MA1008 Introduction to Computational Thinking
Mini Project: A Cam Design System
Semester 1, AY 2021/2022, Week 10 – Week 13

1. Introduction

The objective of the mini project is for you to produce a program of a moderate size and depth that will require you to utilise what you have learned in this course, and more, to do something interesting. Through this, you would learn to design, manage and execute a sizable program.

2. The Project

Disc cam is a common mechanism used in many machines, including car engines. It converts the rotational motion of the cam to a translational motion in the follower. This website shows some examples:
https://www.creativemechanisms.com/blog/common-mechanisms-explained-with-animation.-part-1.

This project requires you to write a program in Python for designing disc cams.

A cam system has two main components: the cam and the roller follower, with the follower always in contact with the circumference of the cam. Its shape (also called profile) is a smooth closed curve determined by the motion of the follower. This motion is periodic, repeated over a revolution of the cam, with three basic parts: rise, dwell and return, which you can observe in the animation in the above website. Rise is where the follower goes up, dwell is where it remains stationery and return is where it falls. The follower motion is defined in a displacement graph, as shown in Figure 1.

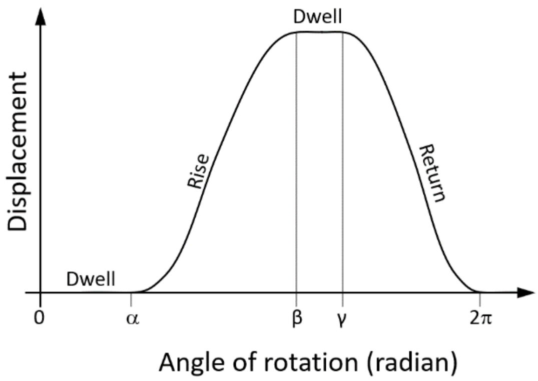

Figure 1: Displacement graph; α, β, γ are the angles at which one motion changes to another

The dwell is a horizontal straight line in the displacement graph, the rise and return curves are smooth mathematical curves that connect tangentially with the dwell line at their end points. For different dynamic behaviours, the rise and fall curves may take different forms. The mathematical formulae of the curves are given in Appendix 1.

Figure 2 shows a cam with its constituting elements. There is always a base circle whose circumference provides the lowest position for the follower. The pitch circle is the circle on which the centre of the follower lies at this lowest position. As the cam rotates, this follower centre traces out the pitch curve, which is basically the displacement curve wrapped around the pitch circle. The cam profile is formed by offsetting the pitch curve by the radius of the follower in the direction perpendicular to the pitch curve towards the centre of the cam. Therefore, this profile is dependent on three things: the radius of the base circle, the displacement curve and the radius of the follower. The mathematical formulae for the cam profile are given in Appendix 2.

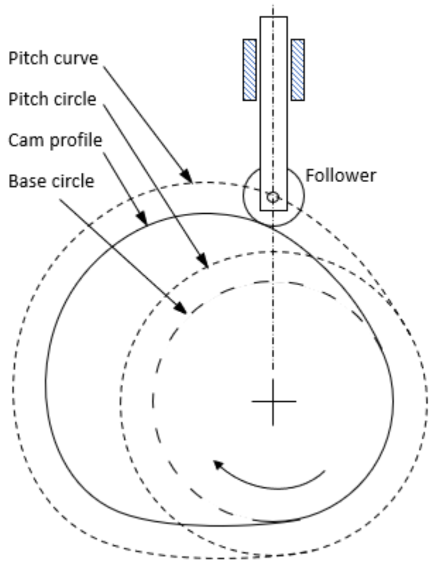

Figure 2: A cam with its elements. The pitch curve is the curve traced by the centre of the follower whose displacement is the desired displacement.

The Program

You are to write a program for designing disc cams. Your program should do the following:

i. Obtain inputs from the user for defining the displacement curve over one revolution of the cam. This requires the number of motion segments, the angular range for each segment, and the type of motion for each segment. Note that the number of segments is variable.
ii. Display the displacement graph.
iii. Obtain inputs from the user for the radius of the base circle and the radius of the follower.
iv. Generate the cam profile, display it together with the base circle and the follower. Your display only needs to be in 2D.
v. (Optional) Create an animation showing the cam in rotation with the follower moving accordingly. To create the animation, you also need the speed of rotation of the cam.
vi. For testing purposes, you need to provide a data file that contains the input data required for the definition of a cam system. When the data is read in by your program, the displacement curve and the cam system are displayed.
vii. Provide at least four sets of test data in your file. Make sure that upon the execution with one set of data, your program pauses for the user to examine the output. Only upon a user command that the program moves on to the next set of data.

Here are some points you should note:

i. The size of a cam depends on the radius of the base circle and the maximum follower displacement. They can be any size. But as a rough guide, the maximum displacement should be no more than the diameter of the base circle.
ii. The number of motion segments is variable, but typically no more than six.
iii. Your screen size is fixed, but your cam size is variable. Therefore, you should scale your display such that the graph and the cam fit nicely in the display window.
iv. You may choose to have the displacement graph and the cam displayed side by side in the same window or in separate windows.
v. Your display should be properly annotated, especially the graph. The axes should have tick marks marking the values in incremental steps and correctly labelled.

3. Resources

The main extra resource that you will need for this project is for the graphical display, for which you need the graphics library provided with Python, called **Turtle**. Turtle capabilities and functions are described fully in Section 24.1 of the Python documentation, which you can access via the IDLE interactive shell by clicking <u>Help</u> > <u>Python Docs</u> and then search for "Turtle". Appendix 3 provides some guidance on the use of Turtle.

4. Prohibitions

Apart from the graphics library, everything you need for this project can be done through what you have learnt in this course. To allow you to exercise what you have learned and to prevent you from veering wildly beyond the scope of this project, your program <u>should</u>:

i. Not use the *class* construct to define objects you require in your program
ii. Only use the Turtle library for graphics. You cannot use *tkinter* or *matplotlib*.
iii. Not use the library *numpy*.

You may use the standard Python libraries like *math*.

5. Submission

You need to submit the following:

i. **A working Python program** that gets the relevant inputs from the user or from a file and displays the displacement graph and the resulting cam.
ii. **A data file** containing the required data for four different cams which your program can read.
iii. **A report in a Word file**, providing
    - A guide on how to run your program

- Pictures of your graphics window, with displays of the displacement curve and the resulting cam, together with the input data for generating the cam. You should provide pictures for at least two different cams.
- Highlights of the **key strengths and limitations** of the program.

The **deadline** for project submission is **23:59 hours, Friday 12 November**. Please submit all your files through the course site using the same link to the project file.

6. Grading Rubrics

Here are what the graders will be looking for when grading:

i. The quality and correctness of the outputs, especially the graphics display.

ii. The ease in interacting with the program, which includes specifying the input data and working with the input files.

iii. The quality and correctness of your program, which includes:
   - How easy it is to read and understand your program. This means your program should be adequately commented with good choice of variable and function names.
   - Modularisation of the program including appropriate use of functions.
   - The design of the functions which include the appropriateness of parameters in the functions.

7. Epilogue

You should start working on the program immediately. Do not wait till the last week. What you produce depends on the time you spend on the project and your ability in creating good algorithms and writing good code.

This is an individual project. You may consult the tutors and discuss with your classmates, but everyone must write their own program. Any programs deemed to have been copies of each other will be penalised heavily, regardless of who is doing the copying.

If in doubt or in difficulty, always ask. And ask early!

There is a great deal more to the science and engineering of cams than what you are required to do in this project. We have trimmed the project requirements so that it is doable for a first-year engineering computing project. For more information on cams, you may refer to books on engineering mechanisms, such as Mechanisms and Mechanical Devices Sourcebook by Sclater and Chironis.

Appendix 1: Displacement curves

The follower displacement, y, is a function of the angle of rotation of the cam, θ. Hence,

$$y = f(\theta),\ 0 \le \theta < 2\pi \tag{1}$$

For the displacement curve, the 2π range is divided into consecutive segments with each segment between two angles thus: $\theta_{i1} \le \theta < \theta_{i2}$, i = 1 … n, where n is the number of segments, $\theta_{i2} = \theta_{(i+1)1}$, $\theta_{11} = 0$ and $\theta_{n2} = 2\pi$.

There are three different types of motion: rise, dwell and return, each with its own curve type on the displacement graph.

The dwell curve on the displacement graph is a horizontal straight line. For rise and return, there are different choices of curve depending on the dynamic characteristics the designer wishes to achieve with the cam. For this project, we limit the choices to three: constant acceleration, simple harmonic and cycloidal.

1.1 Constant Acceleration

This is a polynomial curve that ensures that the acceleration of the follower is constant. The plot of the curve over its period is shown in Figure 3.

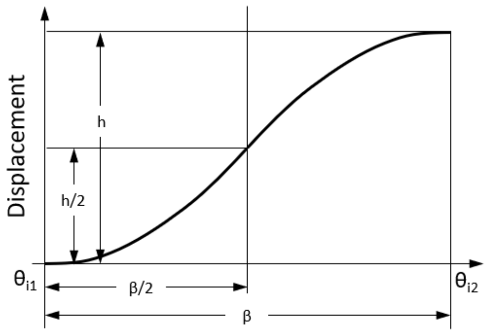

Figure 3: Constant acceleration motion graph

The span of the curve is $\theta_{i1} \le \theta < \theta_{i2}$ and $\theta_{i2} - \theta_{i1} = \beta$ and the full displacement here is h. The curve has two parts, each occupying half the span. It can be shown that

For the rise displacement, the first half is $y = 2h(\theta/\beta)^2,\ 0 \le \theta < \beta/2.$ (2)
and the second half is $y = h(1 - 2(1 - \theta/\beta)^2),\ \beta/2 \le \theta < \beta.$ (3)

For the return displacement, the first half is $y = h(1 - 2(\theta/\beta)^2),\ 0 \le \theta < \beta/2.$ (4)
and the second half is $y = 2h(1 - \theta/\beta)^2,\ \beta/2 \le \theta < \beta.$ (5)

1.2 Simple harmonic Motion

The simple harmonic motion is a sinusoid motion as shown in Figure 4. It is one single curve, and it can be shown that its rise displacement is

$$y = h(1 - \cos(\pi\theta/\beta))/2 \tag{6}$$

with $\theta_{i1} \le \theta < \theta_{i2}$ and $\beta = \theta_{i2} - \theta_{i1}$. Its return displacement is

$$y = h(1 + \cos(\pi\theta/\beta))/2 \tag{7}$$

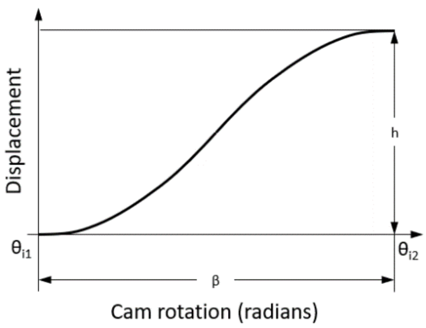

Figure 4: Simple harmonic motion graph

1.3 Cycloidal Motion

The displacement diagram for cycloidal motion is obtained from a cycloid which is the locus of a point on a circle as the circle rolls on a straight line. The graph for the motion is similar to that of simple harmonic motion shown in Figure 4. The rise displacement equation for the motion is

$$y = h\,(\theta/\beta - \sin(2\pi\theta/\beta)/(2\pi)) \qquad (8)$$

with $\theta_{i1} \le \theta < \theta_{i2}$ and $\beta = \theta_{i2} - \theta_{i1}$. The return displacement equation is

$$y = h\,(1-\theta/\beta + \sin(2\pi\theta/\beta)/(2\pi)) \qquad (9)$$

Note that the equations of all the curves given above are such that the slopes of the curves at their end points are horizontal. This is to facilitate their connection to a dwell segment which is a horizontal straight line. It is essential that two curves are joined smoothly with equal tangent directions at the joining points, to ensure a smooth cam profile, which is essential for proper functioning of the cam.

Appendix 2: The cam profile

For this project, we use a follower with its axis passing through the centre of the base circle. Let

R_b = radius of the base circle
R_f = radius of the follower.

To obtain the cam profile, we first determine the pitch curve, which is the curve traced by the centre of the follower. This is given by the equation

$$R = R_0 + f(\theta), \quad 0 \le \theta < 2\pi \qquad (10)$$

where $R_0 = R_b + R_f$, the lowest dwell position at θ = 0, and f(θ) is the displacement of the follower at angle θ, as stated earlier. See Figure A1. Hence the (x, y) coordinates of the centre of the follower, **C**, is given by

$$x = R\cos(\theta)$$
$$y = R\sin(\theta) \qquad (11)$$

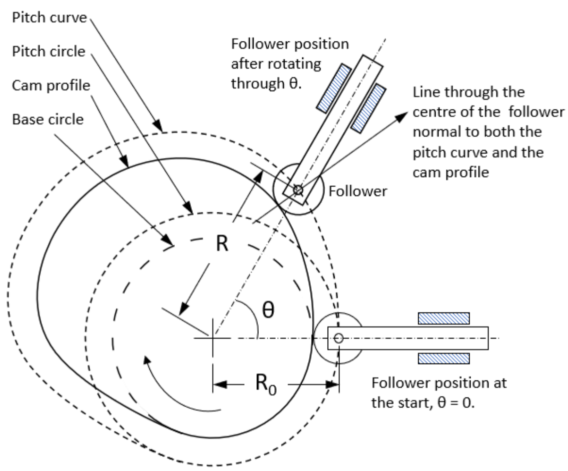

Figure A1: Cam, follower and the rotation angle

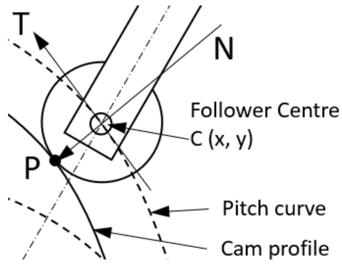

Figure A2: Pitch curve tangent and normal at rotation angle θ

The tangent direction of the pitch circle at the follower centre is

$$\mathbf{T} = \left(\frac{dx}{d\theta}, \frac{dy}{d\theta}\right) \qquad (12)$$

$$\frac{dx}{d\theta} = x' = \cos(\theta)\frac{dR}{d\theta} - R\sin(\theta)$$
$$\frac{dy}{d\theta} = y' = \sin(\theta)\frac{dR}{d\theta} + R\cos(\theta) \qquad (13)$$

Let **P** be the point of contact between the cam profile and the follower. **P** is therefore at a distance R_f from the centre of the follower. See Figure A2. The follower must be tangential to the cam profile, which means the line joining the centre of the follower to **P** must be normal to the cam profile at P. Let this normal be **N**. The cam profile must be parallel to the pitch curve, at a constant offset of R_f. Therefore, **N** must be perpendicular to **T**. The cross product of two perpendicular vectors in 2D is the zero vector, therefore **N** x **T** = **0**, which means

$$\mathbf{N} = (-y', x') = \left(-\sin(\theta)\frac{dR}{d\theta} - R\cos(\theta), \cos(\theta)\frac{dR}{d\theta} - R\sin(\theta)\right) \qquad (14)$$

The magnitude of **N** is

$$m = \sqrt{(-y')^2 + (x')^2} \qquad (15)$$

The unit normal direction of **N** is

$$\mathbf{n} = \mathbf{N}/m = (-y', x')/m \qquad (16)$$

We know from vector algebra that

$$\mathbf{P} = \mathbf{C} + R_f\mathbf{n} = (x + R_f(-y'/m), y + R_f(x'/m)). \qquad (17)$$

As the cam rotates, θ moves through a cycle from 0 to 2π, **P** traces out the cam profile.

You need to use f(θ) from Equations (2) to (9), as appropriate, and apply them in Equation 10 onwards to get the cam profile, as given in Equation 17.

Appendix 3: Some guidance on the use of Turtle

Turtle is a graphics library provided with Python. It contains the functions required for drawing, such as the creation of lines, shapes and posting of text, and controlling their attributes such as location, colour, and more. You should study the different Turtle functions required for your display. There are many Turtle functions, you only need to learn the ones you require. It is therefore necessary that you work out what you need beforehand.

For this project, you have two major graphical outputs: the displacement graph and the cam profile with the follower. You can display these two items in two separate windows or side by side in the same window. Therefore, you need to be able to open a window and control its size, which are done using functions in Turtle.

The sizes of the displacement graph and the cam are variable, depending on the size of the base circle and the maximum displacement the user stipulates. Therefore, the scale for the display in your window needs to be adjustable according to these values. You cannot use the pixel coordinates of the window which have fixed values. You need to have a user coordinate system, set up according to the size of the objects, and then use these to scale the display coordinates.

The displacement graph and the cam profile are curves. Turtle provides only basic functions for drawing straight lines. You can break a curve down to many connected short straight-line segments. By drawing these segments, you get what looks like your curve. The quality of the curve depends on how many segments are used to approximate it. You can experiment to find out how many segments make a good approximation. Note that if you have more segments, then there is more computation, and it takes more time to draw.

You need other graphical functions such as those for posting text and controlling colours. These are again available in Turtle. You can find the definition of these functions in the Turtle documentation.

By default, Turtle draws slowly to allow us to see how the drawing is done. However, for this project, you should draw quickly. So make sure that you set the drawing speed to the fastest possible, again doable by calling the appropriate Turtle function with the appropriate parameter value. You should hide the turtle too.