Laser Pointer Turret Based Mosquito Air Defence System

Final Report

A. Hartman
20475323

Submitted as partial fulfilment of the requirements of Project EPR402
in the Department of Electrical, Electronic and Computer Engineering
University of Pretoria

November 2023

Study leader: Prof. P. de Villiers

Part 1. Preamble

This report describes work that I did *<to be completed>*.

Project proposal and technical documentation
This main report contains an unaltered copy of the approved Project Proposal (as Part 2 of the report).

Technical documentation appears in Part 4 (Appendix).

All the code that I developed appears as a separate submission on the AMS.

Project history
This project makes extensive use of existing algorithms on ... Some of the algorithms I used were adapted from ... Where other authors' work has been used, it has been cited appropriately, and the rest of the work reported on here, is entirely my own.

Language editing
This document has been language edited by a knowledgeable person. By submitting this document in its present form, I declare that this is the written material that I wish to be examined on.

My language editor was ____________________________________.

____________________________________ ____________________

Language editor signature *Date*

Declaration

I, <u>A. Hartman</u> understand what plagiarism is and have carefully studied the plagiarism policy of the University. I hereby declare that all the work described in this report is my own, except where explicitly indicated otherwise. Although I may have discussed the design and investigation with my study leader, fellow students or consulted various books, articles or the internet, the design/investigative work is my own. I have mastered the design and I have made all the required calculations in my lab book (and/or they are reflected in this report) to authenticate this. I am not presenting a complete solution of someone else.

Wherever I have used information from other sources, I have given credit by proper and complete referencing of the source material so that it can be clearly discerned what is my own work and what was quoted from other sources. I acknowledge that failure to comply with the instructions regarding referencing will be regarded as plagiarism. If there is any doubt about the authenticity of my work, I am willing to attend an oral ancillary examination/evaluation about the work.

I certify that the Project Proposal appearing as the Introduction section of the report is a verbatim copy of the approved Project Proposal.

A. Hartman

Date

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CCL	connected components labelling
GPIO	general purpose input/output
GPU	graphics processing unit
MADS	mosquito air defence system
PID	proportional-integral-derivative
RGB	red, green, blue
RPM	revolutions per minute

Part 2. Project definition: approved Project Proposal

This section contains the problem identification in the form of the complete approved Project Proposal, unaltered from the final approved version that appears on the AMS.

For use by the Project lecturer	Approved		Revision required	

Feedback

To be completed by the student

PROJECT PROPOSAL 2023				Project no	PdV3	Revision no	1
Title	Surname	Initials	Student no	Study leader (title, initials, surname)			
Mr	**Hartman**	**AM**	20475323	Prof P. de Villiers			
Project title							
Laser pointer turret based mosquito air defence system							

Language editor details	Language editor signature
Vedre Hartman	
Student declaration I understand what plagiarism is and that I have to complete my project on my own.	**Study leader declaration** This is a clear and unambiguous description of what is required in this project. Approved for submission (Yes/No)
Student signature	Study leader signature and date

1. Project description

What is your project about? What does your system have to do? What is the problem to be solved?

The problem addressed in this project is the development of an automated system to shoot mosquitoes using a non-lethal laser turret. A non-lethal system is developed as a proof of concept. Mosquitoes carry dangerous diseases, such as malaria and dengue fever. Insect repellents are not 100% effective and need to be regularly reapplied to remain effective. Thus, the problem addressed in this project is to develop a new type of mosquito air defence system.

The concept is to detect mosquitoes using a camera and then "shoot" them down using a laser that is controlled by a turret system. In this project the mosquitoes will be in an enclosed tank. The project will entail the first principle design of an object detection algorithm that can recognise mosquitoes, a multi-target tracking algorithm which can track and make predictions for multiple targets, and a control system for the laser turret which can “shoot” at the target mosquito.

2. Technical challenges in this project

Describe the technical challenges that are *beyond* those encountered up to the end of third year and in other final year modules.

2.1 Primary *design* challenges

A fundamental challenge in the design of the laser pointer turret based mosquito air defence system is the reliable detection and localisation of mosquitoes. To achieve this, an algorithm must be developed that can associate detections of mosquitoes over time to track their movement. Additionally, a reliable prediction algorithm must be developed to predict the future state of mosquitoes based on their previous states. A core design challenge is to convert the position of a detection into an angular coordinate that the laser turret can target. The laser turret system must target a position with sufficient accuracy to illuminate the body of a mosquito while being able to move with enough speed to track it.

2.2 Primary *implementation* challenges

A primary implementation challenge is real-time detection and association of mosquito movements over time, with prediction of their future states, all on an embedded platform. A suitable model to represent mosquito motion must be selected to create an accurate state prediction model. Developing a control model for the laser turret and detecting the actual position of the laser is also a significant challenge. Implementing a turret system will sufficient speed and accuracy poses a critical challenge. Obtaining sufficient mosquitoes for testing and demonstration is an additional practical challenge.

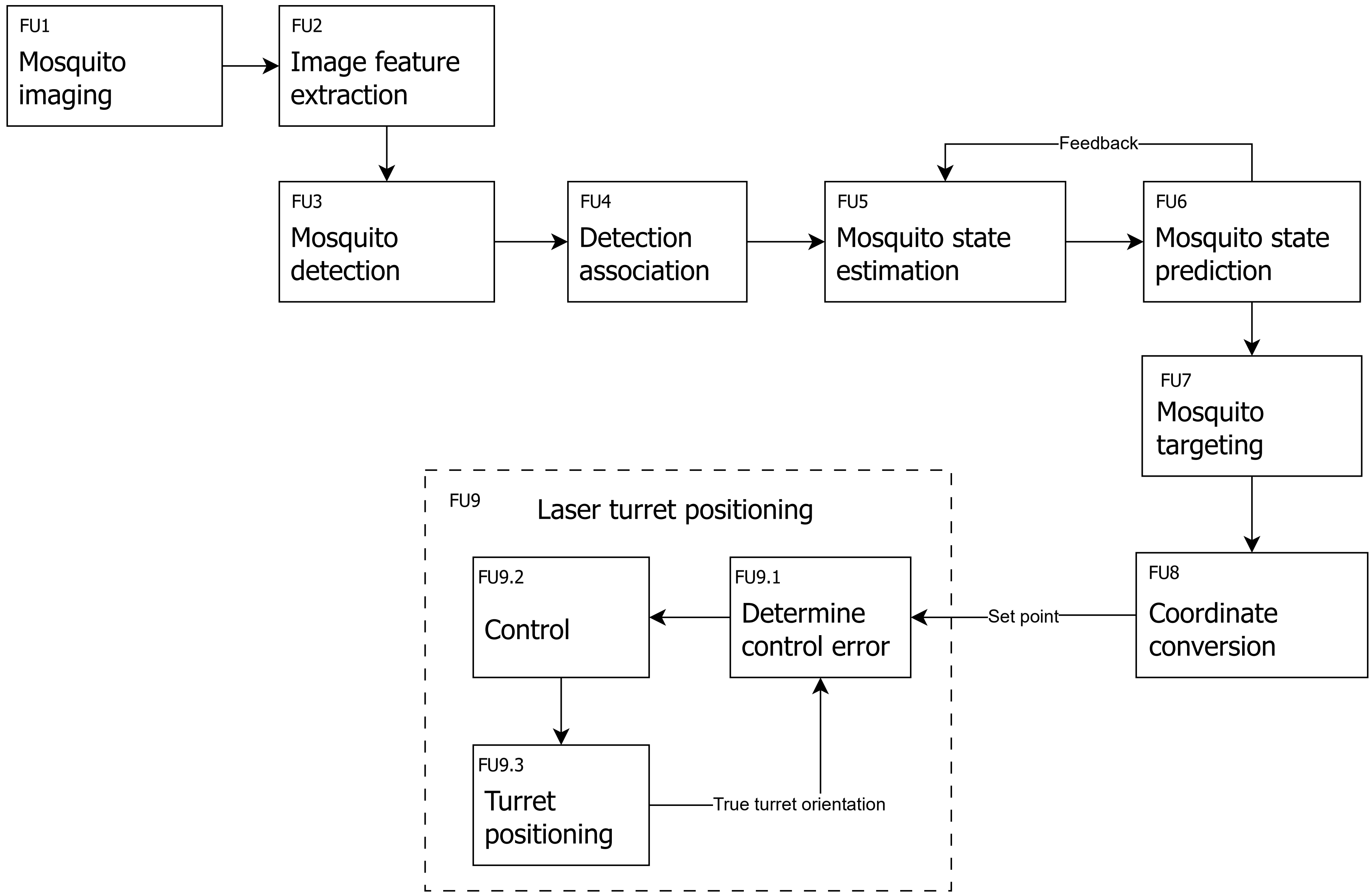
3. Functional analysis

3.1 Functional description

Describe the design in terms of system functions as shown on the functional block diagram in section 3.2. This description should be in *narrative format*. DO NOT use a bullet list.

The system will image the mosquito enclosure (FU1). The system will then extract the features from the image (FU2). The system will detect all the moving mosquitoes in the enclosure (FU3). The system will then associate the mosquito detections over time to determine the past movement of the mosquitoes (FU4). The system will then determine the current state of the mosquitoes using the movement history (FU5). The system will then use the current state of the mosquitoes and a model for the movement of mosquitoes to predict the future state of the mosquitoes (FU6). The system will then select a mosquito to target (FU7). The position of the target mosquito is then converted from rectangular coordinates into angular coordinates that can be targeted by the laser turret (FU8). The laser turret must the move to the target position. The system must then detect the actual position of the laser. This will be used as feedback to adjust the position of the laser to reach the target position (FU9).

3.2 Functional block diagram (this should not be a flow diagram)



4. System requirements and specifications

These are the core requirements of the system or product (the mission-critical requirements) in table format **IN ORDER OF IMPORTANCE**. Requirement 1 is the most fundamental requirement.

	Requirement 1: the fundamental functional and performance requirement of your project	**Requirement 2** (Number 2 in the order of importance)	**Requirement 3**
1. Core mission requirements of the system or product. Focus on requirements that are core to solving the engineering problem. These will reflect the solution to the problem.	The system must track and illuminate mosquitoes with a laser in an enclosure.	Detect moving mosquitoes in the enclosure.	Track mosquitoes over time. (Associate mosquitoes detections between frames.)
2. What is the target specification (in *measurable* terms) to be met in order to achieve this requirement?	The system must be able to illuminate a mosquito every 5 seconds.	The system must have a 0.9 probability of detection and a 0.05 probability of false alarm. Detection will be performed at an update interval of 500ms.	After 5 seconds there must be a minimum of 0.9 probability of correct association in each step of tracking the mosquitoes.
3. Motivation: Defend the specific target specification selected, i.e. the value. *Why* will meeting the specification given in point 2 above *solve the problem*?	5 seconds is a reasonable amount of time to acquire the next mosquito and reposition the laser to be ready for the next "shot".	90% is a high probability of detection and 5% is low enough that little time will be wasted on false alarms (mosquitoes that do not exist).	The system must be able to associate mosquitoes from one frame the next to target the mosquitoes. 90% is high yet reasonable probability of correct association. 90% is an acceptable rate for most tracking systems.
4. How will you demonstrate at the examination that this requirement and specification (points 1 and 2 above) have been met? Be explicit about how you will *prove* these were met.	The display will indicate the time that has elapsed since the targeting began until the mosquito has been illuminated. The constant llumination of mosquitoes by the laser will demonstrate correct functioning of the system.	The system will have a display that shows the video feed with the mosquitoes identified. This can be paused and inspected to verify that the requirement is met.	The display will show the tracks (lines) that follow a specific mosquito over time.
5. Your own design contribution: what are the aspects that *you will design and implement yourself* to meet the requirement in point 2? *If none, remove this requirement.*	The image processing system, the prediction algorithm of the mosquitoes future location, the targeting algorithm, and the control system for the laser turret.	The detection algorithm will be developed by the student and implemented on an embedded microprocessor.	The target tracking algorithm will be developed by the student.
6. What are the aspects to be taken off the shelf to meet this requirement? If none, clearly indicate "none". Explicitly indicate what tasks library functions will be used for (if relevant to the project).	The embedded platform, the camera, the laser, the motors, the motor drivers, and the power supply will be off the shelf.	The embedded platform, the camera, and the display will be off the shelf.	The embedded platform will be off the shelf.

System requirements and specifications page 2

	Requirement 4	Requirement 5	Requirement 6
1. Core mission requirements of the system or product. Focus on requirements that are core to solving the engineering problem. These will reflect the solution to the problem.	Control the position of the laser to illuminate the target position (set point).		
2. What is the target specification (in *measurable* terms) to be met in order to achieve this requirement?	The laser must be able to illuminate the set point within 2 seconds accurate to within 1mm.		
3. **Motivation:** Defend the specific target specification selected, i.e. the value. *Why* will meeting the specification given in point 2 above *solve the problem*?	The laser must be able to reach the set point within one second such that the overall system specification is reached. A typical mosquito is 2mm wide thus the precision must be at least half this.		
4. How will you demonstrate at the examination that this requirement and specification (points 1 and 2 above) have been met? Be explicit about how you will *prove* these were met.	The system will draw a 1mm circle around the set point and indicate if the laser reached this target within 2 seconds.		
5. Your own design contribution: what are the aspects that *you will design and implement yourself* to meet the requirement in point 2? If none, *remove this requirement.*	The turret model and control system with set point, settling time, and overshoot will be developed by the student.		
6. What are the aspects to be taken off the shelf to meet this requirement? If none, indicate "none". Explicitly indicate what tasks library functions will be used for (if relevant to the project).	The motors, turret housing, motor drivers, and the power supply will be off the shelf. The laser will be off the shelf.		

5. Field conditions

These are the REAL WORLD CONDITIONS under which your project has to work and has to be demonstrated.

	Real world field condition 1	**Real world field condition 2**	**Real world field condition 3**
Field condition requirement. In which field conditions does the system have to operate? Indicate the one, two or three most important field conditions.	The mosquitoes will be in an enclosed tank where the lighting conditions will be controlled.	If mosquitoes cannot be obtained a suitable substitute will be used.	
Field condition specification. What is the specification (in measurable terms) for this field condition?	The tank must be at least 1 metre wide. All the sides of the tank except the front facing side will have a white lining.	The substitute will be a similar flying insect.	

6. Student tasks

6.1 Design and implementation tasks

List your primary design and implementation tasks in bullet list format (5-10 bullets). These are *not* product requirements, but *your* tasks.

- The student needs to develop a mosquito detection algorithm.
- The student needs to develop a detection association algorithm.
- The student needs to develop a mosquito state prediction algorithm.
- The student needs to design a control system and interface for the laser turret.
- The student needs to select appropriate hardware for the laser turret.
- The student needs to select an appropriate camera.
- The student needs to acquire a laser that will be able to kill mosquitoes.

6.2 New knowledge to be acquired

Describe what the theoretical foundation to the project is, and which new knowledge you will acquire (*beyond* that covered in any other undergraduate modules).

- The student needs to learn how to develop computer vision algorithms.
- The student needs to learn how to associate, track, and predict the movement of objects.
- The student needs to learn how to model a laser turret.
- The student needs to learn how to interface and control a turret system.
- The student needs to learn how to interface and work with live video.
- The student needs to learn how to control motors.

Part 3. Main Report

1. Literature study

Malaria is still one of the leading causes of death in low-income countries according to the World Health Organisation [1]. Mosquitoes that do not carry diseases are also a general nuisance in the everyday life of people living in mosquito-prone areas. Therefore, it is necessary to pursue improvement in our defence against mosquitoes.

To be able to design a laser pointer turret-based mosquito air defence system it is necessary to understand the principles of computer vision object detection and real-time tracking.

One approach towards tracking is to perform pattern matching. In general pattern matching is searching and checking images for the presence of other given images (patterns) to find and mark the patterns' locations (if any) within the given images. However, the study conducted by Hurtik et al. [2] presents results that indicate the best frame rate they achieved was 0.43 frames per second. This is far too slow to be used in a real-time tracking application.

Another approach is to perform particle filter-based tracking. This considers the proximity and behaviour of other targets. In the case of social insect tracking, it is known that two targets cannot occupy the same space, and targets will actively avoid collisions. Unfortunately, the joint particle tracker proposed in [3] suffers from exponential complexity.

A popular approach is to separate the detection and tracking functions. While numerous deep learning algorithms can detect objects based on appearance, it is worth noting that mosquitoes, particularly when not filmed up close, prove too minute to be reliably detected using appearance-based methods. A viable alternative is to detect objects by isolating the background and foreground of the image [4]. The foreground of the image contains the objects of interest. In [5] objects that are too close to one another are split into two and abnormally small objects are merged.

A proposed tracking method is the Simple Online Real-time Tracking (SORT) algorithm [6]. The algorithm is composed of an estimation model which makes use of a Kalman filter and a data association system that is solved optimally using the Hungarian algorithm.

In the proposed mosquito air defence system mosquito detection will be based on background and foreground isolation. This method is suitable because the system will operate in a known test environment where the background will change minimally. The online real-time nature of this system makes the case for pattern matching and particle filtering unfavourable because of the computationally intensive nature of these techniques. The methods in [4], [5], and [6] will be further investigated for the proposed system.

2. Approach

The aim of this project was to develop a system that would illuminate mosquitoes with a laser turret. Throughout the remainder of this document this system will be referred to as the mosquito air defence system (MADS). The project can be viewed in terms of a set of integrated subsystems. The subsystems are as follows:

Laser Turret Control System
This subsystem is responsible for controlling the laser turret. The laser control system is discussed in

Laser Detection System
This subsystem is responsible for detecting the actual position of the laser with the camera. The laser detection is required to provide feedback to the laser turret control system. The laser detection system is discussed in

Mosquito Detection System
This subsystem is responsible for detecting the position of the mosquitoes. The mosquito detection system is discussed in

Mosquito Tracking System
This subsystem is responsible for tracking the mosquitoes and predicting their future positions. The mosquito tracking system is discussed in

The subsystems were integrated on a real-time embedded system and will be discussed in detail in the relevant sections. The system is designed to operate in a controlled environment constructed specifically for this project. The functional block diagram of the MADS is shown in Figure 1.

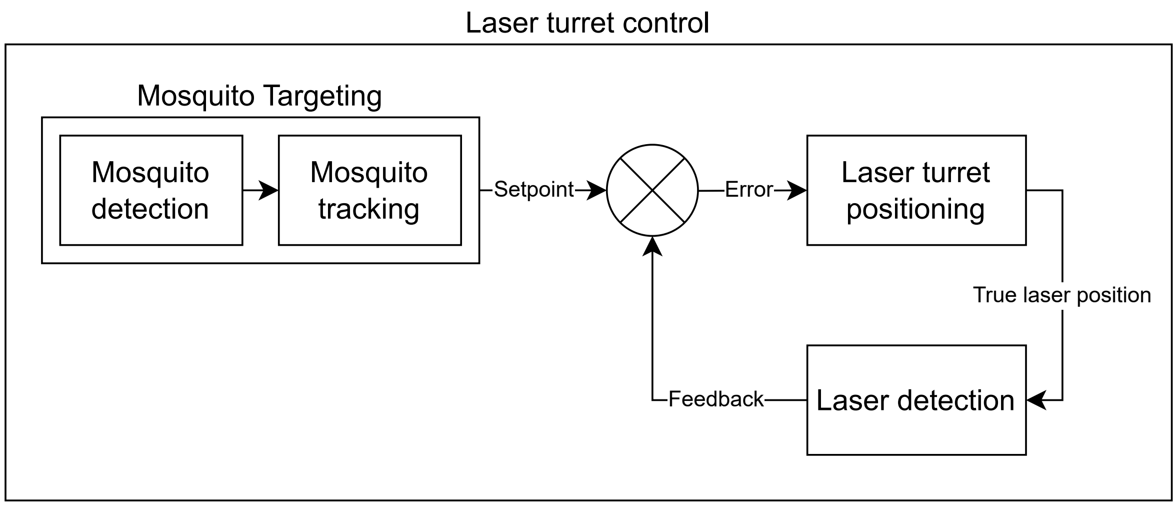


Figure 1.
Functional block diagram of the MADS.

The design choices for the MADS were made with careful consideration of the implicit real-time operation requirement of the system. All aspects of the MADS were designed to be as lightweight as possible in terms of computational complexity.

3. Design and implementation

3.1 Things I made

- Laser turret housing.
- Worked out the speed, torque, and step size required for the stepper motors.
- Interfaced with stepper motors using GPIO pins to drive the stepper motor drivers.
- Mapping between camera pixels and real-world co-ordinates. Had to do camera calibration and account for different perspectives of camera and turret.
- Calculate steps from angle required based on distance using Pythagoras.
- Distinguish between laser reflections using the geometry of the laser turret.
- Laser detection from first principles optimised with GPU kernels. 1. Gaussian smoothing, 2. Binarise with threshold, 3. Morphological operations (closing and opening), 4. Connected components labelling, 5. Find centroids of connected components, 6. Distinguish laser detections with turret geometry.
- Mosquito detection. Same image processing steps expect step 2 is either a less than threshold or background subtraction.
- Mosquito tracking using SORT algorithm. (Kalman filter and Hungarian algorithm).
- Laser turret PID controller. Must still be tuned because the error calculated is inaccurate. Should tuning be done without developing a model? Or should a model be developed? How do you develop a model?
- Feedback for turret. The current steps are saved at the instance that the frame is captured. The frame is then processed using the laser detection system. The pixel co-ordinates are converted to steps. The step error is calculated and added to the current steps.
- System integration on embedded system running in real-time on multiple threads.

3.2 Design summary

This section summarises the project design tasks and how they were implemented (see Table 1).

Deliverable or task	Implementation	Completion of deliverable or task, and section in the report
The mosquito detection subsystem had to be designed and implemented by the student.	The mosquito detection subsystem was designed and implemented from first principles.	Completed.
The laser detection subsystem had to be designed and implemented by the student.	The laser detection subsystem was designed and implemented from first principles.	Completed.
The laser turret control subsystem had to be designed and implemented by the student.	The laser turret control subsystem was designed and implemented from first principles.	Completed.
The mosquito tracking subsystem had to be designed and implemented by the student.	The mosquito tracking subsystem was designed and implemented from first principles.	Completed.
The various subsystems had to be integrated on a real-time embedded system.	The various subsystems were integrated on a real-time embedded system.	Completed.
Appropriate motors needed to be selected for the laser turret.	The stepper motors were selected based on the requirements of the laser turret.	Completed.

Table 1.
Design summary.

3.3 Theoretical analysis and modelling

3.3.1 Mapping pixel co-ordinates to metric co-ordinates

To control the laser, its position must be known in the world co-ordinate frame. The laser position was measured using a camera, thus the camera's pixel co-ordinate frame must be mapped to the world co-ordinate frame. To perform this mapping a camera model is required. The forward imaging model of a camera is shown in Figure 2.

Using the forward imaging model the pixel distance was mapped to the metric distance for the x-axis with

$$X = Z \times \left(\frac{x - x_{ref}}{f_x} \right) \tag{1}$$

where Z is the depth camera with respect to the world co-ordinate frame, x is the pixel of interest, x_{ref} is the reference pixel, and f_x is the effective focal length of the camera. Similarly,

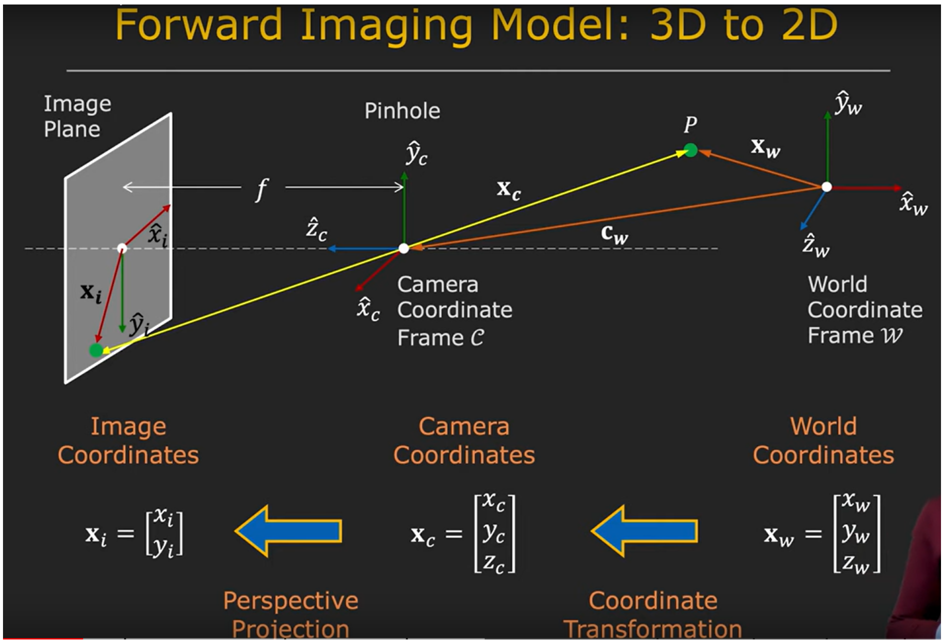


Figure 2.
Forward imaging model of a camera. CITE

the pixel distance was mapped to the metric distance for the y-axis.

The effective focal length of the camera f_x was determined through camera calibration.

3.3.2 Camera calibration

a. Intrinsic parameters

These parameters are inherent to the camera and remain constant unless the camera's internal settings (like focus) are changed. They are typically found by calibrating the camera using multiple views of a known pattern (like a checkerboard).

- **Camera Matrix (K)**: Defines the camera's internal characteristics. The principal components are:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

  where f_x and f_y are the focal lengths in pixels and c_x, c_y are the co-ordinates of the principal point. The principal point is the pixel co-ordinate where the camera's principal axis intersects the image plane.
- **Distortion Coefficients (D)**: Captures lens distortion. This is a vector with up to 5 elements in the common "plumb-bob" model of OpenCV

$$D = [k_1, k_2, p_1, p_2, k_3]$$

  where k_1, k_2, k_3 are radial distortion coefficients and p_1, p_2 are tangential distortion coefficients.

b. Extrinsic Parameters

These parameters capture the camera's orientation and position concerning the world or an external reference frame.

- **Rotation Vector (rvec)**: Represents the orientation of the camera. It's a 3x1 vector used to derive the rotation matrix R.
- **Translation Vector (tvec)**: Represents the position of the camera. It's a 3x1 vector capturing the translation in X, Y, and Z directions.

They represent the position and orientation of the camera relative to the world (or in your case, the turret's frame). Every time you move the camera or change the scene, these parameters would change. These are found using functions like `solvePnP` in OpenCV, which computes the pose of an object given some known 3D points on the object and their corresponding 2D projections in the image.

3.3.3 Morphological operations

Erosion is a fundamental morphological operation used to remove small structures or details from a binary image. It is defined as the basic set operation of moving a structuring element (usually a smaller binary matrix) over the input binary image and finding the intersection of the structuring element with the image. This operation can be mathematically expressed as

$$(A \ominus B)(x,y) = \bigcap \{A(x+i, y+j) | (i,j) \in B\} \tag{3}$$

where

- A is the input binary image.
- B is the structuring element.
- $\ominus$ represents the erosion operation.
- (x,y) are the pixel co-ordinates in the resulting image.

Dilation is another fundamental morphological operation, but it is used to enhance or grow the features in a binary image. Dilation can be defined as the set operation that moves the structuring element over the input image and computes the union of the element with the parts of the image where the structuring element "hits". The mathematical expression for dilation is as follows:

$$(A \oplus B)(x,y) = \bigcup \{A(x+i, y+j) | (i,j) \in B\} \tag{4}$$

where

- A is the input binary image.
- B is the structuring element.
- $\oplus$ represents the dilation operation.
- (x,y) are the pixel co-ordinates in the resulting image.

The erosion and dilation operations are illustrated in Figure 3.

(a)
The erosion of the dark-blue square by a disk, resulting in the light-blue square.

(b)
The dilation of a dark-blue square by a disk, resulting in the light-blue square with rounded corners.

Figure 3.
The figure shows an example of erosion and dilation. This figure was modified from Renatokeshet at the English Wikipedia (2008).

Opening is a compound morphological operation that consists of an erosion followed by a dilation using the same structuring element. It is primarily used to remove noise and small objects from a binary image. The opening of an image A by a structuring element B is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{5}$$

where

- A is the input binary image.
- B is the structuring element.
- $\circ$ represents the opening operation.

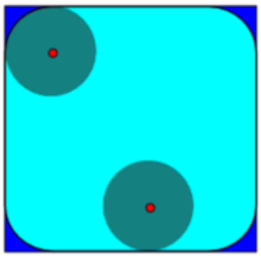
Closing is another compound morphological operation that consists of dilation followed by an erosion using the same structuring element. It is used to close small holes and gaps in objects and to connect nearby objects. The closing of an image A by a structuring element B is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{6}$$

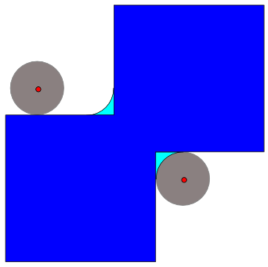
where

- A is the input binary image.
- B is the structuring element.
- $\bullet$ represents the closing operation.

Opening and closing are both idempotent operations, meaning that repeated openings or closing have no effect on the image. Opening and closing is illustrated in Figure 4.



(a)
The opening of the dark-blue square by a disk, resulting in the light-blue square with round corners.



(b)
The closing of the dark-blue shape (union of two squares) by a disk, resulting in the union of the dark-blue shape and the light-blue areas.

Figure 4.
The figure shows and example of opening and closing. This figure was modified from Renatokeshet at the English Wikipedia (2008).

3.3.4 Kalman filter

The Kalman filter is a recursive algorithm that estimates the state of a system from a series of noisy measurements. It is a powerful tool for tracking the state of a system over time and is widely used in control systems and robotics. The Kalman filter is based on a linear dynamical system model, which is defined by the following equations:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \tag{7}$$

where

- x_k is the state vector at time k.
- z_k is the measurement vector at time k.
- A is the state transition matrix.
- B is the control matrix.
- u_k is the control vector at time k.
- w_k is the process noise vector at time k.
- H is the observation matrix.
- v_k is the measurement noise vector at time k.

3.3.5 SORT tracking

SORT (Simple Online and Realtime Tracking) is a popular algorithm for multi-object tracking in video sequences. The main goal of the SORT algorithm is to associate detection boxes across frames in a video sequence to form trajectories for each individual object. Here is a theoretical background on the SORT tracking algorithm: 1. Overview

SORT is designed to be both simple and efficient, achieving real-time performance. It is based on the tracking-by-detection paradigm, which means it relies on an external object detector to provide bounding boxes around objects of interest in each frame. The algorithm then associates these detections across frames to create tracks. 2. Detection

Before tracking, an object detection algorithm (such as Faster R-CNN, YOLO, or SSD) is applied to each frame to detect objects of interest. The output is a set of bounding boxes along with their associated confidence scores. 3. Association

The core of the SORT algorithm is the association step, where detections are linked across frames to form tracks. This is done based on the intersection over union (IoU) metric, which measures the overlap between two bounding boxes. The steps involved are:

Prediction: For each existing track, the Kalman filter is used to predict the new position of the object in the current frame. Matching: The predicted positions of existing tracks are matched with the current frame's detections based on the IoU metric. A bipartite graph matching algorithm (such as the Hungarian algorithm) is used to find the optimal assignment. Update: The Kalman filter for each matched track is updated with the current detection. Creation and Deletion: New tracks are created for detections that could not be matched to existing tracks. Tracks that have not been matched for a certain number of frames are deleted.

4. Kalman Filter

The Kalman filter is a critical component of the SORT algorithm, providing a way to predict and update the state of the tracks based on the detections. The state of a track is represented as $[x, y, \dot{x}, \dot{y}]$, where x,yx,y are the coordinates of the center of the bounding box, ss is the scale/area of the bounding box, rr is the aspect ratio, and $\dot{x}, \dot{y}$ are the respective velocities. 5. Challenges and Limitations

Occlusions: SORT can struggle with objects that are occluded or interact closely with each other, as the IoU metric may not be sufficient for correct association. Identity Switches: In cases where objects cross paths or are close together, SORT can suffer from identity switches. Dependence on Detection: The performance of SORT is highly dependent on the quality of the detections provided by the external object detector.

3.4 Simulation and Prototyping

Detection and tracking done in python with videos.

3.5 Hardware design

Turret inspired by 2d laser scanners. Motor step resolution, speed, and torque calculations. Motor driver selection based on microstepping capabilities. Turret CAD design. Math translation between angle and distance of laser.

3.5.1 Mosquito enclosure and system positioning

The mosquitoes were placed inside an enclosure with a white lining and a glass front panel for the laser beam to shine through. The enclosure is a rectangular prism with dimensions $90 \times 38 \times 32$ cm as seen in Figure 5. Internal lighting was added to the enclosure to ensure contrast between the background and the mosquitoes and to minimise the camera noise. The laser turret and the camera were placed outside the enclosure in known positions relative to the enclosure. The bracket, shown in Figure 6, was designed to hold the laser turret and the camera in place relative to the mosquito enclosure.

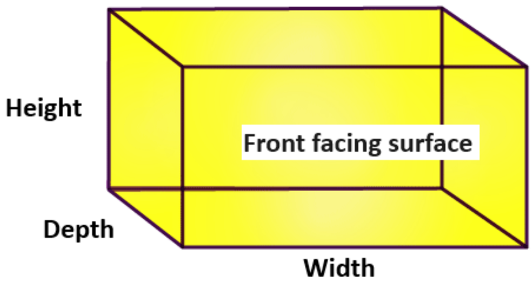


Figure 5.
Mosquito enclosure dimensions.

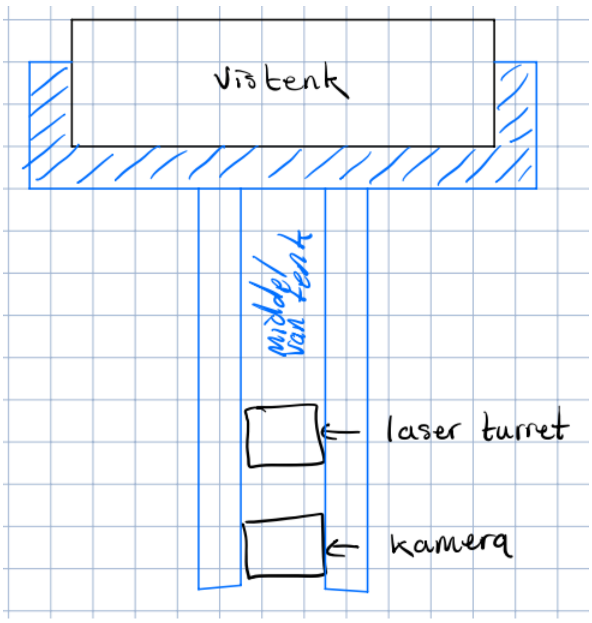


Figure 6.
System positioning bracket.

3.5.2 Laser turret design

The laser turret design was inspired by commercial two-axis laser scanners. A schematic for a typical two-axis laser scanner can be seen in Figure 7. The mirrors are connected directly to the output shaft of the motor. The origin of the laser turret is where the laser beam shines orthogonal to the mosquito plane, which occurs when the two mirrors are at 45° relative to the angle of the laser beam when the origin of the laser beam is parallel to the mosquito plane. When a single axis of the laser turret is considered it can be seen that a right triangle is formed between the laser turret and the mosquito plane as shown in Figure 8. Using the properties of a right triangle the mirror angle θ required to shine the laser a distance $\pm x$ from the origin of the laser turret can be calculated using

$$\theta = \arctan\left(\frac{\pm x}{z}\right) \tag{8}$$

where z is the distance between the turret and the mosquito plane.

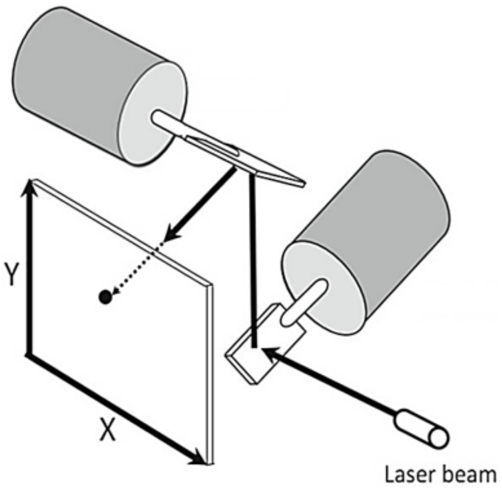


Figure 7.
Two-axis laser scanner schematic. This figure was modified from [7].

The specific geometry of the laser turret was designed with the goal to practically obtain a sufficiently small lateral step size of the laser on the mosquito plane, while maintaining sufficient speed.

a. Lateral step size and speed

The lateral step size of the laser must be on the order of 2 mm (size of mosquito plus size of laser).

The required lateral speed is determined in accordance with the field specifications the longest length of the mosquito enclosure is 1 m. To ensure that the laser could illuminate the set point within 2 seconds of receiving a step input it was assumed that the laser must be able to move with a velocity v of least 1 m/s opposed to 0.5 m/s required to move the 1 m length of the mosquito enclosure within 2 seconds. This was done to accommodate for the settling time

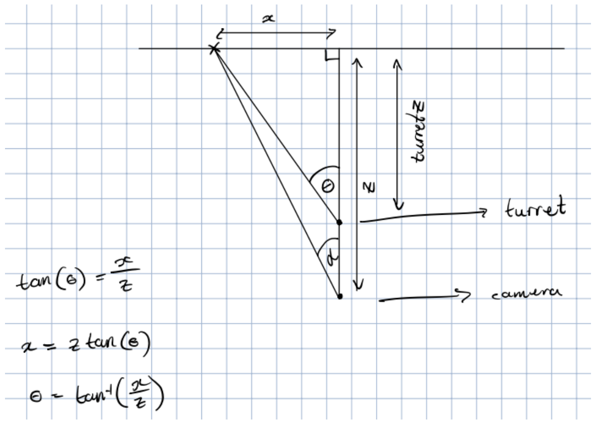


Figure 8.
Mirror angle calculation.

of the control system. This is also faster than the top speed of a mosquito which is 0.67 m/s according to SOURCE.

b. Lateral to angular step size

In the design of the laser turret there were multiple dimensions that were dependent on each other, it is up to the designer to choose certain practical dimensions from which the rest of the dimensions can be calculated. In this design the dimensions of the mirrors were chosen for practicality as 30×15 mm. These dimensions will be used throughout the rest of the design.

The turret axis on which the laser beam is first incident on will be referred to as the first axis and the other axis will be referred to as the second axis for the remainder of this discussion. The rotation range of the first axis is bounded by the geometry of the turret since the laser beam reflected from the first axis must be incident on the second axis. With the chosen mirror dimensions, the maximum angle through which the first axis can rotate from the origin (45° point) is $\theta_{max} = \pm 20°$ (add conservatively estimated) resulting in a 40° range of motion. This was determined geometrically in Figure 9. The lines representing the mirrors in Figure 9 are 30 mm long and the whole figure is drawn to scale. The rotation range of the second axis is not bounded by the geometry of the turret since the laser beam reflected from the second axis is incident on the mosquito plane. Therefore, the turret was oriented such that the first axis moves the laser beam parallel to the shorter width of the mosquito enclosure and the second axis moves the laser beam parallel to the longer length of the mosquito enclosure.

To maximise the angle through which the turret must rotate to move the laser beam across the height of the mosquito enclosure the minimum distance z_{min} between the turret and the mosquito plane was determined by rearranging Equation 8 to give

$$z = \frac{\pm x}{\tan\theta}. \tag{9}$$

where $x = 38$ cm is the height of the mosquito enclosure. The minimum distance z_{min} was calculated by substituting $\theta_m ax = 20°$ into Equation 9 to give

$$z_{min} = \frac{38\text{cm}}{\tan 20°} = 104.4.\text{cm}. \tag{10}$$

The required motor step resolution was calculated by substituting x_{min} and z_{min} into Equation 8 to give

$$\arctan\left(\frac{x_{min}}{z_{min}}\right) = \arctan\left(\frac{2\text{mm}}{104.4\text{cm}}\right) = 0.11^\circ \tag{11}$$

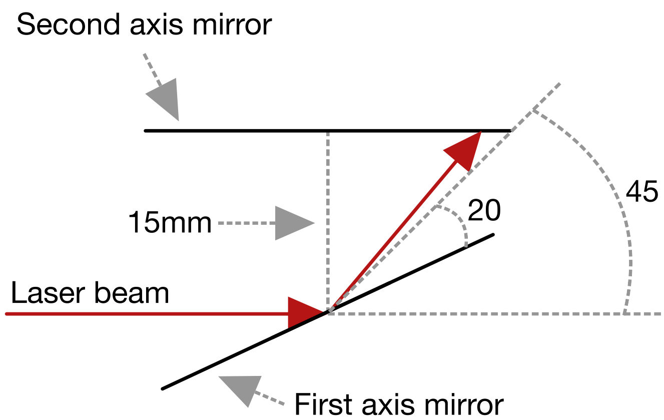


Figure 9.
Rotation range of first axis with chosen mirror dimensions.

c. Lateral speed to angular speed

From paragraph b it is known that 0.37999

$$\begin{aligned} &0.37999\,\text{m}\,\text{s}^{-1} = 40^\circ\,\text{s}^{-1} \\ \Longrightarrow\ &0.37999 \times \frac{1}{0.37999} = 40 \times \frac{1}{0.37999} \\ \Longrightarrow\ &1\,\text{m}\,\text{s}^{-1} = 105.26978\,\text{s}^{-1} \end{aligned} \tag{12}$$

The angular speed required to move the laser beam at a lateral speed $v = 1\,\text{m}\,\text{s}^{-1}$ was calculated with θ_{max} determined in paragraph b using

$$\omega = v \times \theta_{max}. \tag{13}$$

3.5.3 Motor requirements

~~The motor requirements were determined based on system requirement 4 (The laser must be able to illuminate the set point within 2 seconds accurate to within 1 mm.).~~

The need for precise position control of the laser meant that only stepper and servo motors were considered. The required torque and required revolutions per minute (RPM) was calculated to enable appropriate motor selection. ~~and the resulting laser step size on the mosquito plane were calculated.~~ In the calculations only a single axis of the turret was considered since the motors and mirrors for both axes will be identical and operate independently.

The required torque τ was calculated using

$$\tau = I\alpha \tag{14}$$

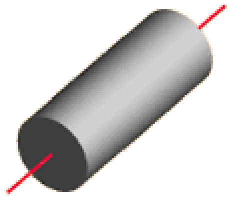


Figure 10.
Central axis of a cylinder.

where I is the moment of inertia and α is the angular acceleration.

The moment of inertia for the mirror was calculated by assuming the mirror was solid cylinder with diameter d equal to the width of the mirror. This will produce an inflated moment of inertia, however, this is not a concern since having extra torque is not a problem. The moment of inertia I of a cylinder with rotation about its central axis as seen in Figure 10 is given by

$$I = \frac{1}{2}Mr^2 \tag{15}$$

where M is the mass of the cylinder and $r = {}^{d}\!/\!_{2}$ is the radius of the cylinder. The mass of the cylinder M was calculated using the density of glass $\rho = 2500\text{kg/m}^3$ and the volume of the cylinder V given by

$$V = \pi r^2 h. \tag{16}$$

The height h and radius r of the cylinder was determined by assuming practical dimensions for the mirror. The dimensions of the mirror was assumed to be 30 mm by 15 mm. Thus, the mass of the mirror M is given by

$$M = \rho V = 2500\text{kg/m}^3 \times \pi\left(\frac{0.015}{2}\text{m}\right)^2 0.03\text{m} = 0.01325\text{kg} = 13.25\text{g}. \tag{17}$$

Given the above assumptions, the moment of inertia I was calculated as

$$I = \frac{1}{2}Mr^2 = \frac{1}{2} \times 0.01325\text{kg} \times \left(\frac{0.015}{2}\text{m}\right)^2 = 1.172 \times 10^{-7}\text{kg}\cdot\text{m}^2. \tag{18}$$

The angular acceleration of the motor α is calculated by

$$\alpha = \frac{\Delta\omega}{\Delta t} \tag{19}$$

where $\Delta\omega = \omega_f - \omega_i$ is the change in angular velocity and Δt is the change in time. This velocity was translated into the angular velocity ω_f using

$$\omega = v \times \theta_{max} \tag{20}$$

where θ_{max} is the angle through which the mirror must rotate to produce a 1 m lateral displacement of the laser on the mosquito plane.

The change in time Δt was determined by assuming the laser must be able to accelerate from a stand still to 1 m/s extremely rapidly to ensure that the motors could respond to the irregular flight pattern associated with a mosquito. It was assumed that this acceleration must occur within 10 ms.

Thus, the required angular acceleration α is given by

$$\alpha = \frac{\Delta\omega}{\Delta t} = \frac{0.698\text{rad/s}}{0.01\text{s}} = 69.8\text{rad/s}^2. \tag{21}$$

and thus the required torque τ is given by

$$\tau = I\alpha = 1.172 \times 10^{-7}\text{kg}\cdot\text{m}^2 \times 69.8\text{rad/s}^2 = 8.17 \times 10^{-6}\text{N}\cdot\text{m}. \tag{22}$$

3.6 Hardware implementation

general purpose input/output (GPIO) interfacing with stepper motor driver.

3.7 Software design

Mapping between pixels and real distance. Mosquito detection and tracking. Laser detection and tracking. proportional-integral-derivative (PID) controller. Feedback for turret. Distinguishing between laser reflections.

3.7.1 Mosquito and laser detection

It was assumed that the mosquitoes would be the only dark blobs in the enclosure and that the laser and its reflections would be the only bright blobs in the enclosure. The video frame would be cropped such that only the white back wall of the enclosure is visible.

The detection system was designed to work with a red laser because the computational complexity could then be decreased by only considering the red channel of the red, green, blue (RGB) frame.

The mosquito detection was designed to detect dark blobs in the enclosure and would not be able to distinguish between mosquitoes and other dark blobs. The laser detection was designed to detect bright blobs in the enclosure and would not be able to distinguish between the laser and its reflections. The mosquito and laser detection processes are similar in nature. The basic detection process flow is shown in Figure 11.

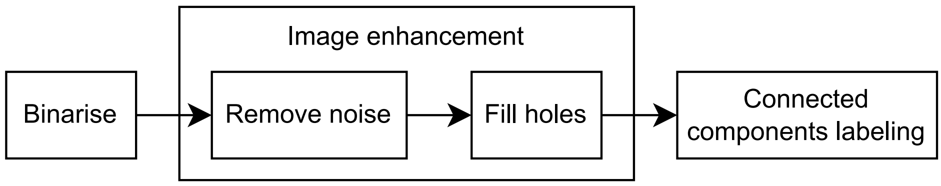


Figure 11.
Detection process flow.

The red channel of the RGB frame will be extracted and a copy would be made such the image processing operations can be performed independently to detect both the mosquitoes and the laser.

a. Binarisation

The only difference between the mosquito and laser detection is the binarisation step. The red channel will be binarised using a threshold value. The binarisation will be using a greater than threshold value for the laser detection and a less than threshold value for the mosquito detection. The ideal result of the binarisation is a binary image with the laser and its reflections as foreground pixels and the background as background pixels, likewise for the mosquitoes. This will not be the case since the foreground pixels can not be perfectly isolated using a threshold value and frame will contain noise.

b. Image enhancement

The respective binarised frames are subject to noise, holes, and erroneously joint or disjointed sections. This can be resolved using morphological operations. Morphological operations work by passing a structuring element over the pixels in an image and performing a logical operation on the pixels in the image that are covered by the structuring element.

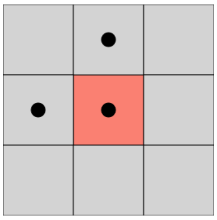
The noise and erroneously joint sections are removed using the morphological opening operation. Opening has been formally defined in subsubsection 3.3.3. The opening of A by B is the union of all translations of B that are completely contained in A as illustrated in Figure 4a. The holes and erroneously disjointed sections are removed using the morphological closing operation. Closing has been formally defined subsubsection 3.3.3. The closing of A by B is the complement of the union of all translations of B that do not overlap A as illustrated in Figure 4b.

The goal of the morphological operations is to ensure that all the foreground pixels are connected and that there are no erroneously connected foreground pixels such that the connected sections can be identified. The connectivity of pixels will be explained in paragraph c. The structuring element B will be tuned to obtain the best results.

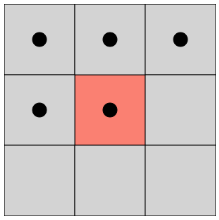
c. Connected components labelling

Once the processed binary image has been obtained the co-ordinates of the blobs need to be identified. This is done using connected components labelling (CCL). CCL is the process of assigning a label to each foreground pixel in a binary image such that connected pixels have the same label. The connectivity of pixels are defined using either four or eight connectivity. Four connectivity defines the neighbourhood of a pixel as the pixels to the left, right, top, and bottom of the pixel. Eight connectivity defines the neighbourhood of a pixel as the pixels to the left, right, top, bottom, top left, top right, bottom left, and bottom right of the pixel. The CCL connectivity is shown in Figure 12.

Various algorithms exist to perform CCL. Two of the most commonly used algorithms are the two-pass algorithm and the single-pass algorithm. The two-pass algorithm is generally faster for images with large connected components and the single-pass algorithm is generally faster for images with small connected components. The nature of the mosquito and laser detection is such that the connected components will be small, thus the single-pass algorithm was chosen. The single-pass algorithm is illustrated in Algorithm 1.



(a)
Four connectivity.



(b)
Eight connectivity.

Figure 12.
Connected components labelling connectivity.

Algorithm 1 Single-Pass Connected Component Labeling

```
procedure SINGLEPASSCCL(image)
    rows, cols ← dimensions of image
    label ← 1
    labels ← 2D array of zeros of size rows × cols
    for y ← 0 to rows − 1 do
        for x ← 0 to cols − 1 do
            if image[y][x] = 1 and labels[y][x] = 0 then
                ExploreBlob(image, labels, x, y, label)
                label ← label + 1
            end if
        end for
    end for
end procedure

procedure EXPLOREBLOB(image, labels, x, y, label)
    queue ← empty queue
    Enqueue(queue, (x, y))
    while not empty(queue) do
        (x, y) ← Dequeue(queue)
        if x < 0 or x ≥ cols or y < 0 or y ≥ rows or image[y][x] = 0 or labels[y][x] > 0 then
            continue
        end if
        labels[y][x] ← label
        for each neighbor (nx, ny) of (x, y) do
            Enqueue(queue, (nx, ny))
        end for
    end while
end procedure
```

3.7.2 Mosquito tracking

The mosquito tracking was performed using the simple online and realtime tracking (SORT) algorithm. The SORT algorithm is based on the tracking-by-detection paradigm, which means it relies on an external object detector. The object detector used was the mosquito detection algorithm described in **??**. The SORT algorithm associates the detections across frames to create tracks.

a. Tracks

The mosquito tracks are created using the Kalman filter. The theoretical background of Kalman filter is discussed in subsubsection 3.3.4. The kinematic equation representing the flight of a mosquito must be defined to determine the state space of the Kalman filter. The kinematic equation used to model the flight of a mosquito is

$$\mathbf{x_k} = \mathbf{x_{k-1}} + \dot{\mathbf{x}}_{\mathbf{k-1}}\Delta t + \frac{1}{2}\ddot{\mathbf{x}}_{\mathbf{k-1}}\Delta t^2, \tag{23}$$

where $\mathbf{x_k}$ is the position vector defined by the 2D pixel co-ordinate of a mosquito $\begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}}$. A constant velocity model with acceleration as noise will be used since mosquitoes have an erratic flight pattern. Therefore, the set of differential equations describing the state space of the Kalman filter is

$$\begin{gathered} \mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \\ \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta t \\ y_{k-1} + \dot{y}_{k-1}\Delta t \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix}. \end{gathered} \tag{24}$$

The state is $\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^{\mathrm{T}}$ and the state transition matrix is

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{25}$$

The position of a mosquito is the only component of the state the is measured by the detection system. Thus, the observation matrix H is given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{26}$$

The error in the constant velocity model of The process noise covariance matrix Q is given by

$$Q = \begin{bmatrix} \sigma_x^2 & 0 & \sigma_x\sigma_{\dot{x}} & 0 \\ 0 & \sigma_y^2 & 0 & \sigma_y\sigma_{\dot{y}} \\ \sigma_{\dot{x}}\sigma_x & 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & \sigma_{\dot{y}}\sigma_y & 0 & \sigma_{\dot{y}}^2 \end{bmatrix} \tag{27}$$

where σ_x and $\sigma_{\dot{x}}$ are the standard deviations of the position and velocity, respectively. The standard deviation of the position is defined as the standard deviation of the acceleration σ_a

multiplied by $^1/_2\Delta t^2$ since this is the effect that the acceleration will have on the position as shown in

The measurement noise covariance matrix R is given by

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{28}$$

where σ_x^2 and σ_y^2 are the variances of the measurement noise. The initial state covariance matrix P is given by

$$P = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 \end{bmatrix} \tag{29}$$

where σ_x^2, σ_y^2, $\sigma_{\dot{x}}^2$, and $\sigma_{\dot{y}}^2$ are the variances of the initial state. The variances of the process noise and the measurement noise were tuned to obtain the best results.

b. Association

3.8 Software implementation and optimisation

Only using red channel, graphics processing unit (GPU), low resolution, etc.

3.9 Final system integration and testing

Real-time multi threading.

4. Results

4.1 Summary of results achieved

4.2 Qualification tests

5. Discussion

5.1 Critical evaluation of the design

5.1.1 Interpretation of results

5.1.2 Critical evaluation

5.1.3 Unsolved problems

5.1.4 Strong points of the design

5.1.5 Expected failure conditions

5.2 Considerations in the design

5.2.1 Ergonomics

5.2.2 Health and safety

5.2.3 Environmental impact

5.2.4 Social and legal impact

5.2.5 Ethics clearance

6. Conclusion

6.1 Summary of the work completed

6.2 Summary of the observations and findings

6.3 Contribution

6.4 Future work

7. References

[1] W. H. Organisation. (2020, 12) The top 10 causes of death. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

[2] P. Hurtik, D. Číž, O. Kaláb, D. Musiolek, P. Kočáek, and M. Tomis, "Software for visual insect tracking based on f-transform pattern matching," in IEEE Second International Conference on Data Stream Mining & Processing, Lviv, Ukraine, 2018.

[3] Z. Khan, T. Balch, and F. Dellaert, "Efficient particle filter-based tracking of multiple interacting targets using an mrf-based motion model," in Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, 2003.

[4] W. Liang, H. Wang, and H. Krim, "A behaviour-based evalutation of product quality," in International Conference on Acoustics, Speech, and Signal Processing, 2016.

[5] Y. Bao and H. Krim, "Video tracking of insect flight path: Towards behavioral assessment," in IEEE, 2018.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online realtime tracking," Queensland University of Technology, University of Sydney, Tech. Rep., 2017.

[7] "Polarization effects of galvos in SLA 3D printing," https://henryquach.org/galvopolarization.html, accessed: 2023-10-14.

Part 4. Appendix: technical documentation

HARDWARE part of the project

Record 1. System block diagram

Record 2. Systems level description of the design

Record 3. Complete circuit diagrams and description

Record 4. Hardware acceptance test procedure

Record 5. User guide

SOFTWARE part of the project

Record 6. Software process flow diagrams

Record 7. Explanation of software modules

Record 8. Complete source code

Complete code has been submitted separately on the AMS.

Record 9. Software acceptance test procedure

Record 10. Software user guide

EXPERIMENTAL DATA

Record 11. Experimental data