University of Colorado Boulder

EXERCISE 6

MOVING OBJECT DETECTION AND FOLLOWING ROBOT

ECEN 5863 Real Time Embedded System

BY

Amreeta Sengupta and Ridhi Shah

04/23/2019

Final exercise proposal

Overview and Design description

We aim to develop a system for the detection and tracking of a circular object. The system will first detect the shape of the object, determine its color and then classify the object as a golf ball based on the previous shape and color detection using the OpenCV libraries. It is also the secondary objective to develop a ball following mechanism by tracking the position of the circular object in the frame and driving two motors to follow the golf ball.

First the logitech camera and the openCV libraries are configured on the Jetson TK1 board. We use OpenCV for detection of shape and color of the object. To determine the shape, we use the hough elliptical transform and develop the algorithm for detecting the circular shape. Then we detect the colour of the object, if white color is detected then we determine that it is the object to be tracked and the robot has to follow the object if it is moving. For determining the direction in which the robot has to move, we calculate the offset of the circle from the entire image captured, i.e. we take the difference between the center of the image and the center of the circle and depending on its value a decision is made about the direction in which the robot will move. If the offset is positive, then the robot moves in the right direction and if it is negative the robot turns left otherwise it goes straight.

We have chosen the Rate monotonic policy for the implementation of services, the application has been divided into two services, one service is used for the image processing part where the shape and color are detected, and the offset value is computed. This offset value is used in the Motor control task where the motor uses this and determines the direction of the movement of the robot. We have calculated the WCET and the deadlines based on how much time will it take to execute the Hough elliptical algorithm, the color detection algorithm and also calculating offset of the image. Similarly, we have calculated WCET for the motor control task based on the decision-making algorithm which uses the offset data and determines the direction. We run cheddar using the WCET, T and D for the system. We can see that this service set is schedulable. The 2 tasks share the data of the offset and the tasks have to be synchronized, this synchronization will be taken care of in the implementation.

Overall, the system working includes the following steps:

1. The Logitech C200n camera is used to detect an object.
2. Once, an object is detected, its shape is determined.
3. If a circle is detected, the system then proceeds to determine its color.
4. If a white circle is detected (which is basically a golf ball), the offset of the circle is calculated from the center of the image which in turn will issue turning orders to the motor.
5. To control the speed of the motor, the PWM is configured on the GPIO pins.
6. If a moving object is detected, the motors are configured and driven in a manner such that the robot follows the object. Thus, it helps in keeping track of the object.

Citations

- https://pdfs.semanticscholar.org/549a/2bff5d595e759156fb2cb9bad14d5a2fc892.pdf:This paper tells us about the use of different Scheduling policies.
- https://www.researchgate.net/publication/4277229_PARTES_Performance_Analysis_of_Real-Time_Embedded_Systems: This paper tells us about the performance analysis of Real-Time Embedded Systems.
- http://www.allresearchjournal.com/archives/2015/vol1issue9/PartG/1-9-20.pdf: This paper tells us about use of OpenCV in object tracking and detection.

Personal contribution

The OpenCV configuration must be done and algorithm for detection of shape and color of the object has to be developed. After detecting that the object is a white ball, the offset is calculated by taking the difference between the center of the circular object and the center of the entire image is calculated. The offset calculated is used to determine whether the object has to turn left or right, and the motor is controlled according to it.

Ridhi - Shape detection algorithm uses the Hough transform to detect the circle. The coordinates of the center of the circle are also displayed. If the shape detected is a circle, then we go ahead and detect the color of the object. The color detection of the object and determining the color depending on the grayscale and setting up of other parameters will be done and coded accordingly, then a code is developed which computes the difference between the center of the entire image. The synchronization between the 2 tasks, i.e. the shape and color detection and the motor control will also be taken care of.

Amreeta - According to the offset values computed, the motor has to be controlled. Initially, the motor configuration has to be done and then a power supply is given to the motor. Next, the motor driver PWM pins are configured using GPIO. The proper functioning of the motor and movements of the motor, according to the PWM input, will be tested first. Then the offset values determined previously, will be used and the code is developed to control the motor according to these values. A proportional control algorithm will be used for this and depending on the offset values, we will determine the direction in which the motor will move i.e. left or right. If the offset value is positive, then the robot moves to the right and if it is negative the robot moves to the left.

Functional (capability) requirements for system design

- Shape Detection
  The circular shape is detected by using the Hough Elliptical Transform algorithm using OpenCV. The Hough transform is the feature extraction method used in image processing, The Hough transform was originally used to identify the lines in the image, but the Hough transform has been extended to identifying arbitrary shapes, most commonly circles or ellipses. HoughCircles function can be used to find the circles in a greyscale image.

- Color Detection
  This is done by first defining the upper- and lower-pixel values and then inRange function is used to return a mask which determines the pixels that fall in the defined range. Bitwise_and function can be used to apply the mask on the image and determine the color.

- Object Tracking
  The camera is used to capture the image of the object at each instance. Hence, as the object moves, the coordinates of the object change. We can continuously check for this change to calculate the offset of the circle from the center of the image and determine the direction in which the motor will be driven.

- Motor Control
  Motor Driver TB834A2 is a dual motor driver which is used to interface two DC motors. It is used to drive the motor to change the direction of the robot. For taking turns, one wheel is stopped while the other wheel will be turned in the desired direction.

- Proportional Controller
  On tracking the object, the distance between the center of the circle and the center of the image is calculated to determine the offset value. If this value is less than zero, then the proportional controller will decide to make a left turn and otherwise it will take a right turn. The proportional controller basically decides the direction in which the motor will be driven.

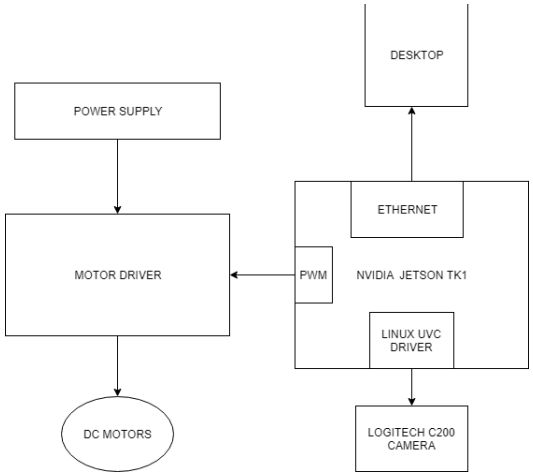
- Synchronization between Tasks
  The code runs a sequencer with two services. The sequencer runs at HCF of request periods of the two tasks. The priority is assigned based on Rate Monotonic Policy which states that higher priority will be given to a service with higher frequency. According to the frequency of the services, we make an algorithm to post the semaphores to maintain the schedule. The semaphores get posted and then the sequencer goes to sleep. When the sequencer is in sleep, the semaphores which are posted run to completion. This cycle goes on and on for the sequencer period.

Completeness:

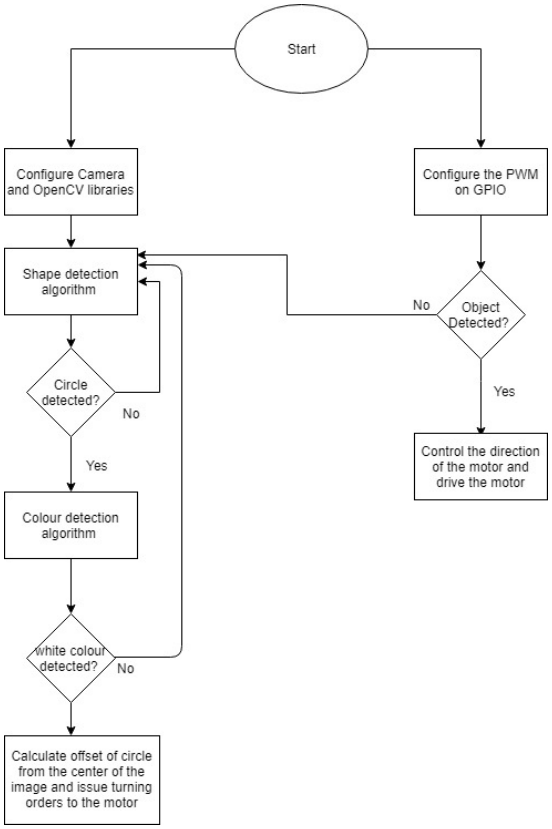
The Rate monotonic policy is used here since 2 tasks are independent. The motor control task depends on the image processing task if the values are to be updated, otherwise it is an independent task. The WCET, T and D are calculated accordingly and scheduled. The 2 service sets are schedulable. Thus, the system is designed to meet the requirements.

High level system and software design

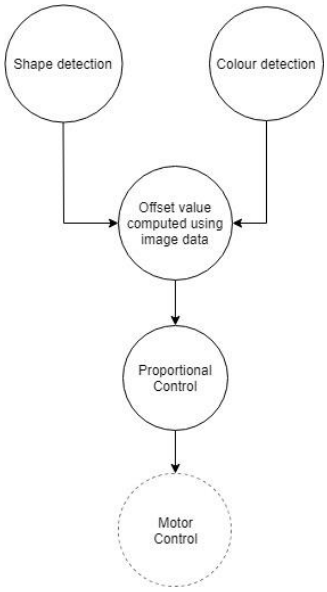
BLOCK DIAGRAM



FLOWCHART



DATAFLOW DIAGRAM



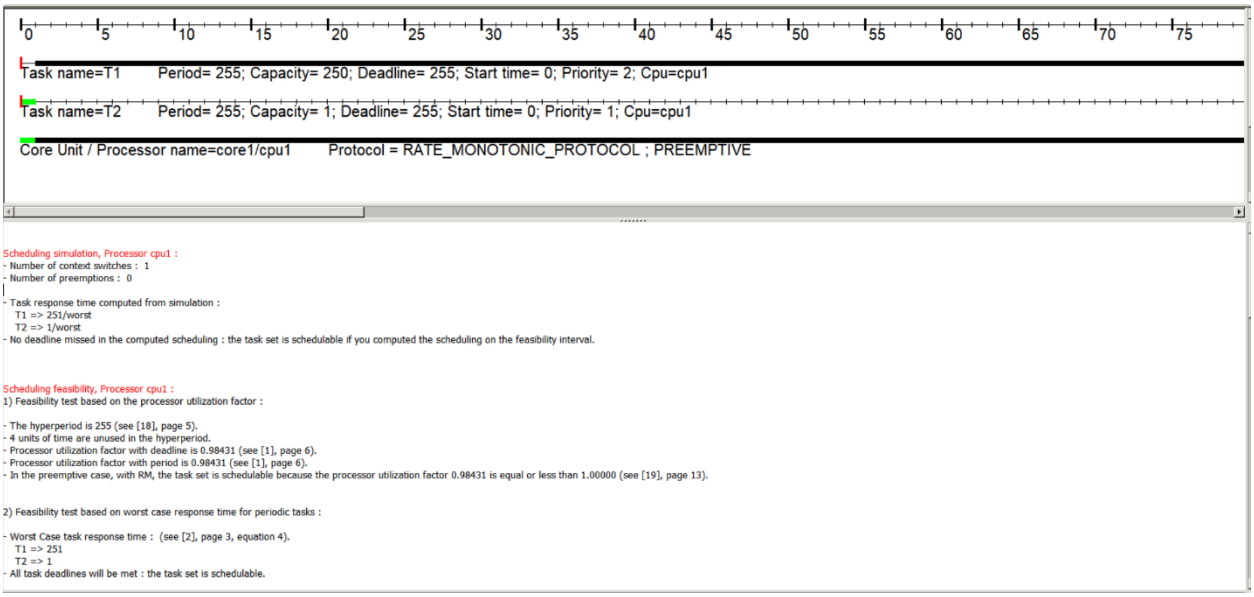
Functional Description

The camera is used to capture the image of the object at each instance. From this, the color and shape of the object is detected through the use of the color and shape detection algorithms. As the object moves, the coordinates of the object change. We can continuously check for this change to calculate the offset of the circle from the center of the image. This offset value is basically the distance between the center of the circle and the center of the image. The value will help the proportional controller determine the direction in which the motor should turn. If this value is less than zero, then the proportional controller will decide to make a left turn. Otherwise, it will take a right turn. The motor will then be driven by the motor controller in the direction determined. This will result in the robot following the golf ball.

Real-Time Services and Requirements

Services

Tasks (S_i)	T_i	WCET (C_i)	D_i
Image processing	255 ms	250 ms	255 ms
Motor Control	255 ms	1 ms	255 ms



Service 1 is the image processing task which is basically used to detect the object and identify its shape to determine if it's a circle. If its shape is circular, then its color is determined and if its is identified to be white, then golf ball is detected. On detecting the object, the distance between the center of the circle and the center of the image is calculated to determine the offset value.

Service 2 use is basically used to configure and control the motor. It uses the offset value determined above to control and drive the motor. If this value is less than zero, then the proportional controller will decide to make a left turn. Otherwise, it will take a right turn. This will basically result in the robot following the golf ball.

C_i and WCET descriptions specification

Motor control thread only executes PWM and control signals for H-bridge. Any update in motor control is issued by the image processing task, in which case few register writes are required to change the GPIOs and PWM registers. Jetson, having its clock in GHz range, makes 1 ms a sufficient overestimation for few register writes. Estimation of Hough Elliptical Transform from previous exercise is 90 ms for 640 X 480 resolution on Jetson. This would further require color detection algorithm and proportional control which makes 250 ms an appropriate WCET estimate. This value will be updated with actual data in final report. Also, execution time of these algorithms depends on image complexity like number of circles in an image and the gradient of the RGB value of neighboring pixels. To estimate WCET, it is required to assume maximum number of possible circles in the image and color variation. Here, we will

assume two circles i.e. two golf balls and sharp color contrast as the ball will be white in green background of grass.

T_i and D_i specification

Schedule is implemented using RM Policy. Hence, request time and deadlines are same. Estimated WCET for image processing task is 250 ms and motor control task has WCET of 1 ms only. These values suggest that request period above 251 ms will make the overall schedule feasible even though RM LUB test will fail because of processor utilization being almost one. This is not an issue as RM LUB is only a sufficient test, not a necessary one. So, we have set request period for image processing task as 255 ms which will also account for kernel preemption in Linux. Also, any update in motor control won't occur until the Image processing task is computed. Thus, request period for motor control task is also set to 255ms.

REFERENCES

- https://docs.opencv.org/3.4.2/dd/d1a/group__imgproc__feature.html#ga47849c3be0d0406ad3ca45db65a25d2d
- https://www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/
- https://www.pololu.com/product/713
- https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/