**FLUTTER ANALYSIS**

**Variable used:**

**cap** 🡪 To read the input video

**fps** 🡪 To store the frames per second of the video

**arc** 🡪 To store the angular distance travelled by the wing.

**maxAngle1, maxAngle2** 🡪 maximum angles of flutter above and below the horizontal respectively.

**c 🡪** Counts number of frames when a red object is visible.

**d 🡪** Counter to specify if contour is above or below horizontal.

**f 🡪** Oscillations of the wing.

**frame 🡪** Read and store the frame.

**red 🡪** Frame with only the red objects displayed.

**contour 🡪** Stores a particular shape in the frame.

**M 🡪** Stores the moments of the contour.

**cx, cy 🡪** Store the center of the contour.

**angle 🡪** Stores the angle of the center of contour with the horizontal.

**Procedure:**

* **Lines 1 – 3:** Importing the necessary libraries.
* **Line 4:** Reading the video file and storing in variable cap.
* **Line 5 – 6:** Finding and printing the fps of the video feed.
* **Line 7:** Initializing variables to store sum of distance travelled both above and below horizontal.
* **Line 8:** Specifying the font in which to display the angle in the output frame.
* **Line 9:** Initializing the counters.
* **Line 11:** Reading a frame from the input video and storing in “frame”.
* **Line 13 - 18:** Thresholding the frame to show only objects of red color.

*[160, 87,111] – [180, 255, 255] is the HSV range of the color red.*

* **Line 19:** Finding the contours of only red objects in each frame and storing in “contour”.
* **Line 20:** Check if the length of contour is greater than 50.
* **Line 21:** Describe the moments of “contour” and store in “M”
* **Line 23:** Find the center of “contour” and store in cx, cy.
* **Line 24:** Draw a line joining a point in “frame” to the center of “contour”
* **Line 25 - 26:** Calculate the angle of the line with the horizontal and display it.
* **Line 27:** Check if angle >0 and d%2 != 0
* **Line 28 - 29:** Check if angle>maxAngle1 then maxAngle1=angle
* **Line 30 – 35:** d+=1, f+=0.25, add the angular distance to arc and print maxAngle1 and break the for loop

*f increased by 0.25 because each path above or below the horizontal axis contributes to 1/4th of the flutter oscillation.*

* **Line 37:** Initialize maxAngle2 to 0
* **Line 38:** Check if angle<0and d%2 == 0
* **Line 40-41:** Check if angle>maxAngle2 then maxAngle2=angle
* **Line 42-47:** d+=1,f+=0.25, add the angular distance to arc and print maxAngle2 and break the for loop

*f increased by 0.25 because each path above or below the horizontal axis contributes to 1/4th of the flutter oscillation*

* **Line 48:** Initialize maxAngle1 to 0
* **Line 49:** Display the frame
* **Line 52 to 55:** Display the values of flutter speed and oscillations per second and release the capture.