

DEDAN KIMATHI UNIVERSITY OF TECHNOLOGY (DKUT)

**SCHOOL OF ENGINEERING MECHATRONIC ENGINEERING DEPARTMENT**

**STUDENTS’ EXTERNAL ATTACHMENT LOG-BOOK**

# DURATION: 12 WEEKS

**THE LOG BOOK**

1. INTRODUCTION

This book is to assist the student to keep record of the training. It will show the departments and sections in which the student has worked and the periods of time spend in each.

1. DAILY REPORT

The daily work carried out during the periods of training is to be recorded clearly with sketches and diagrams where applicable.

1. WEEKLY REPORT

This is a summary of work done in a week and should cover theory/practical report on the work covered. Students are required to present the log-book weekly to the supervisor for assessment of content and progress. The Supervisor can use any page for his comments where necessary.

1. CHANGE OF ATTACHMENT

A student is expected to start and finish his external attachment in the university. If it becomes absolutely necessary that he must change his place of attachment, the student should first secure permission in writing from the college. His application for change of place of attachment should indicate the name and the address (not just Post Office Box) of the company or industry to which he wishes to transfer. Any attachment not properly authorized will be cancelled.

1. COLLEGE SUPERVISOR’S VISIT

The training supervisor of the DKUT will check the log-book. When he/she assesses the project to ensure that the proper training is being received, and record his/her comment on the paper provided for this purpose, towards the end of the book.

1. SPECIAL REQUEST FOR THE UNIVERSITY SUPERVISOR Please assess the student as per assessment form provided.
2. REPORT WRITING

In addition to the daily and weekly record the student should submit a summary report of the work done during the attachment duration e.g. full coverage of the course, problems encountered. Suggest improvements to make the programme worthwhile. The report should contain a summary of activities of the organization, manufacturing services/processes the student was involved in. This includes the highlights of the project the student is involved in. The student is expected to point out the weak and strong points of the attachment.

1. REPORT SUBMISSION

The log-book and report must be submitted to the relevant departmental Attachment Coordinator at the end of the attachment.

S T U D E N T ‘S P A R T I C U L A R S

**MUIGAI PETER NDIBA**

Name of student: ……………………………………………………………………

(Surname first)

**E022-01-2454/2020**

Registration No. of the student: ……………………………………………………..

Department: …………………………………………………………………………..

**MECHATRONICS ENGINEERING**

Course of study: …………………………………………………………………………

**BSC.MECHATRONICS ENGINEERING**

Year of course: ………………………………………………………………………….

**3rd YEAR**

Name of the University Supervisor: ……………………………………………………….

**DR. MORRIS KIMATHI**

**22.05.2023**

Duration: From: ……………………………………………………………………………

To: ………………………………………………………………………………..

**11.08.2023**

**ATTACHMENT TIME-PLAN**

(The student should draw a time-table indicating time to be sent on each task/section)

**WEEKLY PROGRESS CHART (WEEK ENDING: ------ ONE )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Discussion on the project expectation with the supervisor. The project was to be a line tracking vehicle that can be used to pick objects placed in a specified game field. There were rules and regulations of the project which were discussed. |
| TUES | We discussed the different designs of line tracking vehicles that already exist and picked the appropriate design and started designing it on solidworks. |
| WED | Continued on the design of the vehicle this included the completion of the chassis of the AGV and the robotic arm also. Showed the supervisor the design so he would acquire the required equipment. |
| THUR | While waiting on the equipment, we started laying the canvas on the floor while following the design given on the rules provided. |
| FRI | We used the masking tape to represent the lines that the AGV is supposed to follow and a black one to permanently attach the canvas on the floor. |
| SAT |  |

TRAINEES WEEKLY REPORT

**PROJECT OBJECTIVES**

1. The Game Robot must move from the start zone (Start) to the picking zone (Engine).

2. The Game Robot must pick an object (engine) and move with it from the picking zone to the assembling zone (chassis).

3. The Game Robot must correctly place the object (engine) in the groove (carving) of the corresponding shape on the assembling point.

4. The Game Robot must move towards the wheel rack.

5. The Game Robot must successfully pick the coloured wheel.

6. Repeat step number 5, for the other three (3) coloured wheels.

7. A robot that successfully moves towards the assembling point after successfully picking the coloured wheel

8. Repeat step number 7, for the other three (3) coloured wheels

9. A robot that successfully fits the coloured wheel on their corresponding part of the shaft on the assembling point on the chassis

10. Repeat step number 9, for the other three (3) coloured wheels

11. A robot that moves towards the ramp

12. A robot that successfully climbs the ramp 5

13. A robot that successfully picks the cabin on the ramp

14. A robot that successfully descends the ramp with the cabin

15. A robot that correctly moves towards the assembling point with the cabin

16. A robot that successfully fits the cabin on the assembling point

17. A robot that moves towards the trailer

18. A robot that successfully picks the trailer

19. A robot that successfully moves towards the assembling point with the trailer

20.A robot that successfully fits the trailer on the assembling point

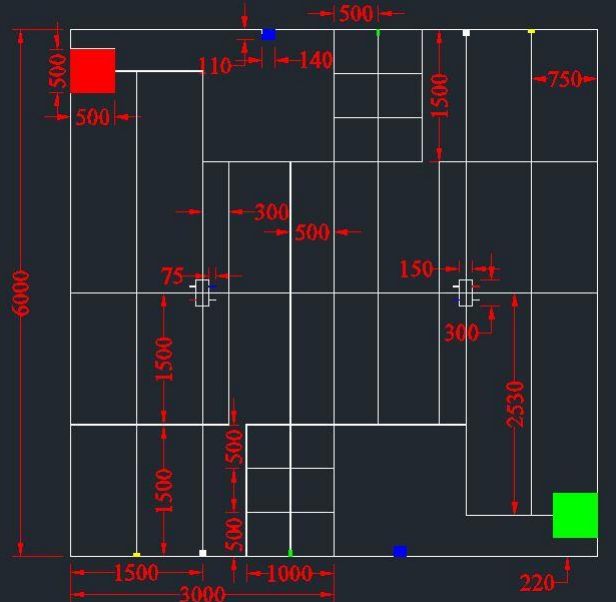
21. A robot that parks successfully at the starting zone.

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

27/05/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.



Student’s Signature: …………………………… Date…………………………………………..

27/05/2023

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

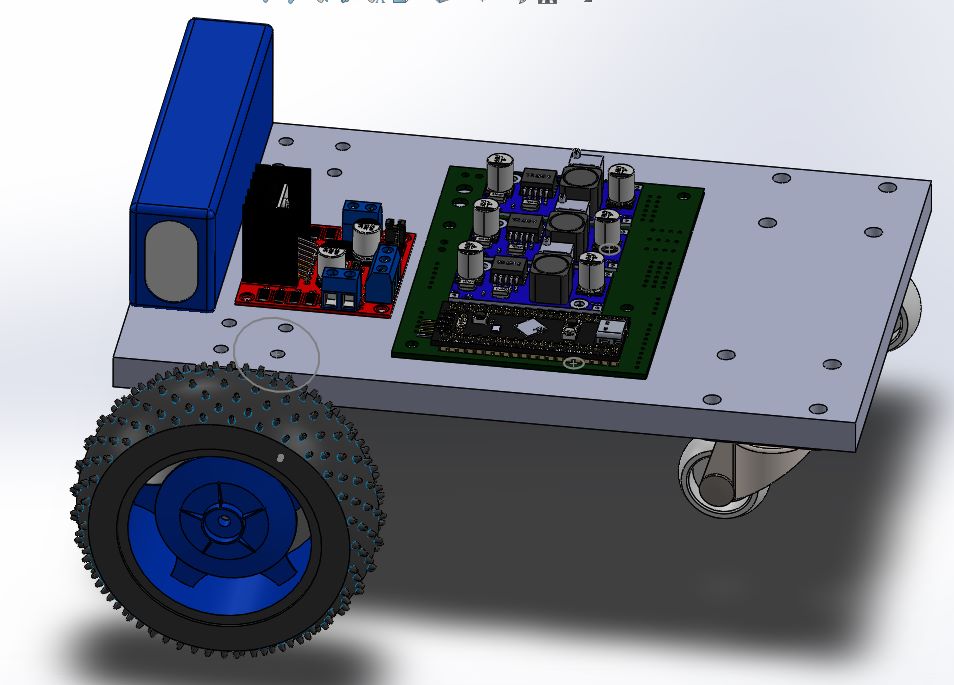
……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………..

Signature: …………………………………….. Date: ………………………………………..…

**WEEKLY PROGRESS CHART (WEEK ENDING: -----TWO )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | The materials arrived during the weekend so we started with cutting the acrylic sheet according to our design using a hacksaw. We also drilled the acrylic to get points to mount our pcb and the battery. |
| TUES | We started assembling the PCB having designed it some weeks before and had it printed. We assembled the needed parts such as the header pins, XT60 connector and the buck-converters. Adjusted the buck-converters output to match the needed voltage. |
| WED | Mounted the assembled PCB, the motors, motor driver, battery and castor wheels on the chassis. We balanced the whole system to make sure all wheels would be touching the ground on a flat surface |
| THUR | PUBLIC HOLIDAY |
| FRI | We zeroed the servo motors using the STM32F401CCU6 micro-controller and started the assembly of the 4-D.O.F robotic arm(this involved servo motors and long U-aluminium brackets) that we were to use to pick the different parts on the game field. |
| SAT |  |

## TRAINEES WEEKLY REPORT

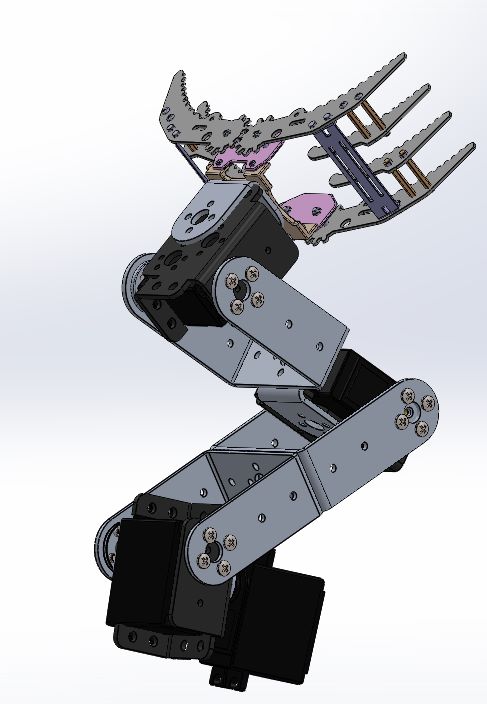


NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

03/06/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.



Student’s Signature: …………………………… Date…………………………………………..

03/06/2023

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………..

Signature: …………………………………….. Date: ………………………………………..…

**WEEKLY PROGRESS CHART (WEEK ENDING: ---- THREE )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Started by mounting the robotic arm on the chassis of the autonomous grounded vehicle. Also tested if the arm worked when combined with the AGV that is the voltage that we had set on the buck converters. |
| TUES | We started moving the motors using the L298N motor driver by writing the code for the STM32 micro-controller using the STM-CUBE-IDE and the HAL library. We read the ADC values of the IR Array which displayed them on our phones using the HC-05 Bluetooth module. |
| WED | We researched on PID control and wrote a code on it, the code also included UART communication that we could use to debug the system using the Bluetooth module (TX of the Bluetooth was to be connected to the RX of the PCB and vice-versa) |
| THUR | Finished the code on PID and started testing where i was to reduce the oscillations that would occur due to the control method we used (PID tuning). We tested the whole system so as to achieve line tracking on a straight line. |
| FRI | We needed the system to determine junction, we did a research on how to achieve that and started working on the code for that. We also wanted to about turn when at some points in the game field so we also worked on it. |
| SAT | Did research on kinematics i.e forward kinematics so we can move the robotic arm to pick the different objects. |

## TRAINEES WEEKLY REPORT

PID control is a widely used control theory that provides a feedback mechanism for controlling systems to achieve desired behavior. The acronym PID stands for Proportional, Integral, and Derivative, which represent the three components of the control algorithm.

1. Proportional (P) Control:

The proportional component of the PID control algorithm produces an output that is proportional to the current error between the desired set-point and the actual value of the system being controlled. It adjusts the control action based on the magnitude of the error. The proportional gain (Kp) determines the strength of the proportional response. A higher value of Kp results in a more aggressive response, but it can also lead to overshoot and oscillations.

1. Integral (I) Control:

The integral component of the PID control algorithm takes into account the accumulation of past errors over time. It integrates the error signal and adds it to the control action. The integral term helps to eliminate steady-state errors that may occur due to external disturbances or system biases. It continuously adjusts the control action until the error is minimized. The integral gain (Ki) determines the response of the integral component. A higher Ki value increases the response speed, but it can also introduce instability if set too high.

1. Derivative (D) Control:

The derivative component of the PID control algorithm considers the rate of change of the error. It helps to anticipate the future trend of the error and provides a damping effect to the control action. The derivative term helps in reducing overshoot and improving the stability of the system. The derivative gain (Kd) determines the strength of the derivative response. A higher Kd value increases the rate of change of the control action, but excessive derivative action can lead to amplification of noise and instability.

The PID control algorithm combines these three components by calculating the control action as follows: Control Action = Kp \* (Proportional term) + Ki \* (Integral term) + Kd \* (Derivative term)

The PID controller continuously computes and adjusts the control action based on the feedback from the system being controlled. By tuning the gains Kp, Ki, and Kd, the controller can be optimized to achieve the desired response, such as stability, speed of response, and minimizing overshoot.

The tuning of PID parameters often involves a trial-and-error process or the use of more advanced tuning methods such as Ziegler-Nichols or optimization algorithms. The choice of tuning method depends on the characteristics of the system and the desired performance criteria.

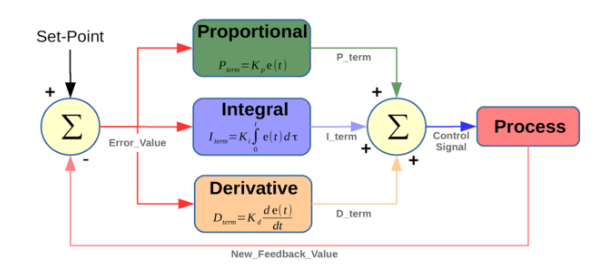
Overall, PID control theory provides a flexible and effective approach for controlling a wide range of systems, from simple to complex, and it is widely utilized in various fields including robotics, process control, automotive control systems, and more.

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

10/06/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.



Student’s Signature: …………………………… Date…………………………………………..

10/06/2023

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………..

Signature: …………………………………….. Date: ………………………………………..…

**WEEKLY PROGRESS CHART (WEEK ENDING: ----FOUR )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | We wrote the code on forward kinematics that could be used to pick the engine and go to its home position after pickup. Tested it and improved it for much more accuracy. We also wrote code to drop the engine on the chasis on the game field |
| TUES | On the game field there was a wheel rack the contained 4 wheels that were to be picked up and placed on the chasis that was on the game field. We wrote the code for picking each wheel and placing it on a container that was to be used to store all the wheels. |
| WED | We combined the line tracking code and the forward kinematics code. Tested the whole code on the robot and adjusted to achieve the needed objectives. |
| THUR | We tested the robot and adjusted it to the best we could. This included the distance to stop so as to pick the objects. |
| FRI | Presented the project to the supervisor. |
| SAT |  |

## TRAINEES WEEKLY REPORT

## Forward kinematics is a fundamental concept in robotics and kinematics that deals with determining the position and orientation of an end effector (e.g., robot's hand or tool) based on the configuration of the robot's joints. It involves calculating the transformation from the robot's base frame to the end effector frame using the known geometric parameters and joint variables.

## The forward kinematics problem is commonly solved using the Denavit-Hartenberg (DH) parameters and homogeneous transformation matrices. The DH parameters provide a systematic way to assign coordinate frames to each joint of the robot. Each joint is described by four parameters: link length (a), link twist (α), link offset (d), and joint angle (θ). These parameters define the geometric relationship between adjacent joints.

## The transformation matrix (T) represents the homogeneous transformation from one coordinate frame to another. For each joint, a transformation matrix is computed based on its DH parameters. The transformation matrices are multiplied successively to obtain the overall transformation from the robot's base frame to the end effector frame.

## The general form of a transformation matrix is:

## T = Rot(Z, θ) \* Trans(Z, d) \* Trans(X, a) \* Rot(X, α)

## where Rot(axis, angle) represents a rotation matrix around the specified axis by the given angle, and Trans(axis, distance) represents a translation matrix along the specified axis by the given distance.

## By multiplying the transformation matrices for each joint, the forward kinematics equation can be established as:

## T\_total = T\_1 \* T\_2 \* ... \* T\_n

## where T\_total represents the overall transformation matrix from the base frame to the end effector frame, and T\_1, T\_2, ..., T\_n represent the transformation matrices for each joint.

## The position and orientation of the end effector can be extracted from the transformation matrix T\_total. The translation components of T\_total provide the coordinates of the end effector in the base frame, and the rotation matrix within T\_total represents the orientation of the end effector.

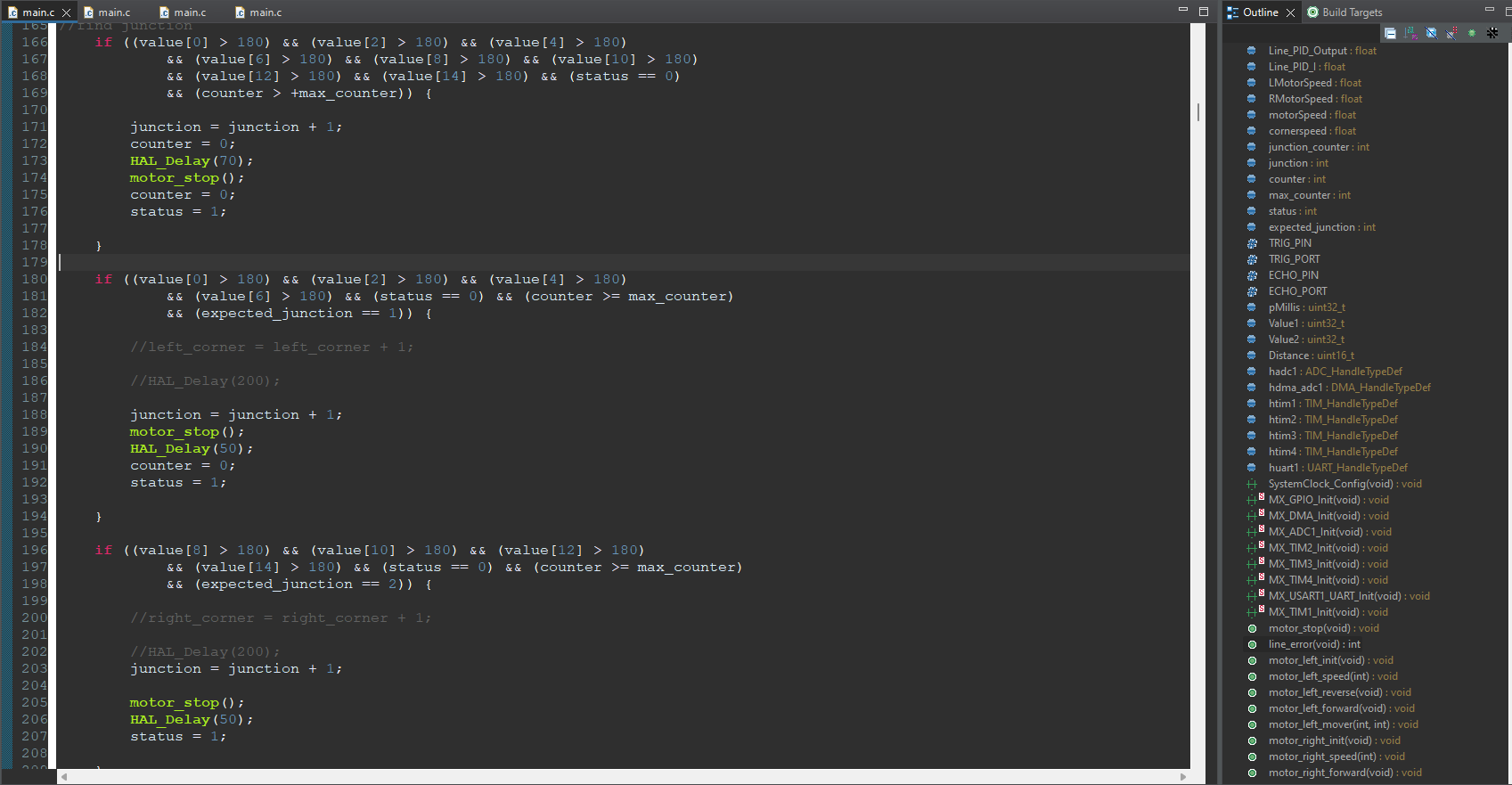
## It's important to note that forward kinematics assumes idealized robot joints without any constraints or external forces. In reality, factors such as joint limits, joint flexibility, and external disturbances can affect the actual position and orientation of the end effector. Therefore, forward kinematics serves as a fundamental tool for understanding and modeling robot kinematics, which is essential for planning trajectories, inverse kinematics, and overall robot control.

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

17/06/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.



Student’s Signature: …………………………… Date…………………………………………..

17/06/2023

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………..

Signature: …………………………………….. Date: ………………………………………..…

**WEEKLY PROGRESS CHART (WEEK ENDING: ----FIVE )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Had a discussion with the supervisor on a new project which was a clocking device for workers in a company where the data is sent to company’s server using Ethernet module. |
| TUES | We assessed the project that had been made previously that did the same but didn't have an Ethernet module instead had an ESP32 that would send the data to the server. |
| WED | Familiarized with the project components which were: 2.8" TFT LCD Screen, Ada-fruit fingerprint sensor, Arduino Mega 2560 and 4x4 keypad system. Selected the pins that we would use on connect each component to the Arduino mega 2560 micro-controller. Started prototyping the whole system on a perf-board by connecting the different components to each other. |
| THUR | Continued connecting and soldering the components to the perf-board. Started by doing research on the TFT lcd screen and displaying letters on it we also wrote a code to test this. |
| FRI | Did more research on how to use the fingerprint sensor and the working principle behind it. We wrote a code to enroll a fingerprint, delete a fingerprint and checking which fingerprint is already registered. |
| SAT |  |

## TRAINEES WEEKLY REPORT

## A clocking device with a fingerprint sensor, a 4x4 keypad, and Ethernet connectivity provides an advanced solution for time tracking and attendance management. This device combines multiple features to ensure accurate identification and seamless data transmission to a central server. Here's a breakdown of the theory behind its components and operation

## **1. Fingerprint Sensor:**

## - The fingerprint sensor captures the unique patterns and ridges on an individual's fingertip.

## - It uses optical or capacitive technology to obtain high-resolution images of the fingerprint.

## - The captured fingerprint image is converted into a digital template using algorithms that extract key features for identification.

## - The device compares the captured fingerprint template with stored templates to authenticate the user's identity

## **2. 4x4 Keypad:**

## - The 4x4 keypad consists of sixteen buttons arranged in a grid of four rows and four columns.

## - Each button corresponds to a specific numeric or alphanumeric value.

## - The keypad enables employees to enter additional information, such as their employee ID or PIN code, for identification purposes.

## **3. Ethernet Connectivity:**

## - The device is equipped with an Ethernet port for connecting to a local network or the internet.

## - Ethernet allows the clocking device to communicate with a central server, transmitting clocking data and receiving configuration updates.

## - The device can use protocols such as TCP/IP to establish a network connection and securely transfer data to the server.

## **4. Clocking Process:**

## - An employee approaches the clocking device and selects the clock-in or clock-out option on the display screen.

## - To authenticate their identity, the employee can choose between the fingerprint sensor or keypad input.

## - If using the fingerprint sensor, the employee places their finger on the sensor, and the device matches it against stored templates for identification.

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

24/06/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

## - If using the keypad, the employee enters their unique employee ID or PIN code.

## - Once the employee's identity is verified, the device records the time stamp and associated information (e.g., employee ID) for the clock-in or clock-out event.

## - The clocking device then establishes an Ethernet connection with the server and sends the collected data for storage and processing.

## **5. Data Transmission and Server Integration:**

## - The clocking device uses the Ethernet connection to securely transmit clocking data to a central server.

## - The server receives the data and updates the employee attendance records accordingly.

## - The server may perform additional functions such as generating attendance reports, calculating work hours, and integrating with payroll systems.

## - Integration with the server allows real-time monitoring of employee attendance and simplifies administrative tasks.

Student’s Signature: …………………………… Date…………………………………………..

24/06/2023

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………..

Signature: …………………………………….. Date: ………………………………………..…

**WEEKLY PROGRESS CHART (WEEK ENDING: -----SIX )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Wrote the code for the 4x4 keypad that could be used type in data which could be a password. Also had the system check whether the password typed is the same as the default/ set one and act accordingly to the results. |
| TUES | Combined the code on the keypad, fingerprint sensor and the tft-lcd screen, adjusted it to work simultaneously. Started the design of the admin user interface and working according to the choice the admin picks. |
| WED | Fixing the bugs that occurred and continued working on the admin user interface |
| THUR | PUBLIC HOLIDAY |
| FRI | We researched on the pins that could be used to connect the Ethernet module to our Arduino mega 2560 and added the W5500 Ethernet module to our prototype PCB by soldering it on the perf-board. |
| SAT |  |

## TRAINEES WEEKLY REPORT

## **FINGERPRINT SENSOR**

An optical fingerprint sensor is a biometric device that captures and analyzes the unique patterns and characteristics of an individual's fingerprint for identification and authentication purposes. This type of sensor employs optical technology to capture high-resolution images of fingerprints and convert them into digital data. Here's a theoretical explanation of how an optical fingerprint sensor typically works:

1. Light Source: The sensor consists of a light source, usually an LED (Light Emitting Diode), that emits light onto the surface where the finger is placed. The light source provides illumination to enhance the visibility of the fingerprint ridges and valleys.
2. Finger Placement: The user places their finger on a designated area of the sensor, usually a transparent glass or plastic surface. The finger is pressed gently against the surface, ensuring good contact between the skin and the sensor.
3. Reflection and Scattering: When the light from the source reaches the finger's surface, it undergoes reflection and scattering. The ridges on the finger's skin reflect light differently than the valleys, creating variations in the light patterns.
4. Image Capture: An optical lens or an array of lenses collects the reflected and scattered light from the finger's surface. The lens focuses the light onto an optical sensor or an image sensor, typically a complementary metal-oxide-semiconductor (CMOS) or charge-coupled device (CCD) sensor.
5. Image Formation: The captured light forms an optical image on the sensor, which consists of dark and light areas corresponding to the fingerprint ridges and valleys. The sensor converts this optical image into an electrical signal proportional to the light intensity at each pixel.
6. Signal Processing: The electrical signal containing the fingerprint image is then processed by the sensor's integrated circuitry. This processing involves amplifying the signal, applying filters to enhance image quality, and correcting any distortion or noise present in the captured image.
7. Minutiae Extraction: The processed image is analyzed to identify and extract minutiae points, which are unique features of the fingerprint. Minutiae points include ridge endings, bifurcations, and other characteristics that help distinguish one fingerprint from another.
8. Template Creation: The extracted minutiae points are used to generate a fingerprint template, which is a compact representation of the fingerprint. This template typically consists of mathematical algorithms or data structures that encode the unique characteristics of the fingerprint.
9. Comparison and Matching: When a user attempts to authenticate their identity, the newly captured fingerprint is processed and compared against the stored template(s) in a database. The matching algorithm compares the extracted minutiae points from the captured fingerprint with those in the template(s) to determine a match or non-match.
10. Authentication Result: Based on the comparison result, the optical fingerprint sensor generates an authentication decision, indicating whether the captured fingerprint matches any stored template(s) or not. This decision is then communicated to the system or device that requested the fingerprint authentication.

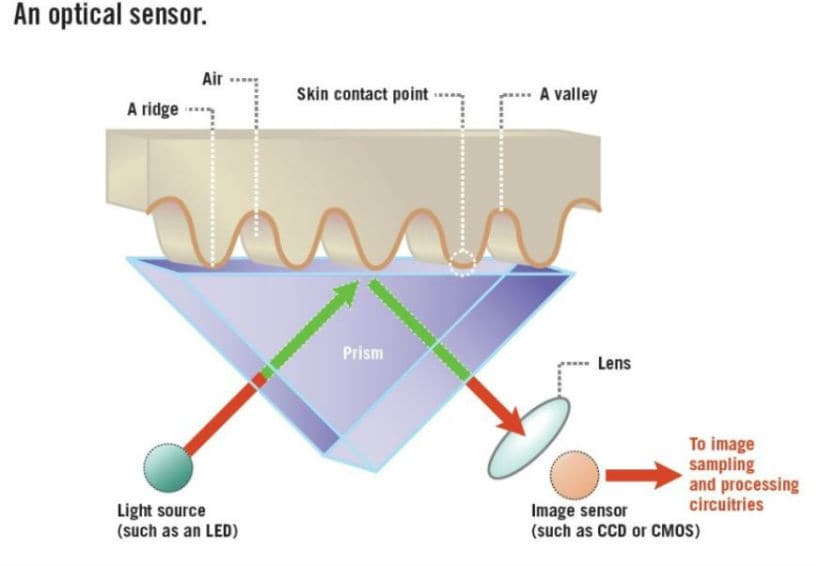
NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

01/07/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

* The working principle of the fingerprint sensor mainly depends on the processing. The fingerprint processing mainly includes two elements namely enrollment and matching.
* In fingerprint enrolling, every user requires to place the finger twice. So that the system will check the finger images to process as well as to generate a pattern of the finger and it will be stored. When matching, a user places the finger using an optical sensor then the system will produce a pattern of the finger & compares it with the finger library templates.
* For fingerprint enrollment, the system will evaluate the exits finger with a precise pattern which is selected within the module. For matching, the scanning system will look for the complete finger records for the finger matching. In both situations, the scanning system will go back to the corresponding result, success otherwise crash.



Student’s Signature: …………………………… Date…………………………………………..

01/07/2023

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………..

Signature: …………………………………….. Date: ………………………………………..…

**WEEKLY PROGRESS CHART (WEEK ENDING: ----SEVEN )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Familiarizing with the ethernet that is its setup, initialization and created a code to send data to a server with it. |
| TUES | Built a python server script file which would use to create our own server on a laptop and use it to test the if Ethernet was sending data to the server. |
| WED | We added an DS1302 RTC module and a memory card. We soldered the both to our prototyping board and wrote a code on the RTC module. |
| THUR | Wrote code on the SD card module and tested it in part of storing data to the SD card and retrieving data from the SD card. |
| FRI | Combined the whole code (RTC module, Ethernet module, memory card module, tft-lcd display, fingerprint sensor, 4x4 keypad) and tested it then submitted it to the supervisor. |
| SAT |  |

## TRAINEES WEEKLY REPORT

**REAL TIME CLOCK.**

The RTC (Real-Time Clock) module DS1307 is a commonly used integrated circuit that provides accurate timekeeping and date tracking in various electronic devices. It utilizes a combination of digital and analog circuitry to maintain real-time information even when the main power supply is turned off. Here's a theoretical explanation of how the RTC module DS1307 typically works:

1. Clock Source: The DS1307 RTC module incorporates an internal quartz crystal oscillator. This oscillator generates precise clock pulses at a frequency of 32.768 kHz. The quartz crystal ensures the stability and accuracy of the clock source.
2. Timekeeping Registers: The DS1307 includes a set of registers to store time and date information. These registers include seconds, minutes, hours (in both 12-hour and 24-hour formats), day of the week, day of the month, month, and year. The timekeeping registers are powered by a small on-board battery, typically a coin cell, to retain data when the primary power is disconnected.
3. I2C Interface: The DS1307 RTC module communicates with the micro-controller or host system using the I2C (Inter-Integrated Circuit) protocol. It acts as an I2C slave device, allowing the micro-controller to read and write data from/to the timekeeping registers. The module has a fixed I2C address (usually 0x68) that the micro-controller uses to initiate communication.
4. Clock Setting: To set the initial time and date, the micro-controller sends a series of I2C commands to the DS1307 module. These commands include specifying the appropriate values for the timekeeping registers. The module updates its registers accordingly and starts running the clock.
5. Timekeeping Operation: Once the DS1307 is set, it begins counting the clock pulses generated by the internal oscillator. The seconds register increments for every pulse received, and when it reaches 60, it rolls over to 0 and increments the minutes register. This cascading effect continues for minutes, hours, day of the week, day of the month, month, and year, following the appropriate rules for leap years.
6. Time Retrieval: The micro-controller can read the time and date information from the DS1307 by sending an I2C read command to the module. The module responds with the current values stored in its timekeeping registers. The micro-controller can then utilize this information for various time-related functions or display purposes.
7. Alarm Functionality: The DS1307 also offers alarm functionality, allowing the micro-controller to set specific time and date conditions to trigger an interrupt. The micro-controller can configure the alarm registers in the DS1307 by writing the desired alarm values over the I2C interface. When the current time matches the set alarm conditions, an interrupt signal is generated to alert the micro-controller.
8. Battery Backup: In the event of a power failure or when the primary power source is disconnected, the DS1307 switches to the backup power supply provided by the on-board battery. This battery ensures that the timekeeping registers retain their data, allowing the module to continue tracking time accurately. When the primary power is restored, the module automatically switches back to using it.

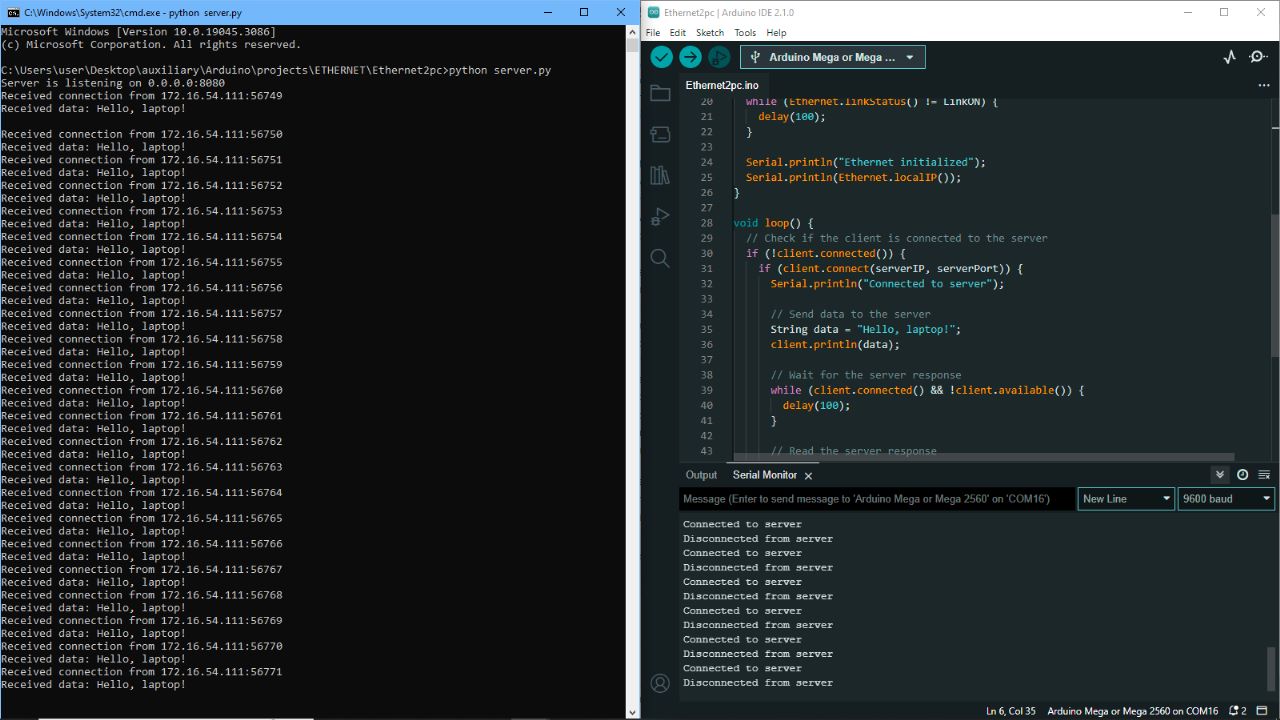
NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

08/07/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

**SERVER CONNECTION TEST**



08/07/2023

Student’s Signature: …………………………… Date…………………………………………..

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………

Signature: …………………………………….. Date……………………………………………..

**WEEKLY PROGRESS CHART (WEEK ENDING: ----EIGHT )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Had a discussion with the supervisor on the new project which is a dash-cam design for vehicle that transmits data to a central server. |
| TUES | Did research on the viable platforms that can be used on the project this included the micro-controller that would be effective and the best camera to use. |
| WED | Continued on the research of the micro-controller that we could use and came to a decision of using the Raspberry pi zero. |
| THUR | Stated studying on the implementation of inverse kinematics on a robotic arm. Checked the five standard types of robotic arms (3 degrees of freedom manipulators). |
| FRI | Checked the different methods of solving the inverse kinematics equations. Started by checking the Denavit Hartenberg method |
| SAT |  |

## TRAINEES WEEKLY REPORT

The proposed project entailed of a dashboard camera that had good recording quality and would save a recording if an accident happens and would send a message to nearby medical centers and any relevant emergency response centers giving off its live GPS location.

When considering the best micro-controller for recording 48 hours of dash-cam video, several factors were to be considered, including storage capacity, video encoding capabilities, processing power, price and power consumption. Below is a comparison of the proposed micro-controllers and microprocessors based on these criteria:

**Raspberry Pi Zero:**

Storage: The Raspberry Pi Zero can use micro-SD cards for storage, allowing for flexible storage capacity.

Video Encoding: It supports H.264 video encoding, which provides good video compression.

Processing Power: The Raspberry Pi Zero is relatively more powerful than other micro-controllers listed, offering sufficient processing power for video encoding.

Power Consumption: The power consumption of the Raspberry Pi Zero is higher compared to other micro-controllers.

**ESP32 Wrover:**

Storage: The ESP32 Wrover typically uses external SPI flash or SD cards for storage, providing flexibility in storage capacity.

Video Encoding: The ESP32 itself doesn't have built-in video encoding capabilities, but with proper libraries and external hardware, it can encode video in formats like H.264 or MJPEG.

Processing Power: The ESP32 is less powerful than the Raspberry Pi Zero but can still handle basic video encoding tasks.

Power Consumption: The power consumption of the ESP32 is generally lower than that of the Raspberry Pi Zero, which can be beneficial for battery-powered applications.

Arduino Mega:

Storage: The Arduino Mega doesn't have built-in storage capabilities, so an external storage solution like an SD card module would be required.

Video Encoding: The Arduino Mega doesn't have native video encoding capabilities. It may require additional hardware and libraries to encode video.

Processing Power: The Arduino Mega has limited processing power compared to the Raspberry Pi Zero and may struggle with real-time video encoding tasks.

## Power Consumption: The power consumption of the Arduino Mega is relatively low compared to more powerful micro-controllers, but it may still depend on the overall system design

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

**Arduino Mega:**

Storage: The Arduino Mega doesn't have built-in storage capabilities, so an external storage solution like an SD card module would be required.

Video Encoding: The Arduino Mega doesn't have native video encoding capabilities. It may require additional hardware and libraries to encode video.

Processing Power: The Arduino Mega has limited processing power compared to the Raspberry Pi Zero and may struggle with real-time video encoding tasks.

Power Consumption: The power consumption of the Arduino Mega is relatively low compared to more powerful micro-controllers, but it may still depend on the overall system design.

**Arduino Nano 33 BLE Sense:**

Storage: The Arduino Nano 33 BLE Sense doesn't have built-in storage capabilities, so external solutions like SD card modules can be used.

Video Encoding: The Arduino Nano 33 BLE Sense doesn't have native video encoding capabilities. It may require additional hardware and libraries for video encoding.

Processing Power: The Arduino Nano 33 BLE Sense has limited processing power and may face challenges in handling real-time video encoding.

Power Consumption: The power consumption of the Arduino Nano 33 BLE Sense is relatively low, which can be advantageous for battery-powered applications.

**Open-MV Cam:**

Storage: The Open-MV Cam supports micro-SD cards for storage, providing flexibility in storage capacity.

Video Encoding: The Open-MV Cam has built-in video encoding capabilities, supporting various formats like MJPEG and GIF.

Processing Power: The Open-MV Cam has specialized hardware for computer vision tasks, including video encoding, making it suitable for dash-cam applications.

Power Consumption: The power consumption of the Open-MV Cam is relatively low, making it suitable for battery-powered applications.

**Raspberry Pi Pico:**

Storage: The Raspberry Pi Pico can use external SPI flash or EEPROM for storage, but the storage capacity may be limited compared to micro-SD cards.

Video Encoding: The Raspberry Pi Pico doesn't have native video encoding capabilities, but it can be paired with external hardware and libraries for video encoding.

Processing Power: The Raspberry Pi Pico has limited processing power compared to the Raspberry Pi Zero, but it can handle basic video encoding tasks.

Power Consumption: The power consumption of the Raspberry Pi Pico is relatively low, which is beneficial for battery-powered applications.

Considering the requirements of recording 48 hours of dash-cam video, the Raspberry Pi Zero, Open-MV Cam, or ESP32 Wrover are more suitable options. They offer better processing power, storage capabilities, and video encoding support compared to the Arduino Mega and Arduino Nano 33 BLE Sense.

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

15/07/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

The choice between them would depend on factors like ease of development, specific requirements, and available libraries for video encoding and storage. Since the raspberry pi zero has good video compression libraries and low cost despite its high camera cost we decided to use it for the initial prototype. It would also require a GSM 808L module, a memory card and an accelerometer to sense a possible collision or overturn of the vehicle.

15/07/2023

Student’s Signature: …………………………… Date…………………………………………..

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………

Signature: …………………………………….. Date……………………………………………..

**WEEKLY PROGRESS CHART (WEEK ENDING: ----NINE )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Checked the different rules needed in order to solve inverse kinematics problem using the Denavit Hartenberg method. And drawing the kinematic diagrams for the different manipulator |
| TUES | Drew the different kinematic diagrams for the five standard types of three degrees of freedom manipulators and also added the axes for all the joints. |
| WED | Worked on the rotation matrices (Rotation matrices expresses the rotation of flame n relative to another frame n) for the five standard types of manipulators. |
| THUR | Worked on the derivation of X, Y and Z rotation matrix i.e the rotation that happens on the specific axis. Worked on the displacement vectors for the 5 standard manipulators. |
| FRI | Worked on the derivation of the homogeneous transformation matrix (H.T.M). And the Denavit Hartenberg table. |
| SAT |  |

## TRAINEES WEEKLY REPORT

**INVERSE KINEMATICS**

Inverse Kinematics (IK) is an essential problem in robotics and computer graphics, where the goal is to determine the joint angles or positions required to achieve a desired end-effector pose. Geometric methods provide one approach to solving IK problems for simple kinematic chains. This report explores the use of geometric methods in the context of inverse kinematics.

* Geometric Approach to Inverse Kinematics:

The geometric method for solving IK problems relies on geometric relationships and trigonometry to find the joint angles that position the end effector as desired. It is suitable for systems with simple kinematic chains, typically consisting of two or three links. The key idea is to exploit geometric relationships among the links to compute the joint angles analytically.

* The Law of Cosines and Trigonometry:

In a two-link planar arm, the law of cosines can be applied to relate the joint angles to the position of the end effector. For example, given the desired end-effector position (x, y), the joint angles θ1 and θ2 can be computed as follows:

θ2 = arc-cos((x^2 + y^2 - L1^2 - L2^2) / (2 \* L1 \* L2))

θ1 = arc-tan(y / x) - arc-tan((L2 \* sin(θ2)) / (L1 + L2 \* cos(θ2)))

where L1 and L2 are the lengths of the two links, and (x, y) is the target position.

* Geometric Transformations:

For more complex systems, homogeneous transformations can be used to represent the link positions and orientations. By chaining these transformations from the base link to the end effector, the overall transformation matrix can be derived. From this matrix, the joint angles can be extracted using trigonometric functions or decomposition methods like singular value decomposition (SVD).

* Limitations:

While geometric methods are straightforward and can provide analytical solutions for simple systems, they have certain limitations:

- Limited Applicability: Geometric methods are most effective for kinematic chains with few links and simple geometries. They become increasingly

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

22/07/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

impractical for complex robotic arms or structures with multiple degrees of freedom.

- Singularities: Geometric solutions can sometimes lead to singular configurations where the IK problem becomes undefined or leads to division by zero. Special handling is required to avoid such singularities.

* Conclusion:

Geometric methods offer a straightforward and intuitive approach to solving inverse kinematics problems for systems with simple kinematic chains. These methods are useful for educational purposes and can serve as a starting point for understanding IK concepts. However, they have limitations in dealing with complex systems, higher degrees of freedom, and singularities. For more sophisticated robotic systems and real-world applications, numerical methods and advanced algorithms become necessary to tackle the challenges of inverse kinematics effectively. Researchers and engineers continue to explore hybrid approaches that combine geometric and numerical methods to achieve efficient and accurate IK solutions for diverse applications in robotics and computer graphics.

22/07/2023

Student’s Signature: …………………………… Date…………………………………………..

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………

Signature: …………………………………….. Date……………………………………………..

**WEEKLY PROGRESS CHART (WEEK ENDING: ----- TEN )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Continuation on derivation of the Denavit Hartenberg for the different manipulators such as cylindrical, spherical, articulated, SCARA and Cartesian manipulator |
| TUES | Continuation on derivation of the homogeneous transformation matrices from the Denavit Hartenberg parameter table for the five major types of three degrees of freedom manipulator |
| WED | Using the homogeneous transformation matrix to find the forward kinematics for the manipulators determined on the previous days. Introduction of the peter corkes simulation toolbox for matlab. |
| THUR | Continuation on familiarization of the peter corkes simulation toolbox and used it to simulate forward kinematics for the 3 D.O.F, 4 D.O.F,5 D.O.F and 6 D.O.F manipulators. |
| FRI | Started solving the inverse kinematics problem for a 3 D.O.F manipulator using the geometric method. And simulated it using the peter corkes simulation toolbox in matlab |
| SAT |  |

## TRAINEES WEEKLY REPORT

**CODE USED FOR THE SOLVING THE INVERSE KINEMATICS FOR 3 D.O.F**

clc

disp('IK FOR 3 D.O.F ARTICULATED MANIPULATOR')

%% NOTE AM USING BOTH PARTS OF THE CALCULATIONS i.e THE ELBOW DOWN PART AND ELBOW UP PART

%% Desired position

X = 1;

Y = 2;

Z = 4;

%% link lengths

a1 = 2;

a2 = 2;

a3 = 2;

%% Equation 1

Theta01 = atand(Y/X);

%round(Theta01);

if (Y<0 && X< 0)

Theta0 = 180 - (Theta01\*-1);

%disp(Theta0);

end

if(Y>0 && X<0)

Theta0 = 180 - (Theta01\*-1);

%disp(Theta0);

end

if(Y>0 && X>0)

Theta0 = Theta01;

end

if(Theta0 > 180 || Theta0 < 0)

error("The point can't be reached");

end

disp ('Theta0 = ')

disp(Theta0);

%% Equation 2

r = sqrt((X^2)+(Y^2));

%% Equation 3

r2 = Z - a1;

%% Equation 4

r3 = sqrt((r2^2)+(r^2));

%% Equation 5

Alpha1= acosd((a3^2 - a2^2 - r3^2)/(-2\*a2\*r3));

%% Equation 6

Alpha2 = atand(r2/r);

%% Equation 7

%Theta1 = Alpha2 - Alpha1; %%For elbow down calculations

Theta1 = Alpha2 + Alpha1; %% For elbow up calculations

%Theta1 = round(Theta1);

if(Theta1 > 180 || Theta1 < 0)

error("The point can't be reached");

end

disp('Theta1 = ')

disp(Theta1);

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

%% Equation 8

Alpha3 = acosd((r3^2 - a2^2 -a3^2)/(-2\*a2\*a3));

%% Equation 9

%Theta2 = 180 - Alpha3; %%For elbow down calculations

Theta2 = (180 - Alpha3)\*-1; %% For elbow up calculations

%Theta2 =round(Theta2);

if(Theta2 > 90 || Theta2 < -90)

error("The point can't be reached");

end

disp("Theta2 = ")

disp(Theta2);

%% Convert the angles to radians

t1 = deg2rad(Theta0);

t2 = deg2rad(Theta1);

t3 = deg2rad(Theta2);

%% Robot arm

H0\_1 = Link([0, a1, 0, pi/2, 0, t1]);

H0\_1.qlim = [0 2\*pi];

H1\_2 = Link([0, 0, a2, 0, 0, t2]);

H1\_2.qlim = [-pi/6 pi/2];

H2\_3 = Link([0, 0, a3, 0, 0, t3]);

H2\_3.qlim = [-pi/2 pi/2];

Arti = SerialLink([H0\_1 H1\_2 H2\_3], 'name', 'Arti');

Arti.plot([0 0 0], 'workspace', [-5 5 -5 5 0 10]);

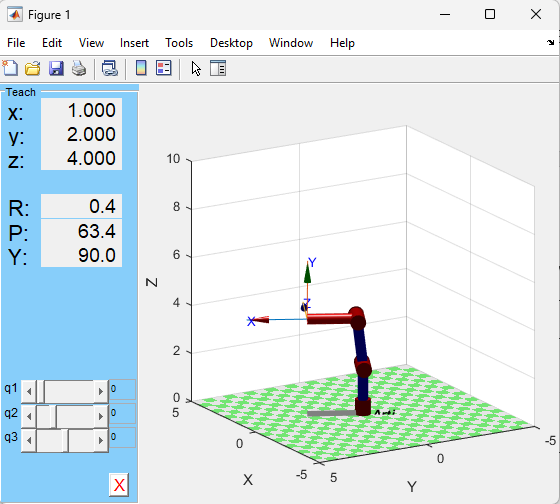
Arti.teach

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

29/07/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

**SIMULATION USING MATLAB**



29/07/2023

Student’s Signature: …………………………… Date…………………………………………..

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………

Signature: …………………………………….. Date……………………………………………..

**WEEKLY PROGRESS CHART (WEEK ENDING: ---ELEVEN---)**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Worked on the inverse kinematics for a 4 d.o.f manipulator. This involved the geometric calculation needed to achieve the desired position for the 4 D.O.F manipulator |
| TUES | Continued working on the inverse kinematics of a 4 D.O.F manipulator. |
| WED | Still working on the inverse kinematics for a 4 D.O.F manipulator i.e adding a pitch angle for the manipulator. |
| THUR | Created a forward kinematics diagram for a 4 D.O.F manipulator and worked out the calculations for the arm. |
| FRI | Created a code for the arm and tested it getting an error due to there not existing the cos inverse for a value greater than 1 and less than -1 which were the results that I got from my calculations. |
| SAT |  |

## TRAINEES WEEKLY REPORT

## **CODE WITH AN ERROR**

import numpy as np

#desired end points

x = 250

y = 0

z = 50

#link lengths

a1 = 100

a2 = 100

a3 = 100

a4 = 100

#orientation of the end effector

pitch = -1.57

pitch = np.rad2deg(pitch)

#calculation fo theta1

theta1 = np.arctan2(y, x)

print("Theta1:", theta1)

#intermediate calculations

rd = np.sqrt((x \* x) + (y \* y))

#finding the lenths that occur due to end effector orientation

zt = z - (a4 \* np.sin(np.deg2rad(pitch)))

rt = rd - (a4 \* np.cos(np.deg2rad(pitch)))

#calculatins for theta2

zo = zt - a1

alpha = np.arctan2(zo, rt)

s = np.sqrt(((rt \* rt) + (zo \* zo)))

beta = np.arccos((a3 \* a3 - a2 \* a2 - s \* s) / (-2 \* a2 \* s))

theta2 = alpha + beta

print("Theta2:", np.rad2deg(theta2))

#calculation fo theta3

gamma = np.arccos((s \* s - a2 \* a2 - a3 \* a3) / (-2 \* a2 \* a3))

theta3 = -(3.141593 - gamma)

print("Theta3:", np.rad2deg(theta3))

#calculation for theta4

theta4 = pitch - theta2 - theta3

print("Theta4:", np.rad2deg(theta4))

NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

05/08/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

For a robotic arm with more than three degrees of freedom, in order to simplify the calculation, we assume the first three degrees of freedom are responsible for positioning of the end-effector and the rest as responsible for orientation of the end effector. This introduces yaw, pitch and roll. Pitch, yaw, and roll are terms used to describe the rotations of an object, typically in three-dimensional space. These terms are commonly used in the context of aerospace, robotics, and computer graphics to describe how an object is oriented in relation to a reference frame.

Pitch, yaw, and roll are terms used to describe the rotations of an object, typically in three-dimensional space. These terms are commonly used in the context of aerospace, robotics, and computer graphics to describe how an object is oriented in relation to a reference frame.

Yaw is a rotation around the vertical axis, typically referred to as the "Y-axis." It involves a left or right rotation of an object's front.

Roll is a rotation around the front-to-back axis, often referred to as the "Z-axis." It involves the tilting of an object's side from one side to the other.

05/08/2023

Student’s Signature: …………………………… Date…………………………………………..

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………

Signature: …………………………………….. Date……………………………………………..

**WEEKLY PROGRESS CHART (WEEK ENDING:----- TWELVE )**

|  |  |
| --- | --- |
| DAY | **DESCRIPTION OF WORK DONE** |
| MON | Doing more research on how to solve the error on different textbooks. |
| TUES | More research on the error and ways to fix it one being using the arc-tan function with sintheta2 and costheta2 as its arguments but another error arose for the sintheta2 |
| WED | Found a research paper on 5 D.O.F and checked its calculation and created a python code for it. |
| THUR | Continued checking the solution of the research paper and did a simulation on the peter corkes matlab toolbox. |
| FRI | Tried downgrading the 5 D.O.F to the 4 D.O.F this was to be done by removing the theta 5 and getting the solutions for 4 D.O.F manipulator. |
| SAT |  |

## TRAINEES WEEKLY REPORT

# This line imports the NumPy library, which is used for numerical computations in Python.

import numpy as np

# These lines define the values for the end-effector position (x, y, z), the pitch and yaw angles, and the link lengths (link\_1, link\_2, link\_3, link\_5).

# These are the input values for the inverse kinematics calculation.

x = 200

y = 200

z = 100

pitch = -1.57

yaw = 1.57

link\_1 = 100

link\_2 = 100

link\_3 = 100

# link\_4 = 100

link\_5 = 100

# c\_1 and s\_1 store the cosine and sine of theta\_1, respectively.

theta\_1 = np.arctan(y / x)

c\_1 = np.cos(theta\_1)

s\_1 = np.sin(theta\_1)

# These calculations use the pitch angle and link lengths.

b\_2 = link\_1 - (link\_5 \* np.cos(pitch)) - z

b\_1 = c\_1 \* x + s\_1 \* y + link\_5 \* np.sin(pitch)

# calculations for theta 3

c\_3 = ((b\_1 \* b\_1) + (b\_2 \* b\_2) - (link\_2 \* link\_2) - (link\_3 \* link\_3)) / (

    2 \* link\_2 \* link\_3

)

theta\_3 = np.arccos(c\_3)

s\_3 = np.sin(theta\_3)

# calculations for theta 2

c\_2 = (((link\_2 + link\_3 \* c\_3) \* b\_1) + ((link\_3 \* s\_3) \* b\_2)) / (

    (link\_2 \* link\_2) + (link\_3 \* link\_3) + (2 \* link\_2 \* link\_3 \* c\_3)

)

theta\_2 = -np.arccos(c\_2)

# calculatons for theta 4

theta\_4 = pitch - theta\_2 - theta\_3

# calculations for theta 5

R\_31 = -(np.cos(yaw) \* np.sin(theta\_2 + theta\_3 + theta\_4))

R\_32 = np.sin(theta\_2 + theta\_3 + theta\_4) \* np.sin(yaw)

theta\_5 = np.arctan(((R\_32) / ((-R\_31))))

print("-------------------------------------------------------------------------")

# angles are printed in radians

print(theta\_1, theta\_2, theta\_3, theta\_4, theta\_5)

print("-------------------------------------------------------------------------")

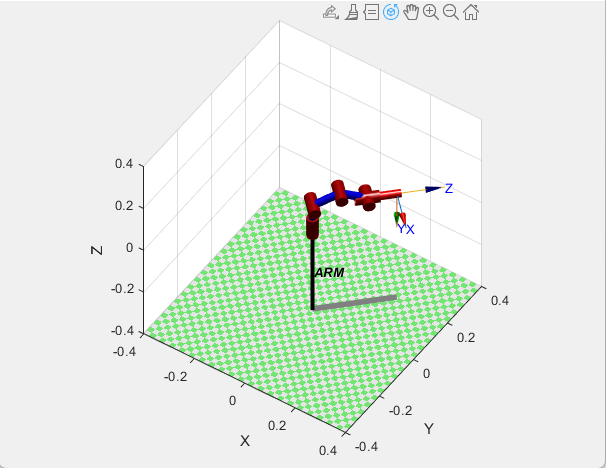
NOTE: USE BACK PAGE FOR THE CONTINUATION OF THE REPORT

## FOR SKETCHES, DIAGRAMS AND GRAPHS DATE………………..

12/08/2023

(Additional drawings, may be attached where necessary) The student may also use this space for additional reports.

**5 D.O.F ROBOTIC ARM**



12/08/2023

Student’s Signature: …………………………… Date…………………………………………..

Comments by Lecturer/Supervisor: ………………………………………………………………..

…………………………………………………………………………………………………………….

……………………………………………………………………………………………………………. Name: ………………………………………………………………………………………………

Signature: …………………………………….. Date……………………………………………..

# FOR THE USE OF THE INSTITUTION SUPERVISOR ONLY

General comments on first/second/third (delete as appropriate) Visit.

Name of the Supervisor: …………………………………………………………………………

Signature of the Supervisor: …………………………………………………………………….

Date: ………………………………………………………………………………………………..