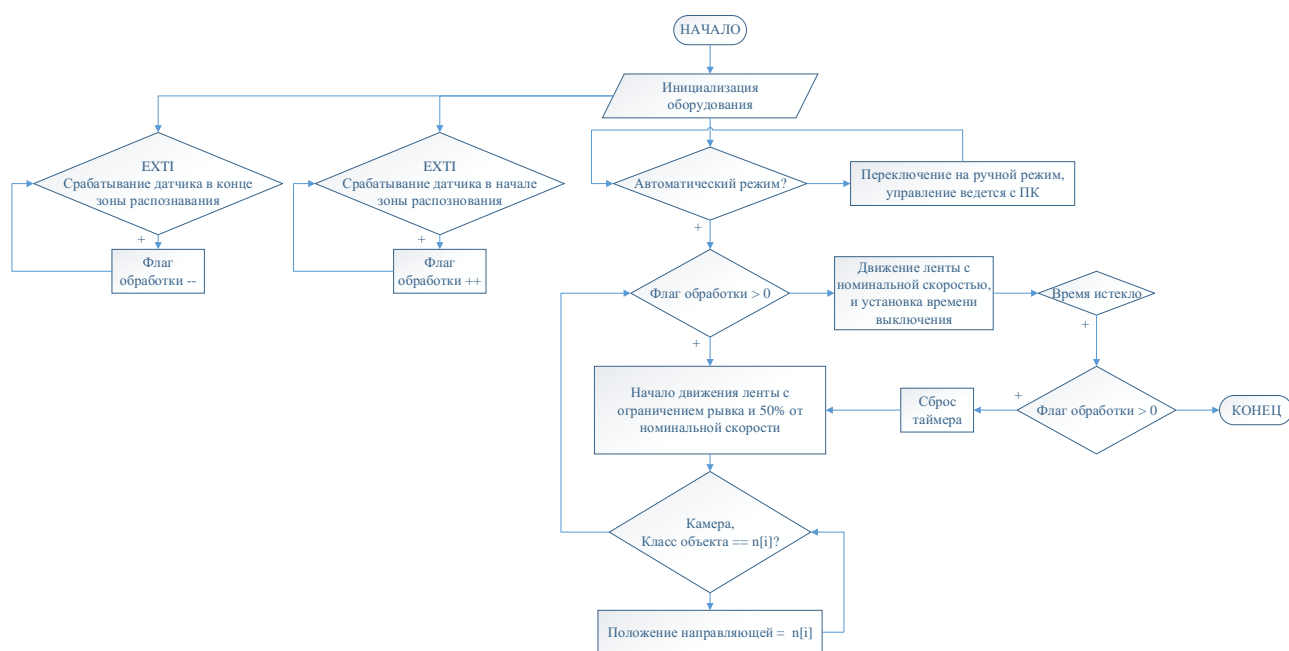


Рисунок 13 – Блок-схема обобщенного алгоритма работы

**Вывод:** Алгоритм распознавания объектов создает достойную конкуренцию системам, использующим только различного рода датчики, но максимальную эффективность можно достичь путем комбинирования таких систем. Таким образом мы минимизируем минусы обеих систем путем увеличения характеристик соответствия. Например, система распознавания не может вычислить массу объекта что компенсируют соответствующие датчики. Или же попросту некоторые функции гораздо легче и надежнее реализовать на основе датчиков, вычисление скорости, позиционирование, точные

размеры и т.д. В свою очередь распознавание даст нам максимально точное внешнее соответствие, цвет, общая форма, текстура, узор и т.д.

### Список литературы

1. Основные типы сортировочных конвейеров В. Голышев [Электронный ресурс]. Режим доступа: <https://sitmag.ru/article/10018-osnovnye-tipy-sortirovochnyh-konveyerov> (дата обращения 13.10.2020)
2. YOLO Real Time Object Detection on CPU [Электронный ресурс]. Режим доступа: <https://pysource.com/2019/07/08/yolo-real-time-detection-on-cpu/> (дата обращения 10.8.2020)
3. YOLO: Real-Time Object Detection [Электронный ресурс]. Режим доступа: <https://pjreddie.com/darknet/yolo/> (дата обращения 1.8.2020)
4. labelImg [Электронный ресурс]. <https://github.com/tzutalin/labelImg> (дата обращения 12.8.2020)
5. RM0090 Reference manual STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs / [www.st.com](http://www.st.com), February 2019. – 1749 с.
6. DocID022152 Datasheet STM32F405xx STM32F407xx ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera / [www.st.com](http://www.st.com), September. – 202 с.