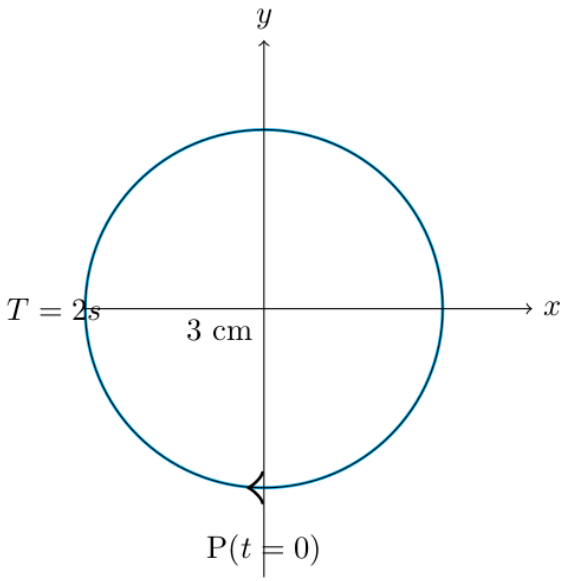
Analog 11.14.11

EE23BTECH11018 - E.Mohana*

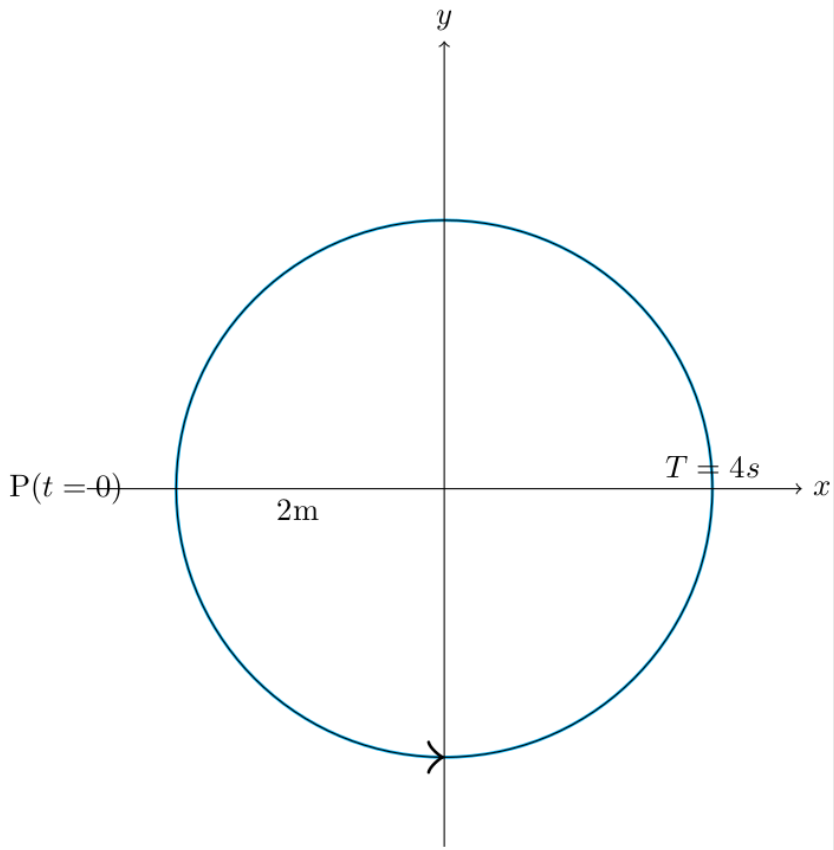
Question

Q: Figures correspond to two circular motions. The radius of the circle, the period of revolution, the initial position and the sense of revolution(i.e. clockwise or anti-clockwise) are indicated on each figure. Obtain the corresponding simple harmonic motions of the x-projections of the radius vector of resolving particle P in each case.

Question



(a)



(b)

Input Parameters Table

Parameter	**Value(a)**	**Value(b)**	**Description**
Radius(r)	3cm	2m	Radius of each circle
Time Period(T)	2s	4s	Time period
Sense	clockwise	anti-clockwise	Indicated by arrow
Initial Phase(ϕ)	$\frac{\pi}{2}$	π	Initial angle with x-axis

Table: Input parameters table

Solution

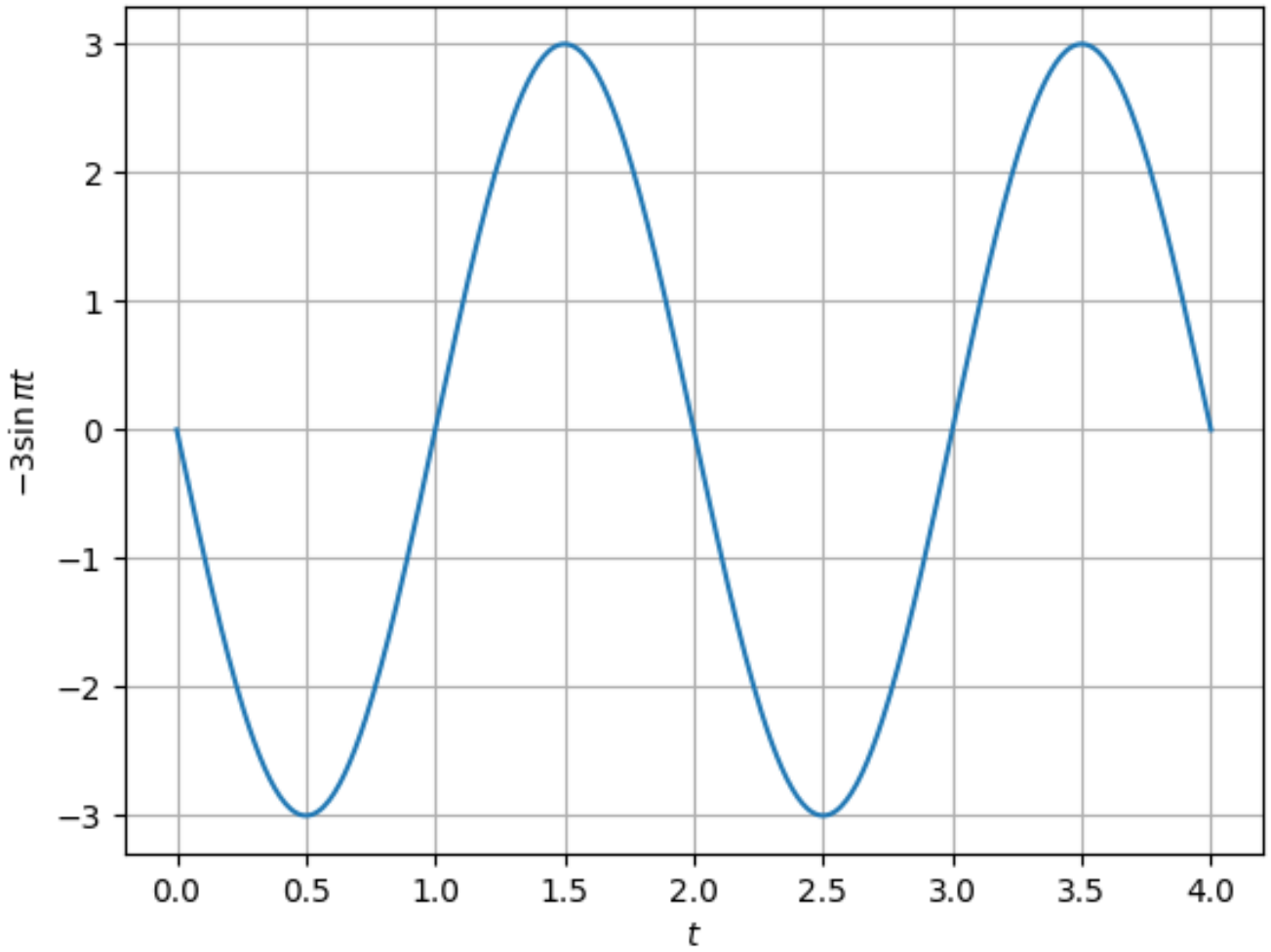
Given $\left(r\right)$ as radius vector making angle θ with positive x-axis, its x-projection $= \left(r\right)\cos\theta$
At $t = 0$, the radius vector makes an angle $\frac{\pi}{2}$ with the positive x-axis, $\phi = \frac{\pi}{2}$,
From Table 1, equation of x-projection of radius:

$$x(t) = r\cos\left(\frac{2\pi}{T}t + \phi\right) \tag{1}$$

$$= 3\cos\left(\frac{2\pi}{2}t + \frac{\pi}{2}\right) \tag{2}$$

$$= -3\sin\left(\pi t\right)\text{cm} \tag{3}$$

Plot-1



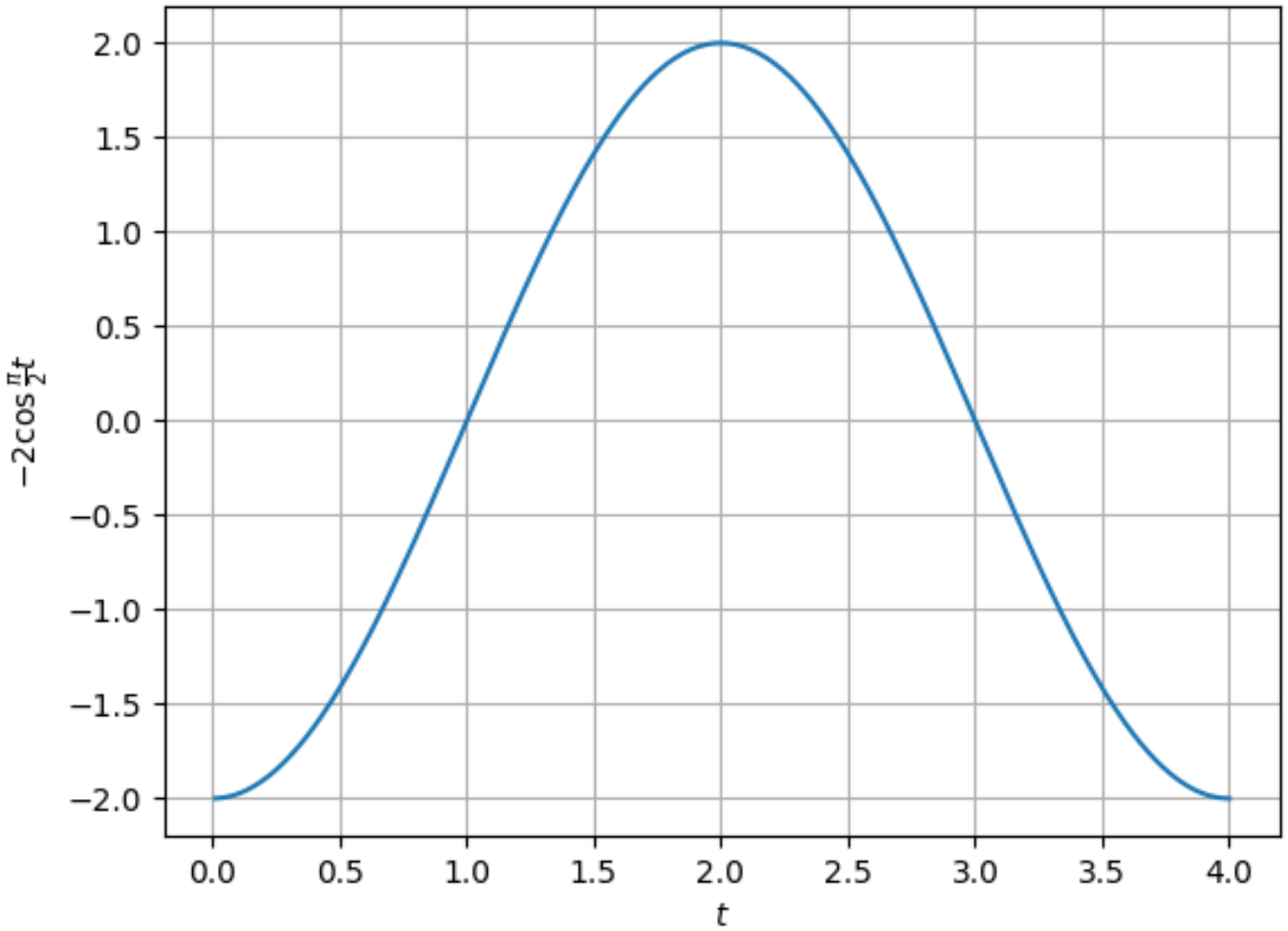
Solution contd.

Similarly,
At $t = 0$, radius vector makes an angle π with x-axis in anti-clockwise direction, $\phi = \pi$,

$$x(t) = r\cos\left(\frac{2\pi}{T}t + \phi\right) \tag{4}$$

$$= 2\cos\left(\frac{2\pi}{4}t + \pi\right) \tag{5}$$

$$= -2\cos\left(\frac{\pi}{2}t\right)\text{m} \tag{6}$$


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
$-2\cos\frac{\pi}{2}t$
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
t

C code

```c
#include <stdio.h>
#include <math.h>
#include "mylib.h"
float f(float x){
return -3 * (sin(M_PI * x));
}
float q(float x){
return -2 * (cos(M_PI/2 * x));
}
int main() {
    // Define the range and step size
    float start = 0;
    float stop = 4;
    float step = 0.01;

    // Calculate the number of values in the range
    int num_values = (stop - start) / step + 1;

    // Allocate arrays to store the generated values
    float x_values[num_values];
    float y_values[num_values];
        float(*func)(float);
        func = f;
    // Call the linspace function
    linspace(start, stop, step, x_values, y_values, num_values, func);
    //Call save function
    save(num_values, x_values, y_values);
    func = q;
    // Call the linspace function
    linspace(start, stop, step, x_values, y_values, num_values, func);
    //Call save function
    app(num_values, x_values, y_values);
    }
```

C-Library code

```c
#include "mylib.h"
#include <stdio.h>
#include <math.h>
void linspace(float start, float stop, float step, float* x_values, float* y_values, int num_values, float(*func)(float)){
for(int i = 0; i<num_values; ++i){
 x_values[i] = start + i * step;
 y_values[i] = func(x_values[i]);
}}
void save(int x, float* x_values, float* y_values){
//Save data to a file
     FILE* file = fopen("output.dat", "w");

    if (file != NULL) {
        for (int i = 0; i < x; ++i) {
            fprintf(file, "%f %f\n", x_values[i], y_values[i]);
        }

        fclose(file);
        printf("Data saved to 'output.dat'.\n");
    } else {
        printf("Error opening file for writing.\n");
    }
}
void app(int x, float* x_values, float* y_values){
//Save data to a file
     FILE* file = fopen("output.dat", "a");

    if (file != NULL) {
        for (int i = 0; i < x; ++i) {
            fprintf(file, "%f %f\n", x_values[i], y_values[i]);
        }

        fclose(file);
        printf("Data saved to 'output.dat'.\n");
    } else {
        printf("Error opening file for writing.\n");
    }
}
```

Python code

```python
import numpy as np
import matplotlib.pyplot as plt

# Load data from the "output.dat" file using numpy's loadtxt
data = np.loadtxt("output.dat")

# Extract n_values and y_values from the data
x_values = data[:401, 0].astype(float)
y_values = data[:401, 1].astype(float)

# Plot
plt.plot(x_values, y_values)
plt.xlabel(r'$t$')
plt.ylabel(r'$-3\sin{{\pi}t}$')
plt.grid(True)
plt.savefig('../figs/fig1.png')
```

Python code

```python
import numpy as np
import matplotlib.pyplot as plt

# Load data from the "output.dat" file using numpy's loadtxt
data = np.loadtxt("output.dat")

# Extract n_values and y_values from the data
x_values = data[402:, 0].astype(float)
y_values = data[402:, 1].astype(float)

# Plot
plt.plot(x_values, y_values)
plt.xlabel(r'$t$')
plt.ylabel(r'$-2\cos{\frac{\pi}{2}t}$')
plt.grid(True)
plt.savefig('../figs/fig2.png')
```