Track check in Helicopter using Image processing

Mansi (2020ME21031)

Rahul Saini (2020ME10031)

B.Tech Mechanical Engineering

Supervisor

Prof. Arpan Gupta

Department of Mechanical Engineering

Indian Institute of Technology, Delhi

September 2023

Contents

List of abbreviations **ii**

List of figures **iii**

1. Introduction

1.1. Background……………………………………………………1

1.2. Objectives…………………………………………………...2

2. Literature Review……………………………………………..2

3. Work Plan

3.1. Research Paper ……………………………………………...6

3.2. Defining Objective and Scope of the Project………………………..6

3.3. Data Preparation and Processing……………………………………6

3.4. Model Training and Performance Evaluation………………………6

4. Progress So far

4.1. Data Acquisition

4.1.1. Experimental Setup……………………………………..6

4.1.2. Introducing Faulty Conditions………………………………..7

4.1.3. Data Collection…………………………………………7

4.2. Blade Tracking……..………………………………………...……7

4.3. Blade Tip detection…………………………………………..7

4.3. Edge detection……………………………………...…………...7

4.3. Corner detection……………………………………………...7

4.3. Finding blade Tip positions……………………………………..7

4.3. Camera Calibration……………………………………………..7

5. Future Work

5.1. Model Training and Performance Parameters………………………7

6. References……………………………………………………..9

List of Abbreviations

List of Figures

1. Introduction

1.1 Background

Helicopters are widely used in modern aircraft for their ability to hover and perform vertical take-offs and landings. However, helicopter rides are often uncomfortable due to the significant levels of vibrations they produce. These vibrations can lead to structural issues and even accidents. There are two main types of vibrations in helicopters: vertical and lateral. Vertical vibrations, in particular, are caused by the rotor blades being out of alignment, as illustrated in **Figure 1**. This type of vibration makes the helicopter bounce up and down during flight. It occurs because one of the blades lifts the helicopter during one part of its rotation but loses lift in the remaining quadrants.

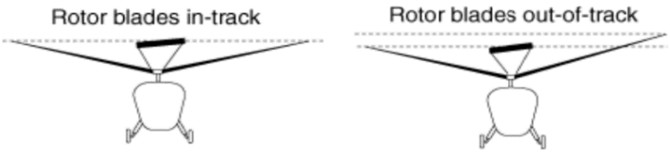

Figure 1. In-track and out of track blades

Blade tracking is a critical process used to manage vertical vibrations. It involves calculating and adjusting the vertical position of each blade's tip based on its position in the air. Typically, one blade is selected as a reference, and the position of the other blades' tips is measured relative to this reference blade. If the difference in blade tip position falls within a certain threshold (usually around 20 mm), the blades are considered to be in track; otherwise, they are deemed out of track or misaligned. This threshold may vary for different helicopter models. Blade tracking can be performed both in-flight and on the ground. Various methods have been employed in the past for blade tracking. The oldest method is known as flag tracking, where each blade is painted with a different coloured wet paint, and an operator holds a flag near the rotating blade tips. When the blades touch the flag, marks are left on it, and the out-of-track values are determined from these marks.

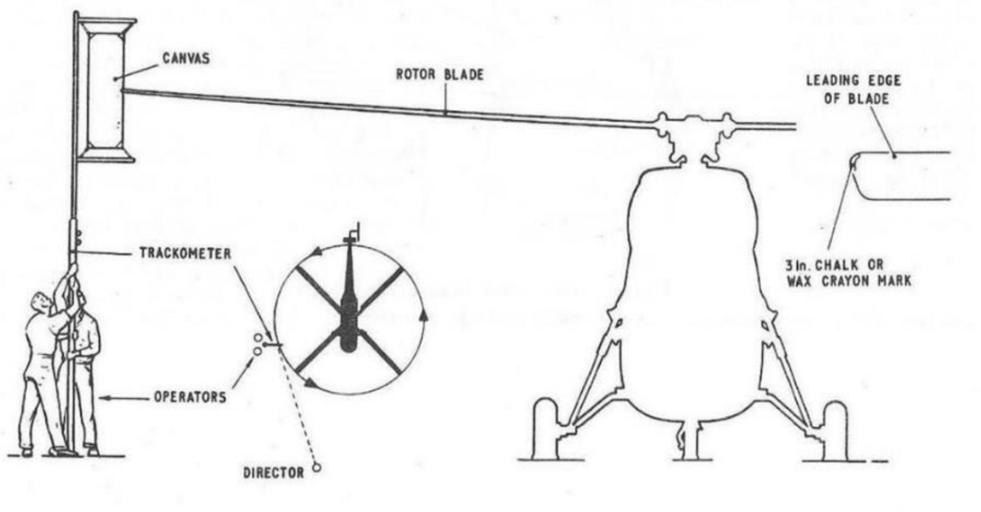

Figure 2. Track Balancing in Helicopters – Evolution in equipment

1.2 Motivation

The motivation for this study was to revolutionize the way industries maintain their critical machines. We aim to learn and apply data analytics and machine learning algorithms to predict the faults in industrial machinery before they even occur. This proactive approach improves equipment life, reduces unplanned downtime, identifies faults through continuous monitoring, and saves the company significant costs.

Furthermore, by reducing unnecessary maintenance and replacement of parts, we can minimize the waste generated and conserve resources, thus aiming at sustainable development.

1.3 Objectives

The objectives of this technique include:

1. Reducing vertical vibrations.
2. Designing a system suitable for indoor/outdoor use and in-flight/on-ground operation.
3. Developing a system that requires no modifications to the blades like attaching sensors to blade tips.
4. Creating an affordable system.

This approach utilizes image processing techniques and is divided into several steps, including image filtering, image thresholding, precise determination of each blade's position, and conversion of pixel distances into real-world measurements. This method aims to provide accurate results while overcoming the drawbacks associated with previous techniques.

2. Literature Review

2.1 Rotor Track and Balance (RTB) Techniques (Bechhoefer1 et al [1])

The study introduced the idea of rotor blade adjustments using weights (WTS), pitch control rods (PCR), and trailing edge tabs (TAB) to balance out inherent blade non-uniformities. Additionally, Ferrer (2001) laid the groundwork for later algorithmic developments in RTB by highlighting the linearity of adjustment coefficients. Multiple equivalent solutions are produced when the Fourier transform is applied to the time domain, necessitating the development of a method to determine actual blade adjustments. Initially, all efforts to reduce blade non-uniformity began with a desire to reduce track split errors. Since rotor track and balance's main objective is to reduce vibration, finding an effective solution to the issue is a driving force.

2.2 Use of the Hough Transformation To Detect Lines and Curves in Pictures (Duda et al [2])

2.3 A Novel Shi-Tomasi Corner Detection Algorithm Based on Progressive Probabilistic Hough Transform (Mu et al [3])

The Moravec algorithm is the foundation of Shi-Tomasi. The concept of Moravec corner detection is as follows: Make a detection window in the image. By moving the window slightly in all directions, the average power of the window is determined. When the energy change value exceeds the threshold value, the central pixel of the window is extracted as a corner point. Assuming that the gradation of the point (x, y) is f(x, y), suppose each pixel point (x, y) in the image is moved separately with the distance of (u, v), the gradation of pixel point can be represented by:

$$E_{u,v}(x, y) = \sum_{u,v} \omega_{u,v}[f(x + u, y + v) - f(x, y)]^2 \tag{1}$$

This method can only detect the intensity change of the window function moving in 8 basic directions, so the Moravec operator lacks rotation invariance. This means that if the target image is rotated by 15 degrees, the original detected corner point cannot be detected, and the point that was not the corner point may become a corner point, causing the method to fail to extract the stable corner point accurately. As a result, Shi-Tomasi algorithm detects corner points using differential operation and autocorrelation matrix. This paper also discusses the Hough transform, which is one of the fundamental methods for recognising geometric shapes from images in image processing.

Shi-Tomasi corner detection based on PPHT:

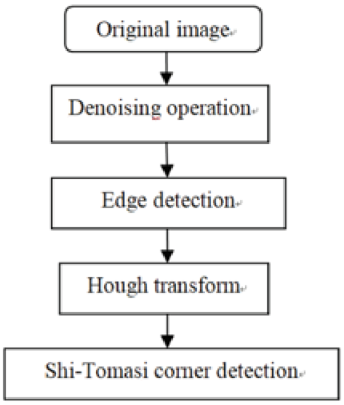

Fig.3 Algorithm flowchart

The algorithm is made up of the following components: First, the necessary preprocessing on the source image is performed, which includes denoising, edge detection, conversion, and grayscale image. The image is then subjected to the Hough line transformation, with the result marked on the image. Finally, the Shi-Tomasi operator is used to detect corners. Hough transform and Shi-Tomasi are two crucial steps in the algorithm. These two steps have a direct impact on the effect of the final corner detection.

The basic intuition is that corners can be detected by looking for significant changes in all directions. We consider a small window on the image and then scan the whole image, looking for corners. Shifting this small window in any direction will result in a large change in appearance if that particular window happens to be located on a corner. Fig. 4(a)

1) Flat regions will have no change in any direction. Fig. 4(b)
2) If there’s an edge, then there will be no major change along the edge direction. Fig. 4(c)

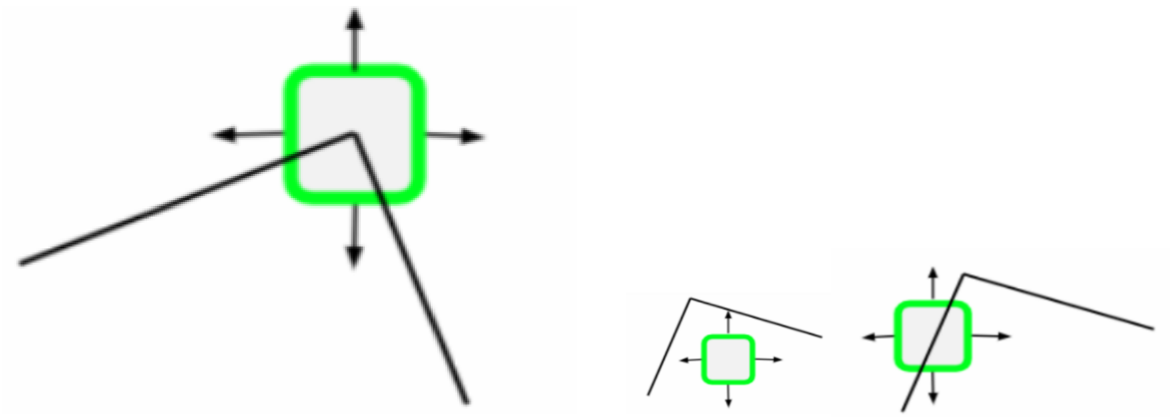

Fig. 4(a) Fig. 4(b) Fig. 4(c)

If we’re scanning the image with a window just as we would with a kernel and we notice that there is an area where there’s a major change no matter in what direction we actually scan, then we have a good intuition that there’s probably a corner there.

2.3 Camera Calibration (Open-cv documentation [4])

For effective use of a camera as a visual sensor, particularly for precise image analysis and 3D reconstruction, camera calibration is a crucial process. This complex procedure is based on precisely determining both internal and external camera parameters. These parameters include lens distortions, intrinsic aspects such as focal length and optical centres, and extrinsic aspects such as the camera's orientation and position relative to a defined world coordinate system.

2.4.1 Parameters:

1. Distortion: Cameras introduce distortion due to lens imperfections. The two main types are radial distortion (straight lines appear curved) and tangential distortion (image appears skewed). These distortions can significantly affect image analysis and 3D reconstruction accuracy.

2. Intrinsic Parameters: Intrinsic parameters are camera specific properties, including focal length (fx, fy) and optical centres (cx, cy). They form the camera matrix and help correct distortions. A common representation is the pinhole camera model.

3. Extrinsic Parameters: Extrinsic parameters involve rotation and translation vectors that position the camera’s coordinate system relative to a world coordinate system. These parameters are crucial for 3D scene reconstruction.

4. Calibration Pattern: A known pattern (e.g., chessboard) is placed in front of the camera. The pattern’s 3D coordinates, and corresponding 2D image coordinates are used to calibrate the camera.

5. Re-projection Error: After calibration, re-projected image points are compared with detected image points. Lower re-projection error indicates more accurate calibration.

2.4.2 Mathematical Formulations:

1. **Radial Distortion:** Radial distortion is approximated using a polynomial expression:

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2)$$

$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3)$$

2. **Tangential Distortion:** Tangential distortion is caused by lens misalignment:

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (4)$$

$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \quad (5)$$

3. Camera Matrix: The camera matrix combines intrinsic parameters and maps 3D points to 2D image coordinates:

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \quad (6)$$

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

2. Work Plan

3.1 Gantt Chart:

Figure 2. Gantt Chart

4. Progress So Far

4.1 BLADE TRACKING

This step involves creating an experimental setup that mimics the configuration of a helicopter. Subsequently, a program is developed to detect the degree of misalignment between the blades' tips.

4.2 BLADE TIP DETECTION

Blade tip detection and vertical tracking can often be achieved through traditional computer vision techniques and algorithms. An experimental investigation was carried out to determine the rotational speed, quantified as revolutions per minute (RPM), of a swiftly rotating fan. In order to facilitate this measurement, a distinctive red marker was applied to one of the fan blades. Subsequently, a Python model was meticulously developed to analyse video footage capturing the fan's movement. This model, included below, possesses the capability to not only scrutinize the video content but also accurately enumerate the complete rotations executed by the fan, thereby enabling precise RPM calculation. It should be noted that this Python script can be readily applied to ascertain the RPM of various rotating objects.

Fig. 5(a) SolidWorks fan with a red mark (b) Red mark on the real fan

1. **Image Processing Techniques:** Depending on the quality and characteristics of your helicopter video or images, you can apply image processing techniques such as edge detection, contour detection, and geometric analysis to locate the blade tip and track its vertical position.
2. **Thresholding:** You can use thresholding techniques to isolate the helicopter blades from the background. This can be effective if the background is relatively uniform.

Fig.6 Corner points detection

This Python script employs advanced computer vision techniques to accurately identify the red marker positioned on the fan blade, meticulously monitor its motion, and compute the RPM based on the number of complete rotations captured within the video footage. To utilize this tool, one simply needs to provide the path to the video recording, and it will dutifully deliver the precise RPM value of the rotating object under scrutiny.

4.3 EDGE DETECTION

Apply edge detection algorithms like Canny or Sobel to detect the edges of the helicopter blades. Once you have the edges, you can find the blade tip by identifying the highest point along the detected edges.

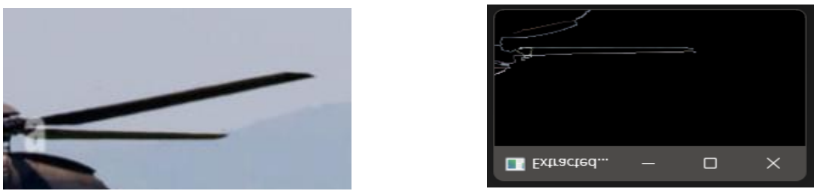

Fig.7 Edge detection of sample blade image

1. **Contour Analysis**: Utilize contour analysis to find the blade's contour, and then calculate the centre or tip of the contour. This can work well for well-defined blade shapes.
2. **Template Matching:** If the helicopter blades have distinct shapes or patterns, you can use template matching to locate them in the images.

3. **Kalman Filtering:** Implement Kalman filters or other tracking algorithms to estimate the vertical position of the blade tip over time. This can help smooth out noisy measurements.

4.4 CORNER DETECTION

The corners of an image are basically identified as the regions in which there are variations in large intensity of the gradient in all possible dimensions and directions. Corners extracted can be a part of the image features, which can be matched with features of other images, and can be used to extract accurate information.

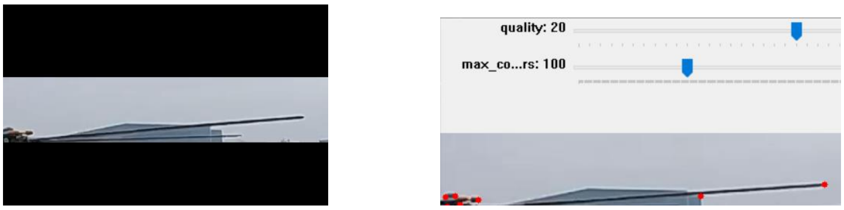

Fig.8 Blade Tip detection

4.5 FINDING BLADE TIP POSITIONS

The reposition of the tips of the rotor blades in the images. To do this, images of the helicopter rotor where the blades appeared flat (horizontally aligned) were taken. The tip of each blade looked like a straight line in the image. X-coordinates (horizontal position) of the tip of each blade were also measured, which helps to calculate how far each blade tip went. This step allowed us to know the pixel coordinates of all blades.

```
Minimum value in column 1: 1
Corresponding row: ['40', '1', '46']
Maximum value in column 1: 403
Corresponding row: ['13', '403', '56']
PS D:\BTP\CODE>
```

Fig.9 Minimum and maximum coordinates

4.6 CAMERA CALIBRATION

To utilize a camera as a visual sensor effectively, it is essential to have a thorough understanding of its characteristics. Camera calibration is the process of determining these camera parameters. These parameters encompass internal aspects such as focal length, optical centre, and radial distortion coefficients of the camera lens, as well as external elements like the camera's rotation and translation concerning a real-world coordinate system.

The camera calibration procedure entails the utilization of images featuring a chessboard from various angles and positions. OpenCV's functionalities are employed to identify the corners of the chessboard within these images. These detected corners serve as the basis for computing critical parameters, including the camera matrix, distortion coefficients, rotation vectors, and translation vectors.

The proposed project utilizes computer vision and is organized into two main stages:
In the first stage, which is called blade tracking, videos of the tips of the helicopter rotor blades are captured and analysed using image processing methods. The primary goal is to compute the extent of misalignment or "out of track" condition by examining the frames. The program

calculates the amount of misalignment between the blades by comparing their positions in the frames. This knowledge is crucial for identifying and fixing any problems with blade tracking and alignment. In the second stage, calibration, which establishes a relationship between pixels and actual distances, a reference table is made. This calibration process is carried out by capturing images of a known rectangular shape, enabling the calculation of the pixel-to-distance ratio. This table serves as a critical reference point for converting pixels into real-world distances in subsequent stages of the process.

5. Future Work

5.1 Model Training and Performance Parameters

The three models mentioned above will be trained separately using the pre-processed and selected training data. Cross-validation will be employed to tune the hyper-parameters for optimal performance. After training the different models, their performance will be tested using parameters namely Recall and precision, CM, T^2 Statistics, and F1 score. The model with the best performance across all these parameters will be selected for further processes.

5.2 Utility Theory Framework and Final Integration

Utility theory was employed alongside ML-generated probability scores to assist decisionmakers in deciding the optimal timing for implementing maintenance activities, ensuring costeffectiveness. Finally, the selected model will be integrated into an industrial equipment for real-time PdM.

6. References

[1] Bechhoefer, E., Fang, A., & Garcia, E. (2013). Rotor track and balance improvements. Annual Conference of the PHM Society, 5(1). [*https://doi.org/10.36001/phmconf.2013.v5i1.2216*](https://doi.org/10.36001/phmconf.2013.v5i1.2216)

[2] Duda, R. O., & Hart, P. D. (1972). Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), 11–15. [*https://doi.org/10.1145/361237.361242*](https://doi.org/10.1145/361237.361242)

[3] Mu, Z., & Li, Z. (2018). A Novel Shi-Tomasi Corner Detection Algorithm Based on Progressive Probabilistic Hough Transform. IEEE. [*https://doi.org/10.1109/cac.2018.8623648*](https://doi.org/10.1109/cac.2018.8623648)

[4] OpenCV: Camera calibration. (n.d.). [*https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html*](https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html)