

# Research Portfolio

Yuanshao Yang

## Contents

<b>1</b>	<b>Series Spring Design of Open-Source Leg</b>	<b>2</b>
1.1	Background . . . . .	2
1.2	Objectives . . . . .	2
1.3	Results . . . . .	2
<b>2</b>	<b>Software Generalization for Open-Source Leg</b>	<b>4</b>
2.1	Background . . . . .	4
2.2	Objectives . . . . .	4
2.3	Results . . . . .	4
<b>3</b>	<b>Motion Analysis &amp; Design of Robot Swimmer Model</b>	<b>6</b>
3.1	Background . . . . .	6
3.2	Objectives . . . . .	6
3.3	Results . . . . .	6
<b>4</b>	<b>Classification Model for Heart Disease Diagnosis</b>	<b>8</b>
4.1	Background . . . . .	8
4.2	Objectives . . . . .	8
4.3	Results . . . . .	8
<b>5</b>	<b>Automated Vehicle with Tracking System</b>	<b>10</b>
5.1	Background . . . . .	10
5.2	Objectives . . . . .	10
5.3	Results . . . . .	10

# 1 Series Spring Design of Open-Source Leg

## 1.1 Background

Recent researches have proved the importance of torsional springs in Series Elastic Actuator (SEA) designs, for their advantage in higher specific energy and energy density, while serving as essential elements in energy storage and force / torque measurement. Such springs are expected to serve for key functions in prosthesis to properly provide net-positive mechanical energy. However, difficulties in design tradeoffs, such as balancing the advantages of better low-stiffness performances and the drawbacks of reduced material strength, often impedes from properly functioning.

Therefore, the strategy of Series Torsional Spring is put forward to address the issue. Such design will enable the design of energy storage element in a relatively light-weight, compact form by series combinations of spring elements, which further spans the design space for SEAs, and empower further applications in prosthesis / exoskeleton designs.

## 1.2 Objectives

1. Generating mechanical design of series spring with easier mounting procedures.
2. Evaluate the backlash effect on different kinds of interfaces (e.g. interface between Cam Shaft and Spring Flexures)
3. Validate the effect on spanning the design space from series spring design.

## 1.3 Results

- Computer Aided Design (CAD) of Series Torsional Spring, shown in Figure 1.

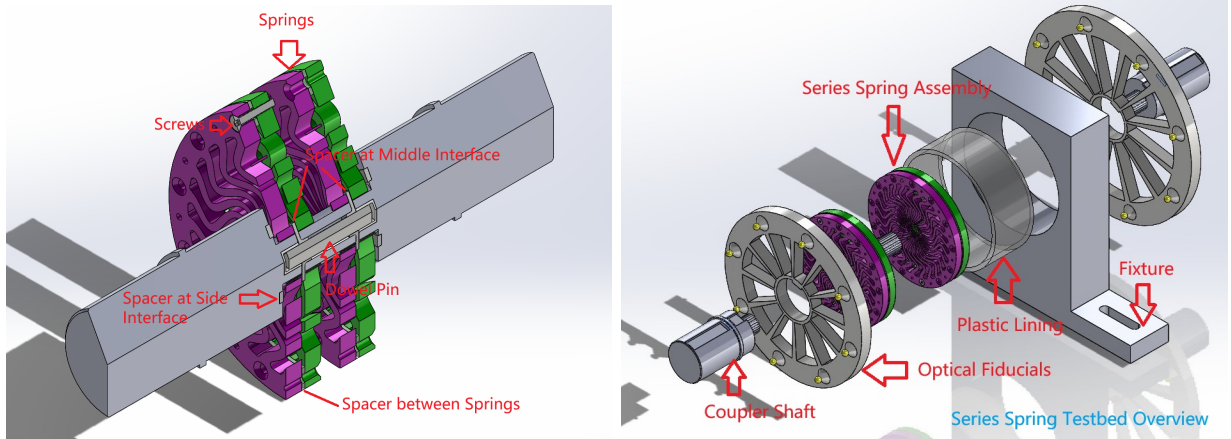


Figure 1: CAD Design of Series Torsional Spring

- Measurement strategies and Noise Evaluation of flexure deflection of spring by computer vision, including masking, blurring & grayscaling, and Hough Circle Transformation to recognize fiducial markers, as in Figure 2. The center of the fiducial ring is determined by the triangular method.

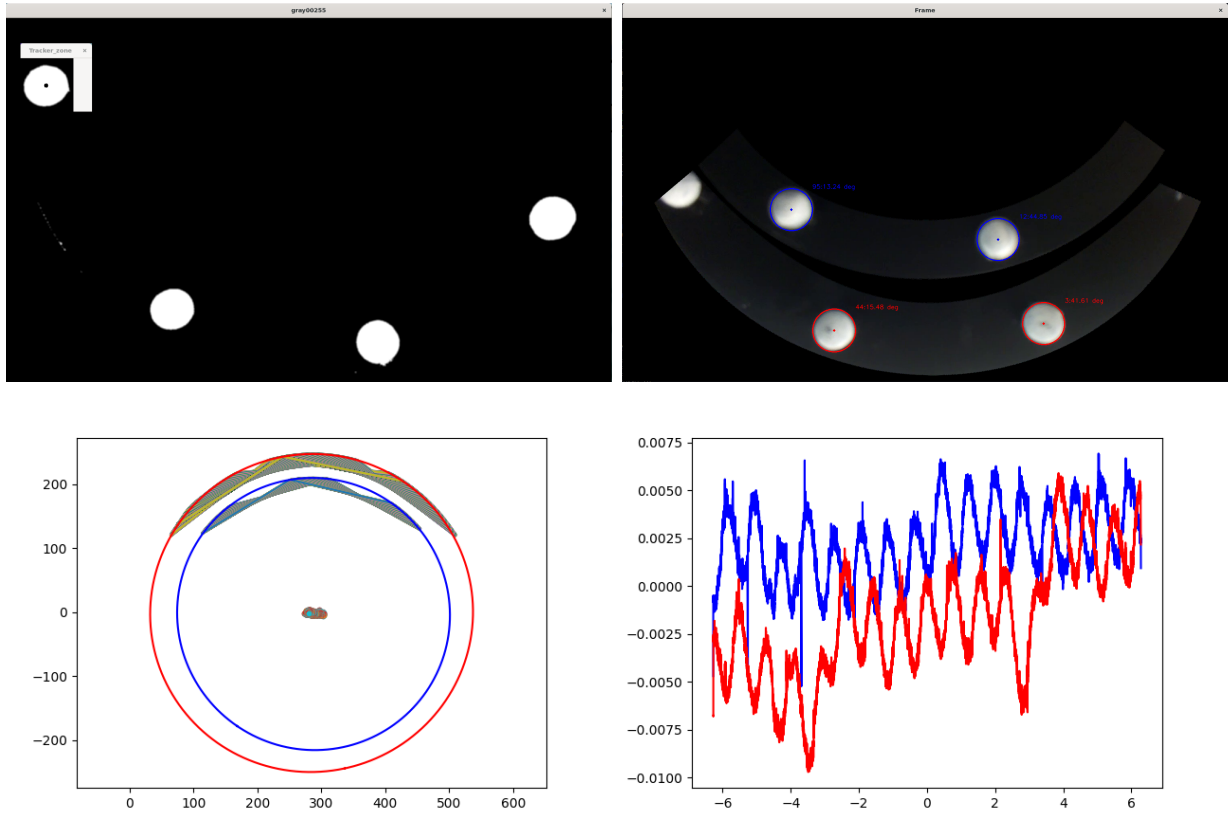


Figure 2: Fiducial Recognition and Deflection Angle Calculation

- Evaluation of the measured stiffness of the series spring, as shown in Figure 3.

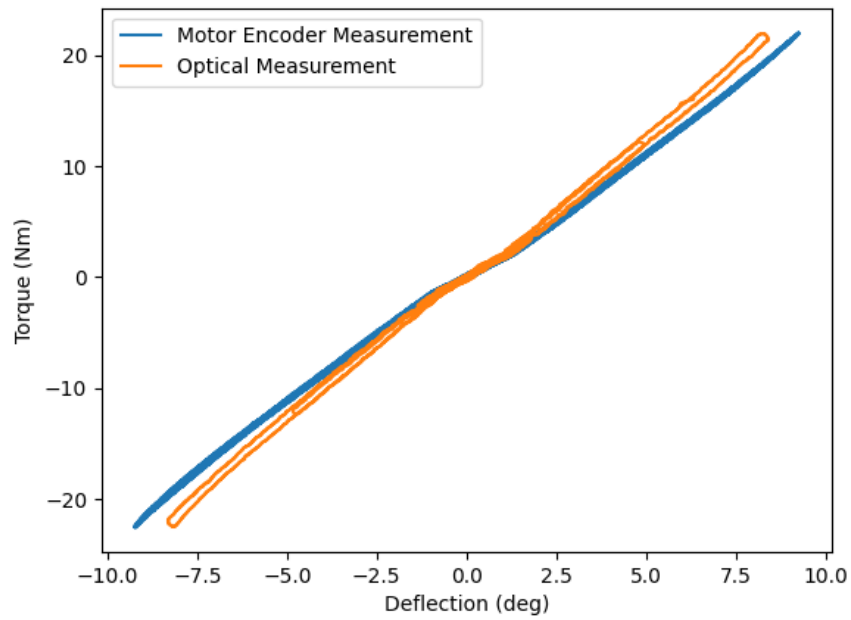


Figure 3: Fiducial Recognition and Deflection Angle Calculation

## 2 Software Generalization for Open-Source Leg

### 2.1 Background

The Open-Source Leg project is a standardized hardware and software platform for prosthesis designs, controls, and tests, aimed to eliminate the barriers among different prosthesis researches and serve as a bridge for collaborative efforts in prosthetic leg design and control. Such platform is expected to be flexible towards hardware choices. However, the previous versions of Open-Source Leg software library is biased to a specific hardware choice, which fails to close the gap between individual researches and adds difficulty in comparing experiment results.

Thus, a generalized, user-friendly version of the software library is needed to serve as a basis for alternative prosthetic leg forks. Such library will allow other researchers to create their version of prosthetic leg with lower costs and more standardized features, in order to offer a common platform to test and evaluate further mechanical designs and control strategies.

### 2.2 Objectives

1. Create a base library for Open-Source Leg actuator and sensor modules, to serve as a template for standardized development, with [merged into GitHub Repo](#).
2. Redevelop the instance of actuator modules based on the generalized library design.
3. Test and compare the performance of different actuators and sensors based on the generalized library design.

### 2.3 Results

- A base library for Open-Source Leg actuator and sensor modules, as Figure 4, and examples of a redeveloped actuator module.

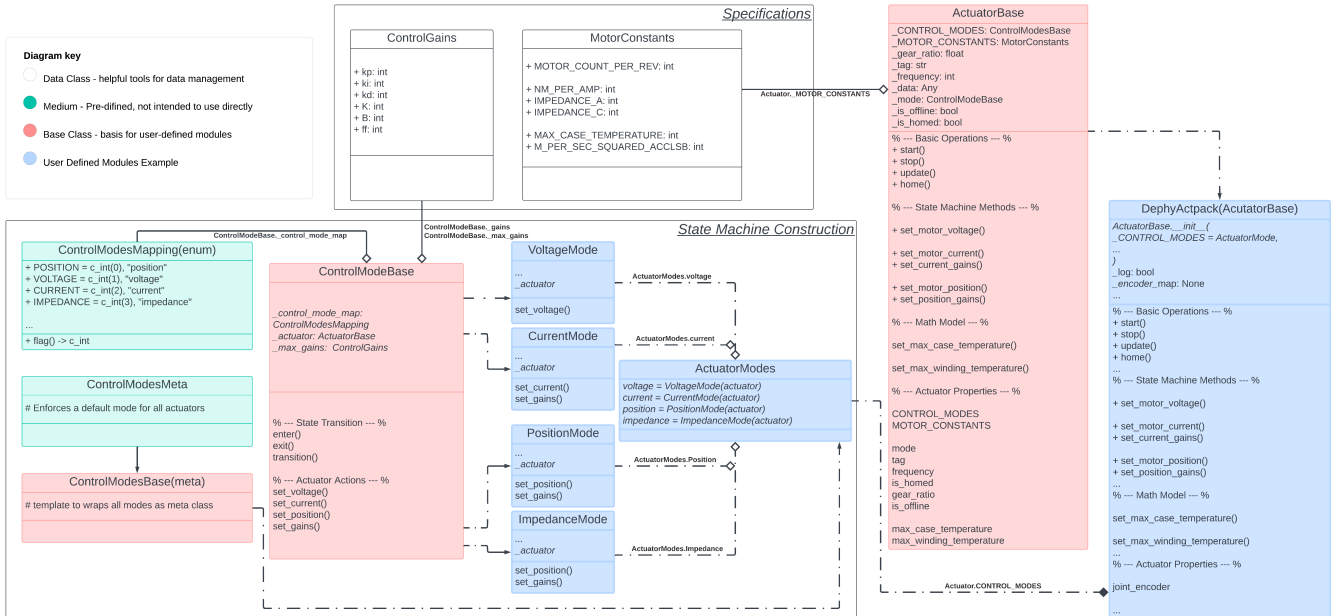


Figure 4: Class Diagram of Generalized Python Library

- Performance evaluation across different actuator choices, as Figure 5

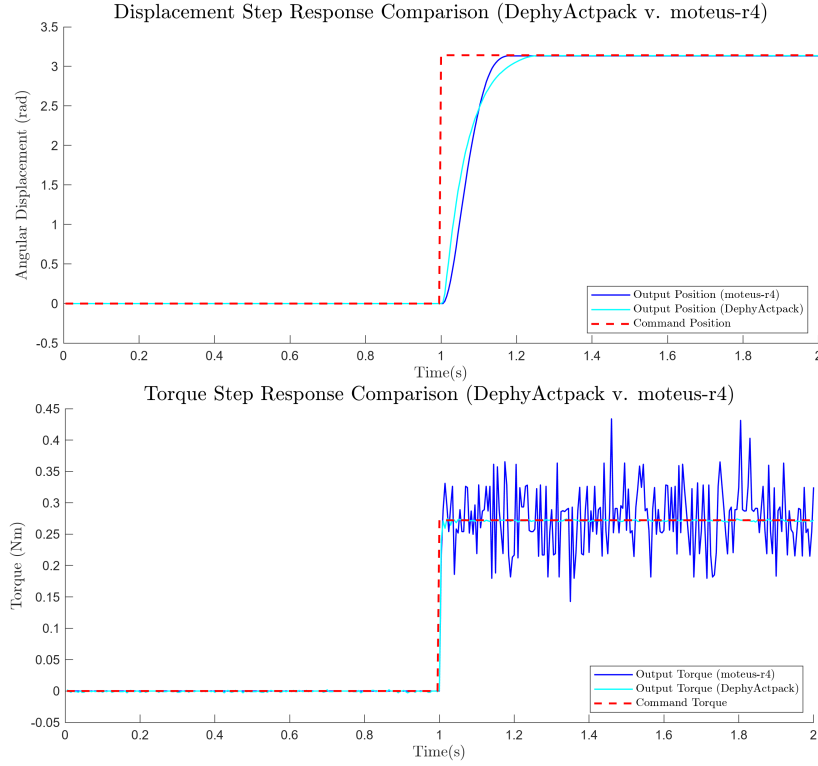


Figure 5: Performance Evaluation & Comparison of Different Actuators on Generalized Library

## 3 Motion Analysis & Design of Robot Swimmer Model

### 3.1 Background

Targeting transport is an emerging field in the medical industry that is attempting to obtain a more precise cure for the diseased area in the human body. Although attempts such as targeted therapy are performed, they still inherits the same problem that both the location and the amount of optimal dosing are hard to be controlled.

Therefore, a complete robot swimmer model should be developed to perform the task, due to its better performance than chemical-based therapies. Such design aims to offer a feasible solution for future research in control strategies for robot swimmers, in order to achieve better performances in targeting transportation.

### 3.2 Objectives

1. Develop a mechanical model with sphere head and flagella (as a mimic of bacteria with dual flagella), and the corresponding kinematics model, that is suitable to perform object tracking.
2. Evaluate the design & control strategies from computational simulations.

### 3.3 Results

1. Mechanical Design of the Components of Robot Swimmer, with Computational Fluid Dynamics (CFD) analysis to validate the design, as shown in Figure 6.

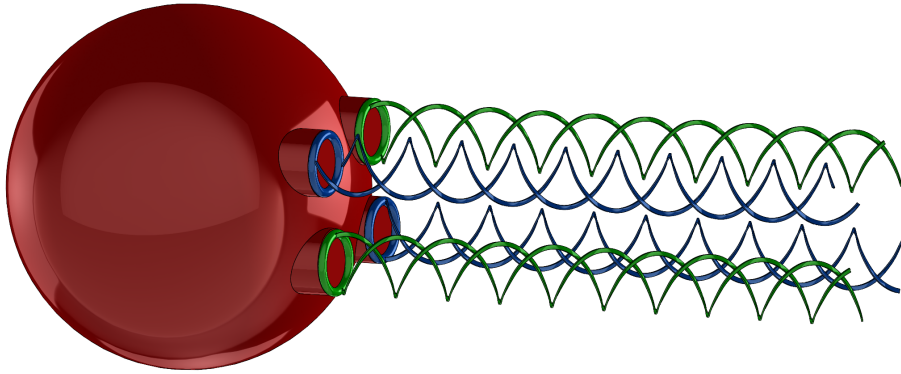


Figure 6: Mechanical Design of Robot Swimmer

2. Kinematics Analysis of the Robot Swimmer Model, as shown in Figure 7.

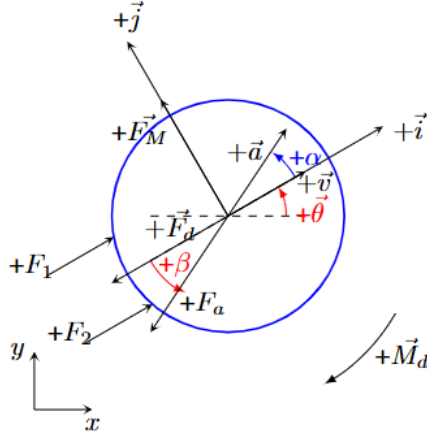


Figure 7: Free Body Diagram of Robot Swimmer Model

- Control strategies applied to the Robot Swimmer Model (Feedback Control & Feedforward Control, as in Figure 8), with performance evaluation in simulations, as shown in Figure 9.

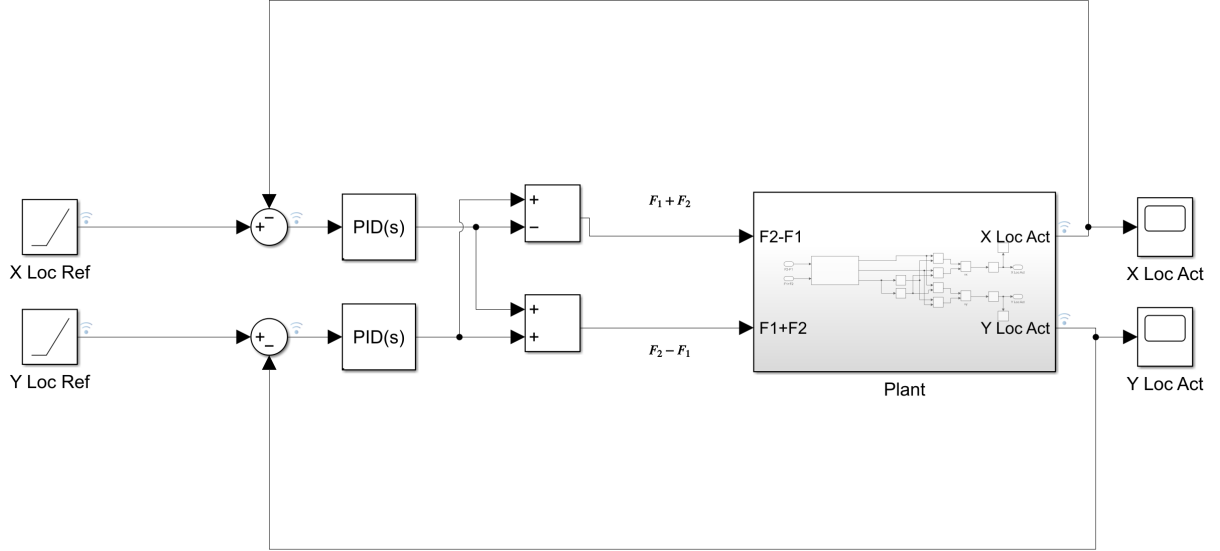


Figure 8: Modelling of Feedback Controller as Block Diagram

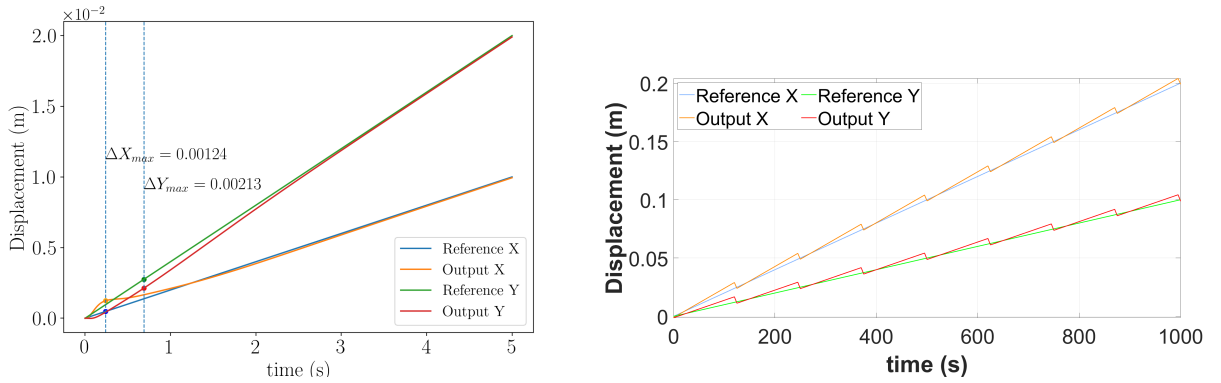


Figure 9: Performance Evaluation of Tracking Behavior Under Different Control Strategies

## 4 Classification Model for Heart Disease Diagnosis

### 4.1 Background

Due to the increase probability to get heart diseases, the hospital is experiencing more and more pressure from the cardiovascular department. The idea to develop a classification model comes up correspondingly, to reduce the heavy burden from . Such model is expected to classify different kinds of heart diseases, predict the probability of having heart diseases when given the specifications of one's health status, and

### 4.2 Objectives

1. Work on diagnostic data of cardiac patients through feature selection
2. Develop regression models (e.g., Random Forest) via Scikit-Learn and PyTorch in Python and use training data sets of clinical diagnostic data
3. Test model performance on internal datasets
4. Perform interpretability analysis of built models using variable sensitivity analysis
5. Enhance classification processes' precision of the model by analyzing visualized Decision Trees

### 4.3 Results

- A random forest classification model to classify and predict heart diseases, as shown in 10. Model posted as [GitHub Repo](#).

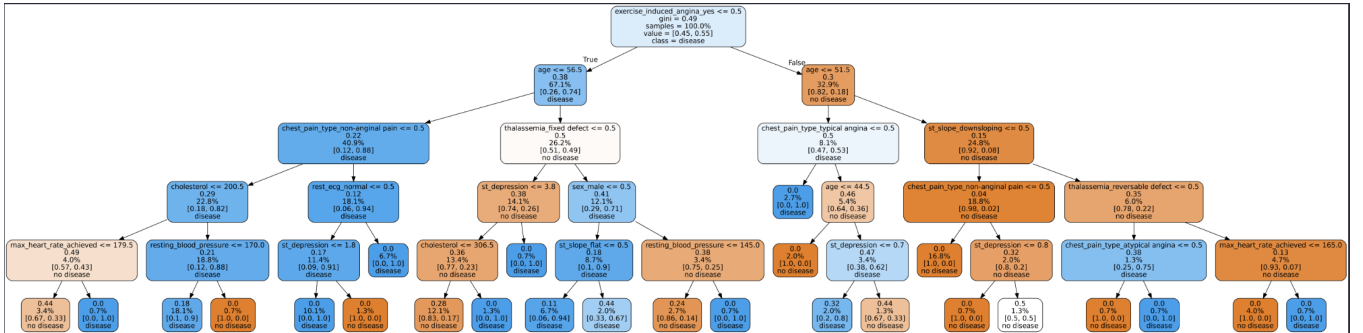


Figure 10: Decision Tree of Random Forest Model

- Evaluation of the prediction success rate, shown in Figure 11



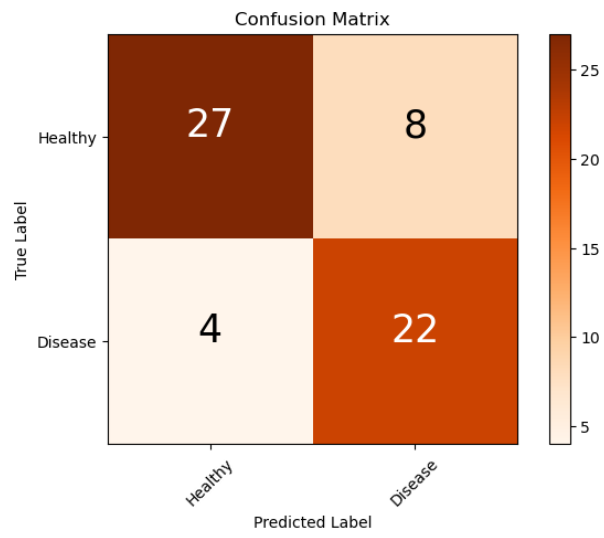


Figure 11: Prediction Result of Classification Model by Confusion Matrix

## 5 Automated Vehicle with Tracking System

### 5.1 Background

### 5.2 Objectives

1. Design and control of the gimbal stabilizer, which processes the estimation of distance and deviation angle from the camera module.
2. Build communication portal among the gimbal stabilizer, the the camera module, and the central controller for cruise control.

### 5.3 Results

- A gimbal stabilizer to stabilize the camera module and performs object tracking, as shown in Figure 12.

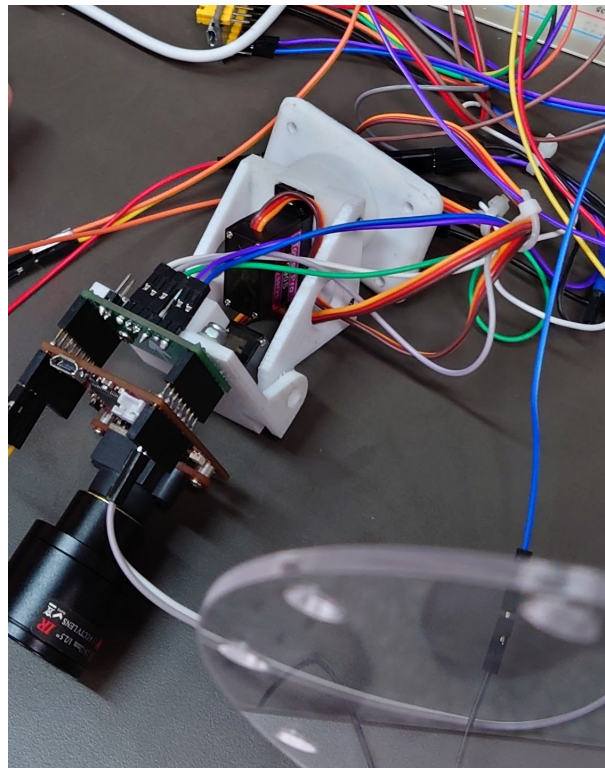


Figure 12: Gimbal Stabilizer

- Control strategies applied to the gimbal stabilizer, [GitHub Repo](#)