KINGSTON UNIVERSITY

FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

Use of Sensors and Actuators in Robotic Systems

submitted by

Raghunandan Subramaniam	**K1901121**
Sameeksha Sridhar	**K1932406**
Tasneema Akhtar	**K1932976**
Nilendu Saha	**K1943079**
Joseph Abraham	**K2001031**

An M.Sc. *ME7732 Module Report* Submitted for the Postgraduate Degree of Master of Science (M.Sc.) *Mechatronic Systems*

Assessment set by: Dr Olga Duran and Akin Delibasi

18th May 2020

Abstract

The inverted pendulum is a commonly studied system because of the simple yet challenging objective of balancing it. In this report, at first a literature review was conducted to study and obtain the suitable pendulum model. The dynamics of this system were studied and change in different parameters on it was researched upon. In order to design a control system, sensors are required. In this report, the angle sensor was chosen to be suitable for this task. Computer Vision is a field of study where certain algorithms are developed in order to make the computer extract out important content from a certain image or a set of image frames. Such information can be useful for any kind of tasks where visualization should be a key required skill. For this assignment, a task was assigned to measure the angle of distribution of an inverted pendulum. The algorithm was prepared in Matlab and it follows a certain workflow. Both images and video frames were processed in order to get a desirable result. Two different control system techniques were performed. Both PID and fuzzy logic controller yielded desirable results, with the pendulum balancing after disturbances were applied to it.

Contents

Table of Contents

Title 1

Abstract 2

Contents 3

Table of Figures 5

Table of Tables 5

i. Introduction 6

ii. Task 1- Mechanical Design and Modelling 6

a. Literature Review 6

b. effects of the mechanical design on the dynamics of the system 7

iii. Task 2- Sensors and Actuators 8

a. Actuators 8

i. Working Principles of Different Actuators 8

ii. Comparison of Different Actuators 9

iii. Selection of Servo Motors 10

b. Sensors 11

c. Compare the different sensors based on the investigation and assess their suitability for the task. 12

iv. Task 3- Computer Vision 12

a. Methodology 12

i. Image Acquisition 12

ii. Image Enhancement 12

iii. Image Restoration 13

iv. Morphological Processing 13

v. Feature Extraction 13

vi. Angle Calculation and Output Generation 13

vii. Testing and Discussion 15

b. Strategies to Improve Accuracy 18

c. Accuracy and Limitations of Computer Vision as a tool for the Inverted Pendulum Task 19

v. Task 4- Modelling and Control 19

a. Approximate Linear Model to Describe the System 19

b. PID strategy Capable of Stabilising the Inverted Pendulum 21

c. Simulation in Matlab/Simulink and Critical Discussion of the Results. 23

vi. Task 5- Fuzzy Control 26
  a. Fuzzy Logic Control 26
  b. Fuzzy Logic strategy 27
  c. Fuzzy Logic Implementation in Matlab 28
  d. Fuzzy Logic Tuning Process 33
  e. Comparison between PID and Fuzzy Logic Controller 34
vii. Video Demonstration 35
Discussion and Conclusion 36
References 37
Bibliography 38
Appendices 39
  Appendices A – Commented m file for computer vision 39
  Appendices B – Fuzzy Controller 49
  Appendices C – Video Link 50

the sensor distance from the object of interest. In addition, like ultrasound, the reading from this sensor must be converted to an angle so it can be used to control the pendulum position.

c. Compare the different sensors based on the investigation and assess their suitability for the task.

Overall, just considering the theoretical approach it can be concluded from the above comparison of few sensors that angle sensor would be best suitable for this application as they are simple, accurate, provides an output in units of angles, causes less friction to the rotating axis, less prone to the external noise and it can be mounted directly on to the axis of rotation. However, it is very important to compare all these sensors and their accuracies with experimental data given the opportunity.

iv. Task 3- Computer Vision

a. Methodology

The methodology for this task is quite simple to understand. A certain set of workflow was developed to process the input image. The primary objectives were to generate a suitable binary image and undergo some morphological operations to hunt down the suitable features for line detection. As these objectives gets fulfilled for an image, it becomes simple for a video file as it is a set of similar images.

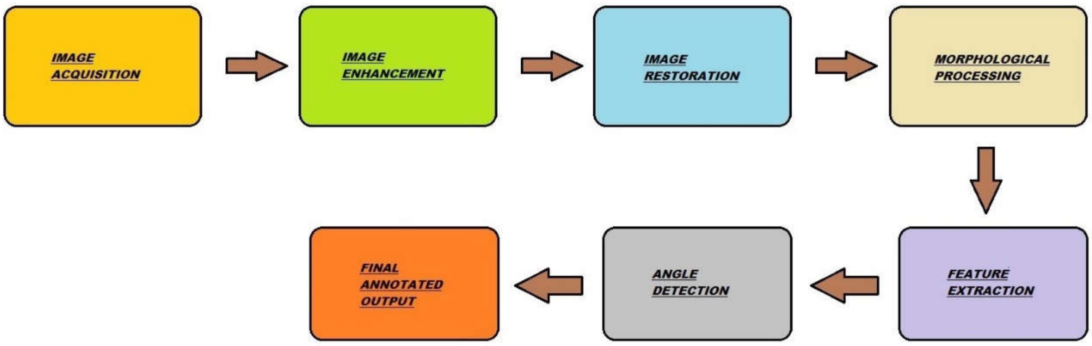


Figure 4 – Methodology

The above given figure represents a pipeline showing the methodology and protocol of the algorithm in a nutshell. These can be explained in the form of subheadings as given below. Elaborately, the methods are as follows below.

i. Image Acquisition

It is the initial stage of the algorithm where image is loaded in the form of single frame or a video. In case of a video file, each frame is acquired and loaded. For images, JPEG files have been used and for videos, MP4 file have been used.

ii. Image Enhancement

This process involves image cropping, resizing, mirroring rotation and many more. Only image rotation function was used to rotate the image files to give a better view.

iii. Image Restoration

This process forms the initial stage where the unimportant information from the images are masked out. Two methods were used to achieve this objective. The methods are as follows,

- Colour Thresholding: This method was done using a colour thresholder app in Matlab. Any wanted colour space can be selected by tuning the parameters in the HSV format. In this case, the object was yellow in colour. Hence, all the pixels having tinges of yellow are shown and rest is blackened out. This creates a lot of noise in the output.
- K-Means Clustering: This method involves the process of utilising a well-known machine learning technique known as K-Means. K-Means is used for clustering the points which have same features (for here it is colour) in the image space. This process clusters each and every pixel of the linear object and mask out the background.

iv. Morphological Processing

At this stage, the main objective is to reduce the noise present in outputs of both the above-mentioned methods in Image Restoration. A set of morphological operations have been utilized sequentially and are given as follows:

- Closing Operation
- Erosion Operation
- Dilation Operation
- Skeletonized Operation

v. Feature Extraction

This is the most important stage of the algorithm where a line is detected using the Hough Transform. There might be many lines in an image file. But in this case, the most prominent and long line was taken into consideration. After fetching the Hough Peak and the coordinates, the detected line was annotated on the image.

vi. Angle Calculation and Output Generation

It is the part where simple formulae of a straight line in coordinate geometry have been implemented to get the required angle. Most importantly, a reference line was fetched by keeping the object upright straight and following the same above process. Then the required angle was measured with respect to the reference line. Finally, the angle value is annotated too.

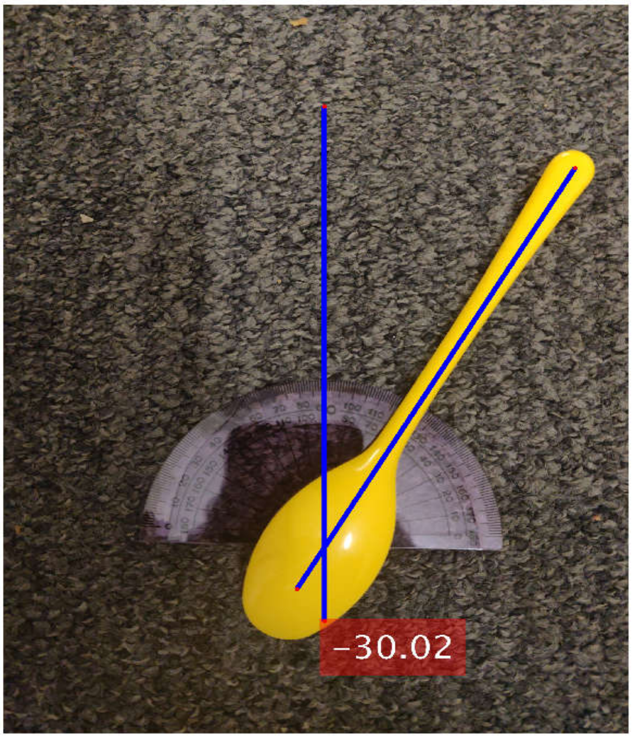
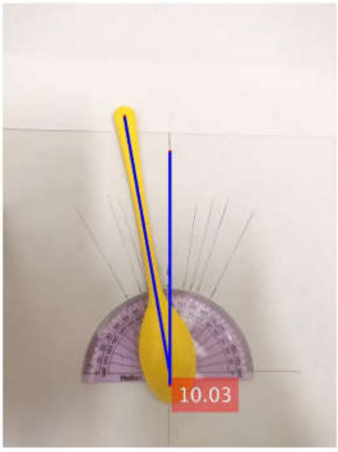


Figure 5 – Expected Output Image after Undergoing the Entire Algorithm

vii. Testing and Discussion

This algorithm was tested both on still frames with known angles and a video recording. For each of the case the coordinates of the straight reference line are kept as the same. If observed carefully, the angle values of the link at the right sided positioned link has a negative value and the left sided has a positive value. These signs indicate the direction of position of the link from the reference line.



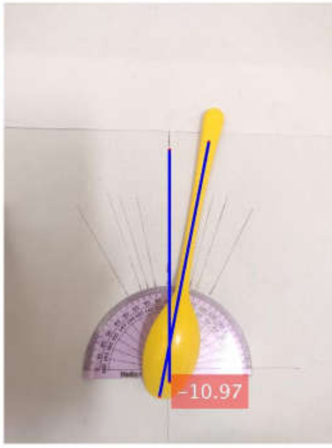


Figure 6 – Two Types of Output Extracted

The following table shows the angle of the link measured in both the above specified methods.

Known Angles	Colour Threshold Method	K-Means Method
10°	10.03°	7.01°
15°	15.04°	14°
20°	19.03°	20.01°
30°	29.01°	28.02°
-10°	-10.97°	-8.96°
-15°	-14.97°	-14.96°
-20°	-19.96°	-17.97°
-30°	-26.96°	-24.99°

Table 1–Angle of the Link Measured in both Specified Methods

As per observing the results from the table above, it can be commented that colour thresholding method was quite satisfying for this kind of setup. A small calculation of error can also be conducted on such results.

The following table shows the error of measuring the required known angles in both the above specified methods.

Known Angles	Colour Threshold Method	K-Means Method
10°	0.03°	2.99°
15°	0.04°	1°
20°	0.97°	0.01°
30°	0.99°	1.98°
-10°	0.97°	1.04°
-15°	0.03°	0.04°
-20°	0.04°	2.03°
-30°	3.04°	5.01°

Table 2– Error of Measuring the Required known Angles in both the Above Specified Methods.

Hence, the maximum and minimum errors are 0.03° and 3.04° for the Color Threshold Method, 0.01° and 5.01° for the K-Means method. Overall the first method has quite satisfactory results for the known angles.
The algorithm as a lot of tunable parameters at different stages. They are sequentially mentioned below as follows:

- **Image rotation using imrotate function**: An image can be rotated in 360 degrees. For this task, the image must be rotated in a correct manner so that it mimics like and inverted pendulum and the further calculations can be done easier.
- **Colour Thresholding**: It is done by a Matlab application where the image is threshold in HSV format.

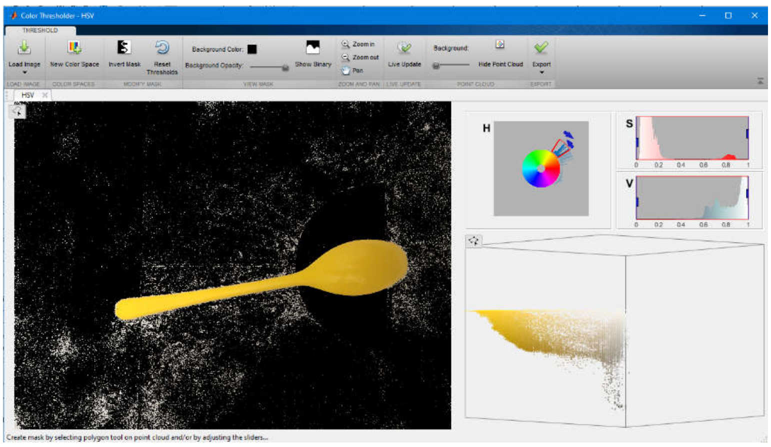


Figure 7 – Method of Bringing out the Important Visual Features

The above figure shows an instance where the objective was to concentrate and bring out the important visual features of the object of interest. This operation can be done by tuning the Hue, Saturation and the Value parameters. H parameter is tuned to make the link visible and mask the rest, in terms of colour (i.e. here all pixels in yellow are only activated). The S and V parameters are tuned to reduce the background noise. Hence all these parameters must be visually tuned and checked.

- **K-Means Clustering**: In such a method, tuning the method and checking the output is really an important part. Number of coloured clusters and the number of run times can be manually selected. The default number of clusters is 2. For this task, the required coloured cluster was identified with the value of 3.

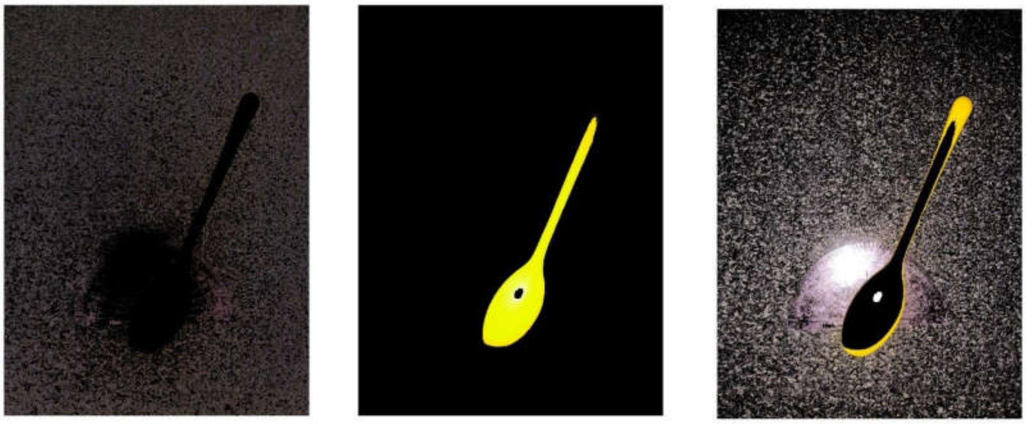


Figure 8 – 3 Cluster Outputs Formed after Running k-means on the Input Image

The above given figure represents the 3 cluster outputs formed after running k-means on the input image. In this case cluster 2 which is fig the right image to carry out with the rest of the algorithm.

- **Morphological Closing using imclose**: This is a morphological procedure where the main content of the image is clarified by not destroying the important information.

- **Erosion using imerode and Dilation using imdilate**: These two morphological functions were used to scrape out and form in feature parts respectively. These functions are used to bring good shape to the features of the image. Erosion in specific parts is done to remove noise particles. Whereas, dilation is used to form some particles which were accidentally trimmed off earlier after erosion. Certain arguments of both the functions can be altered to specify the type and extent of erosion and dilation. The user must put some arguments in order to use the functions.

  Hough Lines in Hough Transform: The above given figure shows the syntaxes used in the line detection method. In order to utilise the algorithm, the skeletonized thin line is used as an input, as shown in line 61. Further, '**FillGap**' and '**MinLength**' are used to connect multiple detected lines whose end points lie in the same line. Hence these gaps are filled to make it a single long line. This simple piece of code must be included otherwise it will be very difficult to consider all the other multiple bits of lines. The values put in as arguments totally depend upon the resolution of the image. It should be equal to or less than the diagonal length of the image.

```
% Extract the required line from hough transform and peaks
hLines = houghlines(BMSkel,T,R,hPeaks,...
        'FillGap',1500,'MinLength',100);

%% View The Detected Line
% Annotate the required line
[linePos,markerPos] = getVizPosArray(hLines);

lineFrame = insertShape(rotated_img,'Line',linePos,...
            'Color','blue','LineWidth',30);
outFrame = insertObjectAnnotation(lineFrame,...
            'circle',markerPos,'','Color','red','LineWidth',15);

```

Figure 9 – Syntaxes used in the Line Detection Method

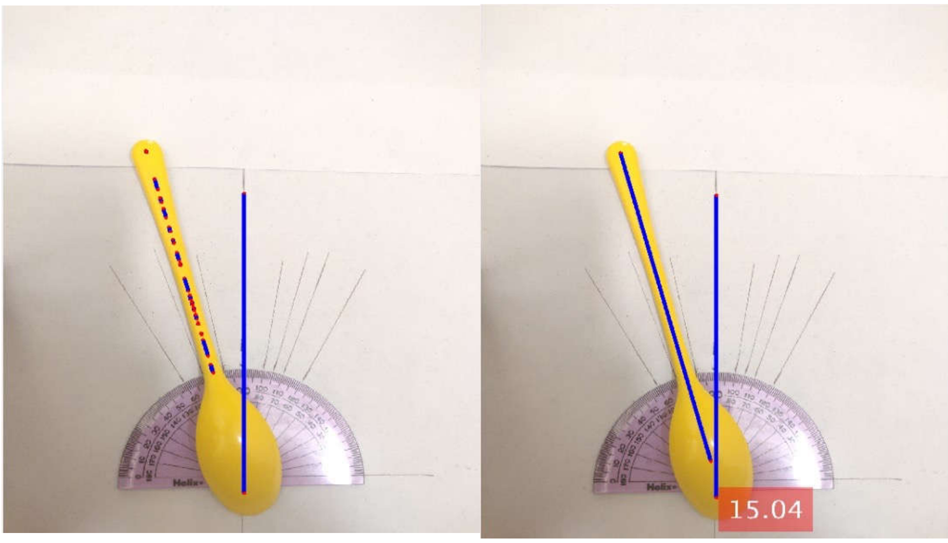


Figure 10 – Problem Faced and the Required Solution

b. Strategies to Improve Accuracy

Some strategies can be followed in order to make the algorithm much more accurate:

- A dark background should be chosen in order to make the object more prominent in front of the camera. If this is not maintained, then the binary image won’t appear as expected.
- An ample amount of balanced light should be provided to the frame. Or else the shadows might bring undesirable noise to the frame.
- The object to be focussed on should be having a light and a bright colour. For this case, a light-yellow coloured spoon was used.
- For the video task, any hazy footage can disrupt the detection process. So, all the videos should preferably be taken from a camera mounted on a stable stand.

- Other than all the above strategies, another best strategy is to tune the mentioned parameters in order to get the desired result.

c. Accuracy and Limitations of Computer Vision as a tool for the Inverted Pendulum Task

This task with respect to the whole assignment was a bit challenging. This is because there was no physical setup. A dummy setup was prepared to mimic the actual inverted pendulum and try to get the angle of distribution. But this is the beauty of computer vision, even though the real setup was not present, the algorithms created for a real inverted pendulum can be used on a similar looking object. This is the most important advantage of the usage of computer vision for this task. Hence, it can be commented that computer vision can be used even though the resources are limited. It is quite a useful tool because we humans see the situation in order to find out an angle, by judging the image in our brain. Hence, for the computers, it is quite simple to extract data from the same image.

Apart from the advantages all systems and tools have limitations too. The most primary one faced was noise. It is quite difficult to judge the solution for erasing the noise. Maybe for a single frame image the noise can be erased using a certain set of parameters. But for a video file, the noise might vary from frame to frame. This system only works when the foreground is easily distinguishable from the background. Both should have a certain amount of contrast difference in order get a suitable binary image. If more amount of morphological operations is used then, the processing time takes a bit long. So, the image should be taken in such a way that a smaller number of processing is required. Other than all these points, computer vision was a very promising tool for this task. It was very interesting, and it shows a new field of engineering in which a variety of such problems can be tackled.

v. Task 4- Modelling and Control

a. Approximate Linear Model to Describe the System

The system consists of an inverted pendulum on a motorized cart. Two degrees of Freedom are considered in the system. A force, F applied on the cart results in a linear displacement x of the cart and an angular displacement θ in the pendulum.
For designing the system, the following considerations are made:

- The settling time of the pendulum angle is always less than 5 seconds.
- The pendulum anlgle θ is never greater than 0.05 radians from the vertical.