TM Homework 2

Timur Islamov

Feb 2024

Link back to GitHub

1 Task 1 (Coding)

1.1 Task description

You should find:

1. simulate this mechanism (obtain all positions.)
2. velocities and accelerations for A, B, C, E, F, D.
3. draw plots: speed and acceleration B, C, F, E; angular speed and angular acceleration CD, CO_2, DO_3.

Needed variables: $\omega_{O_1A} = 2rad/s$; $\phi = 130^0; a = 31; b = 30; c = 50$; $O_1A = 15; O_2B = 30; O_3D = 50; AB = 40; BC = 16$; $CD = 60; CE = 30; EF = 30$.

1.2 Solution

Here is the link for simulation (.ipynb file is in the folder) SIMULATION

First of all let's find the coordinates of all points. We can simply find them using the formulas as intersection of two circles or just equation for midpoint (in case of point E).

From school we know the coordinates of point on the circumference:

$$A = \begin{bmatrix} O_1A\cos(\phi) \\ O_1A\sin(\phi) \end{bmatrix}$$

From this equation we can find coordinates for point B

$$\begin{cases} (x_B - x_A)^2 + (y_B - y_A)^2 = AB^2 \\ (x_B - x_{O_2})^2 + (y_B - y_{O_2})^2 = BO_2^2 \end{cases}$$

But since from this we get two solutions, therefore, we need to check which solution we need. We should just check if $B_{x1} < B_{x2}$,and if that is true, then we get the left point by X coordinate.

Then for point C is the same checking $C_{x1} < C_{x2}$ and the system from which we get the solution:

$$\begin{cases} (x_B - x_C)^2 + (y_B - y_C)^2 = AB^2 \\ (x_C - x_{O_2})^2 + (y_C - y_{O_2})^2 = BO_2^2 \end{cases}$$

Now for point D. We get the correct coordinate by choosing the lowest y-coordinate. And here is the system:

$$\begin{cases} (x_D - x_{O3})^2 + (y_D - y_{O3})^2 = AB^2 \\ (x_D - x_C)^2 + (y_D - y_C)^2 = CD^2 \end{cases}$$

Here $x_{O_3} = O_1F + O_3F$

And the easiest $(E - easiest)$ point is point E that is just the midpoint of rod CD:

$$E = \begin{bmatrix} \frac{x_C + x_D}{2} \\ \frac{y_C + y_D}{2} \end{bmatrix}$$

For point F we need to find only y-coordinate, since it just slides along the $x = 31$ line.

$$\begin{cases} (x_E - x_F)^2 + (y_E - y_F)^2 = AB^2 \end{cases}$$

Here we know that $x_F = O_1F = 31$

Now let's find velocities for points A - F:

Using pen and paper we can find velocities for all points like this:

$$\vec{v_A}(t) = \dot{\vec{r_A}}(t) = \omega \cdot \vec{r_A}(t)$$

where $\vec{r_A}(t)$ is the coordinates of point A in dependence of t.

Just substitute point from B-F instead of A and you'll get equations for all these points.

But, since we need to code it, then we use another approach.

```
point_positions['A'].append((x_A, y_A))
```

While animating we add all points A-F to this list for points and then we use the following formula:

```
vel_x_B.append((point_positions["B"][t][0] - point_positions["B"][t - 1][0])
/ (t_values[t] - t_values[t - 1]))
vel_y_B.append((point_positions["B"][t][1] - point_positions["B"][t - 1][1])
/ (t_values[t] - t_values[t - 1]))
vel_B.append(np.sqrt(np.float_(vel_x_B[t - 1])**2 +
np.float_(vel_y_B[t - 1])**2))
```

And so for each point. Let's rewrite it in more convenient way.

$$\vec{V_B} = \begin{bmatrix} \frac{X_B(t+\Delta t)-X_B(t)}{\Delta t} \\ \frac{Y_B(t+\Delta t)-Y_B(t)}{\Delta t} \end{bmatrix}$$

Just substitute point from C-F instead of B and you'll get equations for all these points.

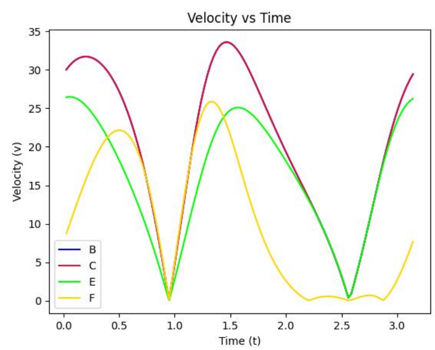
Let me explain the code part. We fill the point positions with the coordinates of points and then while plotting we use

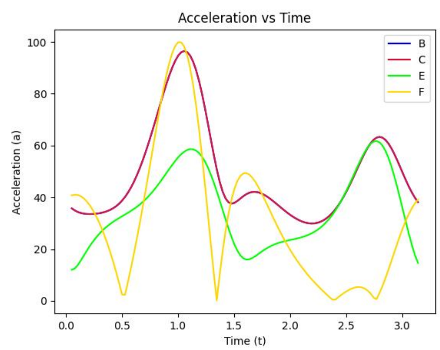
```
t_values = np.linspace(0, np.pi, num_frames)
for t in range(1, len(point_positions['B'])):
```

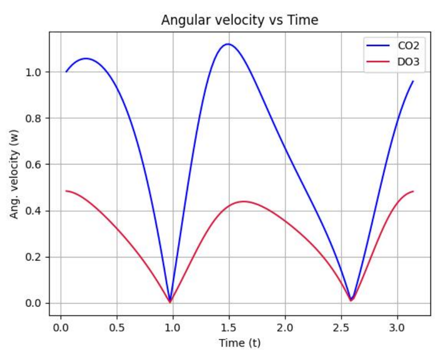
So, t is the number in that range and we get the coordinate of point and subtract from it the coordinates of point at the distance Δt and divide it by Δt

1.3 Plots

Since points B and C belongs to the same solid rod and rotates around O_3 at the same distance, therefore their velocities and accelerations are the same.







1.4 Answers

Since there are too much answers, therefore HIGHLIGHTED ANSWERS ARE IN THE SOLUTION

2 MEME


Кто сказал, что теормех это стресс?
Мне 19 и я чувствую себя замечательно!
meme-arsenal.ru