

<https://doi.org/10.51301/ce.2023.i4.04>

## Development of a robotic system for weed control

D.B. Akimov\*, N.A. Bayanbay

Satbayev University, Almaty, Kazakhstan

\*Corresponding author: [akimovdias77@gmail.com](mailto:akimovdias77@gmail.com)

**Abstract.** The new prototype of the farmer's robot, presented in the article, is an innovative solution for the effective removal of weeds from beds in agriculture. Using the plowing method and a computer vision system, the robot automatically detects and removes weeds, minimizing the risk of crop damage. Sensor data analysis and decision-making are carried out by the microcontroller, providing a high level of automation and efficiency. This prototype provides a significant increase in agricultural productivity and environmental sustainability.

**Keywords:** robot, prototype, agricultural crop, weeds, computer vision, sensors, navigation, farmer.

### 1. Введение

Сельское хозяйство является одной из ключевых отраслей в мире, обеспечивая продовольственную безопасность и экономическое развитие многих стран. Однако, проблема сорняков остается одной из главных проблем, с которой сталкиваются сельскохозяйственные производители. Сорные растения конкурируют с культурными растениями за питательные вещества, воду и свет, что снижает урожайность и экономическую эффективность сельскохозяйственного производства. Уже сейчас применяются разного рода действия, по борьбе с ними, разрабатываются роботы, специально ориентированных на уничтожение сорняков. Целью данной статьи как раз является создания прототипа, способного к вспашке этих нежелательных растений. Разберем методы разработки, ключевые моменты, код программы для реализации всего процесса, и компоненты, которые нам понадобятся. Сам процесс создания нужного нам робота нужно разбить на два цельных и важных блока, а именно темы навигации, то есть передвижения робота по грядке, и метода вспашки, непосредственно процесса удаления сорняков, он будет реализован через сервоприводы и компьютерное зрение, основанного на искусственном интеллекте (ИИ) [1, 2].

Робот представляет собой интересный пример инновационного решения в области сельского хозяйства и автоматизации сельскохозяйственных процессов. Его актуальность обусловлена необходимостью повышения эффективности процесса вспашки и борьбы с сорняками в сельском хозяйстве.

Новизна робота заключается в использовании технологии компьютерного зрения для определения выращиваемых культур и автоматического реагирования на их наличие. Это позволяет улучшить точность и эффективность процесса обработки почвы.

Цель данного проекта состоит в разработке и создании автономной системы для обработки грядок сельскохозяйственных культур с минимальным вмешательством человека.

Задачи проекта включают:

1. Разработку и сборку механической и электронной части робота.

2. Программирование системы управления и обработки изображений на основе компьютерного зрения.

3. Тестирование и оптимизацию работы робота.

Основные проблемы, с которыми можно столкнуться, включают в себя точность определения выращиваемых культур с помощью компьютерного зрения, а также надежность и стабильность работы механической части робота при вспашке почвы.

В итоге, робот представляет собой инновационное решение для автоматизации процессов обработки почвы и борьбы с сорняками в сельском хозяйстве, что делает его актуальным и важным для развития сельскохозяйственной отрасли.

### 2. Материалы и методы

Как и живой человек, робот состоит из мышц, заменяемые приводом, мозга, выступающего в роли микроконтроллера, и источника питания, в виде аккумуляторов и батареек. Далее будет перечисление компонентов, нужных для сборки макета:

- Микроконтроллер Arduino Uno;
- Микроконтроллер Raspberry PI Zero W;
- Модуль камеры Raspberry PI OV5647;
- Датчик движения Arduino L298N;
- Датчик линии Arduino KY-033;
- Arduino MICRO SERVO SG90;
- Зарядное устройство для Li-ion аккумуляторов 18650;
- Аккумулятор 18650 X-Balag;
- 9V батарейка;
- Кабель USB – micro-USB;
- microSDHC 16Gb;
- Редуктор с колесами;
- Пара резисторов для понижения напряжения от аккумуляторов.

Возможные дополнения:

- Датчик влажности почвы Arduino FC-28;

© 2023. D.B. Akimov, N.A. Bayanbay

<https://ce.journal.satbayev.university/>. Published by Satbayev University

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

- Датчик температуры и влажности Arduino DHT11;
- Датчик дождя Arduino FC-37;
- Релейный модуль для Arduino;
- LCD дисплей.

Для создания программной части робота-фермера были использованы два микроконтроллера: Arduino для управления движением робота и Raspberry Pi Zero W для реализации компьютерного зрения.

Микроконтроллер Arduino был выбран для управления движением робота благодаря его простоте в использовании и широким возможностям программирования. С его помощью были реализованы алгоритмы управления моторами и датчиками для навигации и уборки сорняков [6].

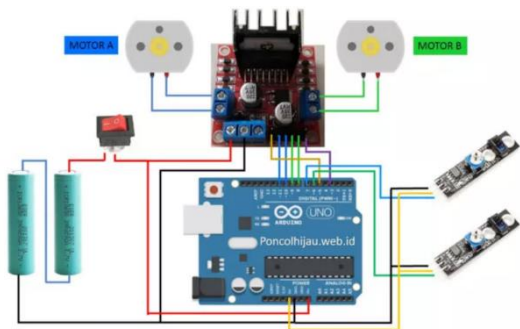


Рисунок 1. Программирование движения робота по линии

Микроконтроллер Raspberry Pi Zero W был использован для обработки изображений и реализации компьютерного зрения [7].

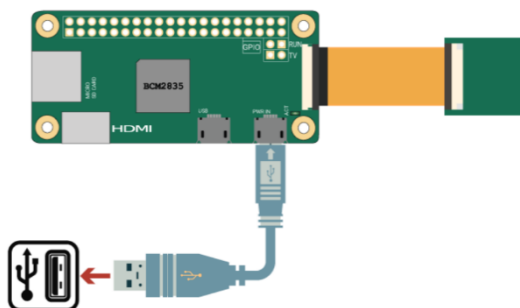


Рисунок 2. Схема подключения Raspberry Pi Zero W к камере и питанию с помощью переходника USB – micro USB

С использованием специализированных библиотек и инструментов компьютерного зрения, таких как OpenCV (Open Source Computer Vision Library), был создан алгоритм обработки изображений. Данный алгоритм включает в себя процессы сегментации и классификации, позволяющие выявить сорняки на изображениях, полученных с камеры, подключенной к микроконтроллеру. После обнаружения сорняков программа принимает решение о дальнейших действиях робота [8].

Принцип работы робота можно довольно легко объяснить следующим образом, робот получает информацию о своем окружении с помощью сенсоров, в данном случае - с помощью системы компьютерного зрения, которая установлена на нем. Эта система позволяет роботу видеть окружающий мир и обнаруживать наличие сорняков и культурных растений на грядке [3].



Рисунок 3. Распознавание сорняков с помощью искусственного интеллекта

Во-вторых, когда камера обнаруживает выращиваемые культуры, она передает полученные данные на микроконтроллер, расположенный внутри робота. Микроконтроллер играет ключевую роль в обработке этих данных и принятии соответствующих решений. Он анализирует информацию, поступающую от компьютерного зрения, и на основе этого анализа принимает решение о дальнейших действиях робота. Например, если робот обнаруживает сорняки, он может начать выполнение программы удаления сорняков. В случае обнаружения культурных растений, робот приостанавливает свои действия, чтобы избежать повреждения урожая [4].

В-третьих, микроконтроллер отправляет команды исполнительным органам робота, которые контролируют движение робота и его манипуляции с грядкой. Например, микроконтроллер может отправить команды на сервоприводы, управляющие движением г-образных конструкций, которые осуществляют срезание сорняков под землей [5].

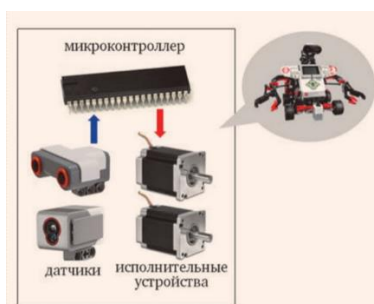


Рисунок 4. Система управления робота

### 3. Результаты и обсуждение

В ходе проделанной работы, мы собрали макет робота, начали с создания корпуса, для этого был приобретен брус 5x3см, а также инструменты для работы с ним. После того как разрезали нужные нам части, все это было склеено, термоклеем, заранее просушили брус, для лучшей клейкости. Корпус готов, можем его обустроить, для этого мы уже ранее приобрели все электронные компоненты. Начинаем с того, что обучаем робота движению по черной линии. Для этого соединяем на макетной плате, и плате Arduino Uno все нужные нам компо-

ненты. Из-за обилия подключенных контактов, у нас образуется мишура в области сборки электронных компонентов, оптимально соединяем контакты, для устранения ненужных помех. Сам робот передвигается на 4-х колесах, но достаточно и 2-ух колес, оснащенных моторным редуктором, данные колеса будут стоять впереди робота, чтобы вести его, а на задние колеса можно установить обычные колёсики, которые бы вращались вслед за роботом.

Передвижение почти готово, можно уже запускать, но нам то нужно, чтобы робот двигался по линии, поэтому обязательно добавляем датчики линии, которые бы корректировали движение нашего робота. Подключили мы их под один из редукторов, так как по центру мы их расположить не могли, ибо там будут разного рода препятствия, а на пути колес, препятствий быть не должно. Работают датчики по типу оптопар, где одна из вершин посылает инфракрасное излучение, а другая вершина принимает. Так как линия у нас была черного цвета, датчик нужно было подстроить под данный цвет, для этого используется переменный резистор, установленный по центру платы датчика, но в ходе опытов, было определено, что он работает только в крайнем положении, и реагирует на черный цвет. Датчики у нас наконец были настроены, и можно приступать непосредственно к написанию кода, особо описывать не буду, в основном вся программа работала на булевых переменных TRUE и FALSE, получаемых от датчиков линий, и в зависимости от сигнала, колеса нашего робота меняли свою скорость подстраиваясь под черную линию.

С этим логика движения робота подходит к концу, и можно плавно приступить к реализации метода вспашки. Для этого у нас уже использовалась другая плата, а именно плата Raspberry, где распиновка не совсем как у Arduino, и будь то цифровые пины, пины питания и заземления, располагались в хаотичном порядке. Разобраться во всем этом нам нужно было для того, чтобы подключить сервоприводы, которые бы уже выполняли свою роль от состояния компьютерного зрения. Опять все подключается, располагается в нужных для эффективной работы местах, и пишется программа. Для реализации компьютерного зрения нам нужна не только плата, но и камера, что передавала бы видеопоток в режиме реального времени на блок обработки информации установленного на плате, по заранее сказанной информации, можно понять, что в роли блока обработки выступает микроконтроллер. Написание кода, для обучения искусственного интеллекта компьютерному зрению является не очень простой задачей. Для начала нужно определиться какую операционную систему мы будем устанавливать на плату, так как первоначально она ее не имеет, загружать ее нужно будет на карту памяти, с помощью картридера, что предоставит возможность к передачи данных на SD-карту. Выбор пал на операционную систему Raspberry Pi OS (32-bit), которая подходит к ранее выбранной плате Raspberry Pi Zero W. Следующим шагом будет подключение непосредственно к операционной системе, имя ее Raspbian, для этого есть два пути подключения, так называемый безголовый режим, и режим подключения с помощью монитора. Легким путем будет подключиться с помощью монитора, для этого нам всего нужно подключить нашу плату к монитору с помощью HDMI-кабеля, подключить остальные компоненты, такие

как клавиатура и мышь, и уже можно приступать к работе с данным микрокомпьютером. Режим безголовы, является альтернативным решением подключения к ОС в случае, если нету возможности подключиться к монитору. Для этого, не вынимая карты памяти с компьютера дополнительно создаем в нем два файла, а именно файл ssh, для того чтобы плата при загрузке понимала, что нужен доступ к ssh, и файл с данными интернет модема, внутри которого будет расписано наименование точки доступа и его пароль, все это нужно чтобы плата подключилась к общему Wi-Fi. После того как добавили нужные файлы, вытаскиваем карты с разъёма, и подключаем к плате. Остается только подать питания и подождать, со временем плата прогрузится, и можно будет приступить к работе с ней.

Далее необходимо подключиться к SSH, для этого открываем командную строку, выполнить это можно нажав клавиши Win+R и введя в поисковую строку слово cmd. В командной строке пишем ssh логин@IP адрес платы Raspberry. Нужно будет ввести пароль, он не будет отображаться, в случае если пароль будет введен правильно, откроется bash консоль Raspberry.

Следующим шагом будет подключение к VNC серверу, для этого пишем в bash консоли, которую ранее открыли команду sudo raspi-config, она предоставит нам доступ к меню настроек Raspberry в котором нас интересует раздел Interfacing Options, внутри которого расположен VNC сервер, для его подключения нужно выбрать режим enable, то есть его включения.

Конечным шагом к доступу операционной системе в режиме без монитора будет установка приложения VNC viewer, где возможно создать подключения к плате, с помощью ее IP адреса. После подключения экран прогрузится, и нас поприветствует графическая среда Raspbian. Наконец операционная система в доступности, и все что остается подключить нужные компоненты и написать код, код пишется в разделе Programming на языке Python. Выбираем удобную среду разработки, и начинаем обучать нейросеть компьютерному зрению. Для этого загружаем вариации изображений интересующего объекта, в целях статьи и разработки — это будет сорняк. Далее пишем программу, импортируя множество библиотек для реализации и обучения машинного зрения. И на выходе получаем систему, способную определять и совершать какие-либо действия при обнаружении изображения заложенного в программу объекта.

Далее важно рассчитать степени свободы исполнительного органа робота и его кинематику, то есть прямую и обратную задачу, также важно вычислить надежность всей системы, то есть ее долговечность.

Степени свободы в области робототехники представляют набор независимых направлений движения и вращения, которые определяют положение и состояние механической системы или объекта. Эти степени свободы включают скорости и их временные производные, предоставляющие полную информацию о состоянии объекта, включая его положение и движение.

Например, свободное твердое тело, или звено, в трехмерном пространстве обладает 6 степенями свободы. Это значит, что оно способно совершать три независимых поступательных движения вдоль взаимно-перпендикулярных осей и три вращательных движения вокруг тех же осей.

В контексте кинематических пар, связанных движений между звеньями, накладываются определенные ограничения, называемые условиями связи. Каждый класс кинематической пары определяется числом этих условий связи, которые варьируются от 1 до 5 в зависимости от способа соединения звеньев. Например, пары первого класса имеют одно условие связи и называются пяти подвижными, в то время как пары пятого класса имеют пять условий связи и называются одноподвижными. Для более лучшего понимания, ознакомьтесь с классификацией кинематических пар можно на рисунке 5.

Схематическое изображение кинематических пар	Условное изображение по ГОСТ ЕСКД	Число связей S	Степень свободы W	Класс кинематической пары	Структурная формула
Шар на плоскости		1	5	1 Пятиподвижная	ВВВ-ПП
Цилиндр на плоскости		2	4	2 Четырехподвижная	ВВ-ПП
Призма на плоскости		3	3	3 Трехподвижная	В-ПП
Сферический шарнир		3	3	3 Трехподвижная	ВВВ
Цилиндрическая пара		4	2	4 Двухподвижная	В-П
Вращательная пара		5	1	5 Одноподвижная	В
Поступательная пара		5	1	5 Одноподвижная	П
Винтовая пара		5	1	5 Одноподвижная	(В) (П)

Рисунок 5. Классификация кинематических пар

Формулы, определяющие число степеней свободы в кинематической цепи, позволяют вычислить количество свободных движений в системе, учитывая все звенья и связи между ними. Например, для пространственной кинематической цепи общего вида используется структурная формула так же именуемая как формула Сомова-Малышева, которая учитывает количество звеньев и классы кинематических пар в цепи. Формула для плоской кинематической цепи, или формула Чебышева, аналогично применяется для плоских систем, учитывая количество звеньев и пар в цепи. Если на движение звена в пространстве, не наложено никаких условий связи, то оно обладает 6 степенями свободы.

Применяемая формула для плоских механизмов (Формула Чебышева):

$$W=3n-2P_5-P_4 \tag{1}$$

Для пространственных механизмов (Формула Сомова-Малышева):

$$W=6n-5P_5-4P_4-3P_3-2P_2-P_1 \tag{2}$$

где W – количество степеней свободы;  
 n – число подвижных звеньев;  
 P<sub>5</sub>, P<sub>4</sub>, ... P<sub>1</sub> – кинематические пары с одной и с двумя наложенными связями.

Рассчитаем степень свободы исполнительного органа, для этого определим класс кинематических пар, и количество подвижных звеньев. На рисунке 6 изображена кинематическая схема для нахождения всех нужных нам элементов.

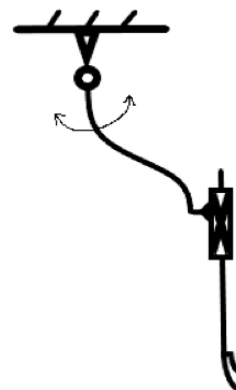


Рисунок 5. Кинематическая схема исполнительного органа

Отсюда видим, что имеем в общей сложности 2 кинематические схемы, а именно одна вращательная расположенная у основания стойки, и поступательная, движущее клинок по вертикали. Из подвижных звеньев подвижны все, кроме конечно стойки, которая изначально статична, таким образом количество подвижных звеньев будет равняться двум. Уже отсюда можно начать рассчитывать степени свободы, так как механизм движется по всем осям координат, делаем вывод о том, что он является пространственным, а значит имеет место быть использовать формулу Сомова-Малышева по формуле (2) для расчетов степеней свободы пространственных механизмов.

$$W=6*2-5*2=2 \tag{3}$$

Подставляя значения подвижных звеньев и пар пятого класса, определяем, что рабочий орган имеет 2 степени подвижности, так же именуемые как степени свободы. Так как кроме пар пятого класса ничего не имели, в расчет иные кинематические пары не берем. На этом определение степеней свободы можно завершить, и перейти к следующим расчетам.

Задача кинематики в робототехнике является одной из ключевых областей исследований, которая фокусируется на прямой и обратной кинематике. Прямая кинематика решает задачу определения положения и ориентации рабочего органа робота на основе известных параметров его звеньев. Этот аспект играет критическую роль в управлении движением робота, предоставляя информацию о его конфигурации в пространстве.

С другой стороны, обратная кинематика решает обратную задачу, определяя значения угловых параметров звеньев для достижения нужного положения и ориентации рабочего органа. Этот аспект широко применяется в программировании движений роботов для выполнения различных задач в разнообразных пространственных средах.

Оба этих аспекта кинематики имеют значительное практическое применение в промышленности, медицине, автоматизации и исследованиях. Понимание прямой и обратной кинематики является ключевым для разработки



и управления современными роботизированными системами, обеспечивая их эффективное функционирование в разнообразных условиях и задачах. Рассчитаем кинематику для исполнительного органа нашего робота на рисунке 6.

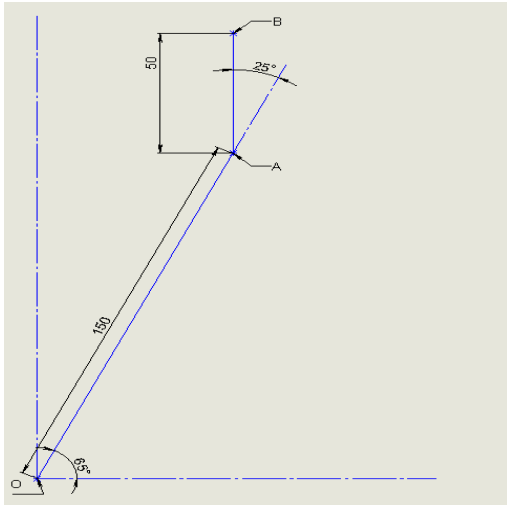


Рисунок 6. Чертеж ИО для расчета кинематики

Из рисунка 6 видим, что чертеж состоит из двух прямых, повернутых на определенный угол. Первая линия повернута относительно оси x, на 65° градусов, вторая же, на 25°, относительно первой прямой. Линии обозначим как L1, L2, углы же как Q1, Q2. Начнем расчёты с прямой задачи кинематики, для этого от начала координат (O) перейдем к точке A, рассчитаем XA, YA по формулам (4) и (7). Для удобства расчетов, преобразим мм в см.

$$XA=L1*\cos(Q1) \tag{4}$$

$$XA=15*\cos(65^\circ) \tag{5}$$

$$XA=15*0.422=6.33 \tag{6}$$

$$YA=L1*\sin(Q1) \tag{7}$$

$$YA=15*\sin(65^\circ) \tag{8}$$

$$YA=15*0.906=13.59 \tag{9}$$

Координаты первой точки, мы нашли, теперь нужно рассчитать положение второго звена, для этого воспользуемся формулами (10) и (13). Второе звено повернуто относительно плеча на угол Q1+Q2, это важно учесть при следующих расчетах.

$$X'=L2*\cos(Q1+Q2) \tag{10}$$

$$X'=5*\cos(90^\circ) \tag{11}$$

$$X'=5*0=0 \tag{12}$$

$$Y=L2*\sin(Q1+Q2) \tag{13}$$

$$Y'=5*\sin(90^\circ) \tag{14}$$

$$Y'=5*1=5 \tag{15}$$

После того, как мы определили положение двух звеньев, нужно найти конечную точку, это точка B, вычислим ее по формулам (16) и (17), координаты ее будут равны сложению прежних вычислений.

$$x=XA+X'=6.339+0=6.33 \tag{16}$$

$$y=YA+Y'=13.59+5=18.59 \tag{17}$$

Наконец мы высчитали прямую задачу кинематики нашего ИО, конечное положение механизма будет на координатах (6.33; 18.59). На рисунке 7 можно увидеть, что наши расчеты были верны.

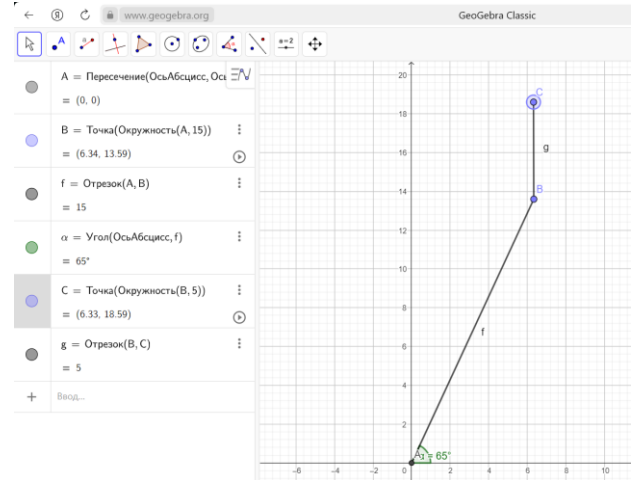


Рисунок 7. Определение конечной точки с помощью приложения GeoGebra

*Обратная задача кинематики*

У нас уже есть уравнение координат прямых, уже на основе их можем найти углы. Для этого преобразуем уравнения через обратную функцию тригонометрии, для угла Q1, Q2.

$$XA=L1*\cos(Q1) \tag{18}$$

$$\cos(Q1)=XA/L1 \tag{19}$$

$$Q1=\arccos(XA/L1) \tag{20}$$

$$Q1=65 \tag{21}$$

$$YA=L1*\sin(Q1) \tag{22}$$

$$\sin(Q1)=YA/L1 \tag{23}$$

$$Q1=\arcsin(YA/L1) \tag{24}$$

$$Q1=65^\circ \tag{25}$$

Для второго угла, выраженного относительно L1:

$$X'=L2*\cos(Q1+Q2) \tag{26}$$

$$\cos(Q1+Q2)=X'/L2' \tag{27}$$

$$Q1+Q2=\arccos(X'/L2) \tag{28}$$

$$Q2=\arccos(X'/L2)-Q1 \tag{29}$$

$$Q2=25^\circ \tag{30}$$

$$Y'=L2*\sin(Q1+Q2) \tag{31}$$

$$\sin(Q1+Q2)=Y'/L2' \tag{32}$$

$$Q1+Q2=\arcsin(Y'/L2) \tag{33}$$

$$Q2=\arcsin(Y'/L2)-Q1 \tag{34}$$

$$Q2=25^\circ \tag{35}$$

Подставляя значения в итоге находим значения углов, а это значит, что обратная задача кинематики тоже решена.

Для расчета надежности всей роботизированной системы, нужно определить либо интенсивность отказов

отдельных компонентов, либо же их среднее время наработки до отказа далее СВНО. Воспользуемся вторым случаем, и получим средние значения работы каждого компонента, которые в целом имеют значение в работе всей системы, ниже приведен список компонентов, с их СВНО:

1. Arduino Uno: MTTF = 62500 часов;
2. Сервопривод: MTTF = 52000 часов;
3. Датчик линии: MTTF = 40000 часов;
4. Raspberry Pi: MTTF = 57500 часов;
5. Камера: MTTF = 60000 часов;
6. Драйвер для двигателя: MTTF = 55000 часов;
7. Мотор: MTTF = 50000.

Так как мы определили СВНО, можем начать расчет интенсивности отказов, он рассчитывается по формуле

$$\lambda_i = 1/T_i \quad (36)$$

где  $\lambda_i$  – это интенсивность отказа компонента;  
 $T_i$  – СВНО отдельного компонента.

$$\lambda_{ard} = 1/T_{ard} = 1/62500 = 0.0001656 \quad (37)$$

$$\lambda_{servo} = 1/T_{servo} = 1/52000 = 0.0001923 \quad (38)$$

$$\lambda_{line} = 1/T_{line} = 1/40000 = 0.000025 \quad (39)$$

$$\lambda_{rasp} = 1/T_{rasp} = 1/57500 = 0.0001739 \quad (40)$$

$$\lambda_{cam} = 1/T_{cam} = 1/60000 = 0.0001667 \quad (41)$$

$$\lambda_{drive} = 1/T_{drive} = 1/55000 = 0.0001818 \quad (42)$$

$$\lambda_{motor} = 1/T_{motor} = 1/50000 = 0.0002 \quad (43)$$

После того как нашли интенсивность отказа компонентов, нужно определить СВНО уже всей системы, для этого воспользуемся следующей формулой:

$$T_{total} = 1/\lambda_{sum} \quad (44)$$

где  $T_{total}$  – это СВНО всей системы;

$\lambda_{sum}$  – сумма значений интенсивности отказов, она рассчитывается по формуле:

$$\lambda_{sum} = \sum \lambda_i \quad (45)$$

где  $\sum \lambda_i$  – сумма интенсивности отказов.

Теперь можем определить сумму значений интенсивности отказов, и высчитаем СВНО всей системы:

$$\lambda_{sum} = 0.000016 + 0.000019 + 0.000025 + 0.000014 + 0.000017 + 0.000018 + 0.00002 = 0.000132 \quad (46)$$

$$T_{total} = 1/0.000132 = 7575 \text{ часов} \quad (47)$$

СВНО системы в итоге составило 7575 часов, теперь, когда известны средние значения СВНО и суммы интенсивности отказов, можно рассчитать вероятность безотказной работы и вероятность отказа системы. Для этого понадобятся следующие формулы:

$$P(t) = e^{-\lambda t} \quad (48)$$

$$Q(t) = 1 - P(t) \quad (49)$$

где  $P(t)$  – вероятность безотказной работы;

$e$  – экспонента;

$\lambda$  –  $\lambda_{sum}$ ;

$t$  – произвольный интервал времени, возьмем  $t=3000$ ;

$Q(t)$  – вероятность отказа.

Рассчитаем вероятность безотказной работы:

$$P(t) = e^{-0.000132 \times 3000} = 0.67 \quad (50)$$

Отсюда видим, что система с вероятностью 0.67 или 67% будет работать бесперебойно при трех тысячах часов эксплуатации, теперь же рассчитаем вероятность отказа, которая рассчитывается зависимостью безотказной работы системы:

$$Q(t) = 1 - 0.67 = 0.33 \quad (51)$$

Вероятность отказа путем несложных математических вычислений оказалась равна 0.33 или 33% процентам отказа.

Определим теперь время работы в целом устройства, надежности его по времени, так как он должен избавляться от сорняков, и помогать в выращивании агрокультуры, стоит учесть сезона взращивания культур, обычно он приход на летний сезон, который в свою очередь равняется 3 месяцам, это уже 90 дней. В сутки робот будет работать до 12 часов, то есть с раннего утра, и до позднего вечера, это равняется половине суток, и, если рассчитать в сумме сколько он дней в году проработает, это у будет равняется полтора месяцам, или же 1080 часам работы. Теперь можно рассчитать сколько лет проработает робот обозначим как  $T_{srok}$  по формуле (52). Для этого поделим СВНО всего устройства на количество часов работы в год, из этого выйдет что:

$$T_{srok} = T_{total}/1080 = 7575/1080 = 7.01 \text{ лет} \quad (52)$$

В итоге робот проработает 7.01 лет, или же 2561 дней.

В будущем при определенном достатке финансов или спонсирования можно модернизировать наш проект, заменить либо же найти альтернативу более лучшими деталями, продумать иные процессы, по типу анализа влажности почвы, температуры окружающей среды. Создать системы покрова от дождя, к примеру, в случае осадков, чтобы не происходило каких-либо сбоев электроники у нашего робота, установить датчик дождя, к которому по цифровой связи был связан сервопривод, покрывающий нашего робота пленкой водостойкой, и в процессе окончания осадков пленка бы возвращалась на место. Не менее важной более совершенной системы навигации, основанной на GPS-маркерах, отображаемых на само прописанной карте, либо же при взаимодействии таких известных карт как Google, Яндекс. Там бы непосредственно выделялось нужная область, в рамках которой бы наш робот совершал требуемые от него действия и совершал более совершенное движение. Конечно же не стоит забывать про искусственный интеллект(ИИ) играющий особо важную роль как в повседневной, так и в промышленных областях нашей жизни, сама перспектива развития ИИ прогнозируется уже на многие годы в перед, и не будет преувеличением сказать, что за ней стоит будущее инженерной отрасли, интересное мне как лицу связанного с технологической специальностью, и скорость развития или же прогресса человечества в общем.

#### 4. Выводы

Подведя итоги нашей работы, мы успешно достигли поставленных целей и создали прототип робота-фермера, способного эффективно бороться с сорняками на полях. Основой для нашего прототипа была методика борьбы с сорняками, интегрированная с использованием датчиков линии. Эти

датчики позволили корректировать движения робота, обеспечивая его равномерное продвижение по полю.

Кроме того, мы внедрили компьютерное зрение, которое обеспечивает контроль за сохранностью выращиваемой культуры. Это важное дополнение к функциональности робота, позволяющее избежать повреждений растений в процессе борьбы с сорняками.

В процессе создания прототипа мы учли основные аспекты его конструкции и функциональности. Мы собрали и оптимизировали корпус робота, интегрировали необходимые электронные компоненты и написали соответствующий программный код для управления движением и обнаружения сорняков.

В результате наших усилий нам удалось создать робота, который станет надежным помощником в сельском хозяйстве. Он способен эффективно увеличить производство растений и овощей, а также сэкономить средства, которые обычно тратятся на ручную прополку и борьбу с сорняками.

## References / Литература

- [1] Article. Robot-fermer Solix samostojatel'no najdet i unichtozhit sornjaki. Retrieved from: <https://www.techcult.ru/robots/11230-umnyj-robot-fermer>
- [2] Article. V Latvii sozdali robota, sposobnogo raspoznavat' i unichtozhat' lazerom sornjaki. Retrieved from: <https://naukatv.ru/news/v-latvii-sozdali-robot-a-s-ii-otlichayusc-hego-sornjaki-i-unichtozhayushego-ikh-lazerom>
- [3] Бойко, А. Propolka i roboty - Katalog sel'skhozjajstvennyh robotov. Retrieved from: <https://robotrends.ru/robotopedia/propolki-robotizaciya>
- [4] Article. Chto takoe Sistemy upravlenija robotami? Retrieved from: [https://www.confires.com/blog/fire-sprinklers-damage-build-https://dzen.ru/a/ZCbxIEVpmFdhejhL#:~:text=Системы%20Управления%20роботами%20\(Robot%20Control,параметры%20для%20выполнения%20определенных%20задач](https://www.confires.com/blog/fire-sprinklers-damage-build-https://dzen.ru/a/ZCbxIEVpmFdhejhL#:~:text=Системы%20Управления%20роботами%20(Robot%20Control,параметры%20для%20выполнения%20определенных%20задач)
- [5] Article. Sel'skhozjajstvennye roboty v bor'be s sornjakami. Retrieved from: <https://rep.bsatu.by/bitstream/doc/14889/1/Andrushkevich-N-A-Selskhozjajstvennye-roboty-v-borbe-s-sornjakami.pdf>
- [6] Evsegneevev, O. Arduino: drajver L298N dlja motora postojannogo toka. Retrieved from: <https://robotclass.ru/tutorials/arduino-l298-dc-motor-driver/>
- [7] Article. Mashinnoe zrenie na Raspberry Pi opredelenie koordinat ob'ektov vybrannogo cveta v kadre. Retrieved from: <https://lesson.iarduino.ru/page/machine-vision-raspberry/>
- [8] Article. Videonabljudenie i iskusstvennyj intellekt: besprecedentnye vozmozhnosti dlja jeffektivnosti. Retrieved from: <https://i-smotr.ru/videonabljudenie-i-iskusstvennyj-intellekt-besprecedentnye-vozmozhnosti-dlya-effektivnosti.html>

## Арамшөптермен күресуге арналған роботтық жүйені дамыту

Д.Б. Акимов\*, Н.А. Баянбай

Satbayev University, Алматы, Қазақстан

\*Корреспонденция үшін автор: [akimovdias77@gmail.com](mailto:akimovdias77@gmail.com)

**Аңдатпа.** Мақалада келтірілген фермер роботының жаңа прототипі ауыл шаруашылығындағы жүйектерден арамшөптерді тиімді жоюдың инновациялық шешімі болып табылады. Жер жырту әдісі мен компьютерлік көру жүйесін қолдана отырып, робот арамшөптерді автоматты түрде анықтайды және жояды, бұл егіннің зақымдану қаупін азайтады. Сенсорлық деректерді талдау және шешім қабылдау, автоматтандыру мен жұмыс тиімділігінің жоғары деңгейін қамтамасыз ететін микроконтроллер арқылы жүзеге асырылады. Бұл прототип ауыл шаруашылығының өнімділігі мен экологиялық тұрақтылығының айтарлықтай өсуін қамтамасыз етеді.

**Негізгі сөздер:** робот, прототип, дақыл, арамшөптер, компьютерлік көру, сенсорлар, навигация, фермер.

## Разработка роботизированной системы для борьбы с сорняками

Д.Б. Акимов\*, Н.А. Баянбай

Satbayev University, Алматы, Казахстан

\*Автор для корреспонденции: [akimovdias77@gmail.com](mailto:akimovdias77@gmail.com)

**Аннотация.** Новый прототип робота фермера, представленный в статье, представляет собой инновационное решение для эффективного удаления сорняков с грядок в сельском хозяйстве. Используя метод вспашки и систему компьютерного зрения, робот автоматически обнаруживает и удаляет сорняки, минимизируя риск повреждения урожая. Анализ данных сенсоров и принятие решений осуществляются микроконтроллером, обеспечивая высокий уровень автоматизации и эффективности работы. Этот прототип обеспечивает значительный рост производительности и экологической устойчивости сельского хозяйства.

**Ключевые слова:** робот, прототип, сельскохозяйственная культура, сорняки, компьютерное зрение, сенсоры, навигация, фермер.

Received: 09 September 2023

Accepted: 15 December 2023

Available online: 31 December 2023