

گزارش سوال ۲ :

این سوال را به این صورت فرمول های ϕ_{1_dot} و ϕ_{2_dot} را بدست آوردیم :
و داخل پایتون هم کد معادلش را زدیم :

در عکس کد صفحه ی بعد هم آمدیم در این تابع به ازای مورد ۱ و مورد ۲ مقدار دهی کردیم و چون خروجی مون به صورت لیست است صفرمین خانه را به چرخ چپ و یکمین خانه را به چرخ راست دادیم و دو متغیر به صورت لیست برای مکان در x و مکان در y تعریف کردیم در داخل `while` هم تا زمان مشخص میایم مکان های x و y ای ربات را با استفاده از `gps` استخراج می کنیم و در دو متغیری که تعریف کردیم اضافه می کنیم در نهایت با استفاده از کتابخانه `matplotlib` این دو متغیر را رسم می کنیم تا به مسیر حرکت ربات در دستگاه $x y$ برسیم. عکس و ویدیو برای هر دو مورد ۱ و ۲ داخل `zip` سوال ۲ همراه این گزارش قرار دارد.

The image shows a handwritten derivation for the inverse kinematics of a two-link robot arm. The derivation starts with the Jacobian matrix equation:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} r\dot{\phi}_1 + r\dot{\phi}_2 \\ r\dot{\phi}_1 - r\dot{\phi}_2 \end{pmatrix}$$

Then, it isolates \dot{x} and \dot{y} to get two equations:

$$\begin{aligned} \dot{x} &= \frac{r\dot{\phi}_1 + r\dot{\phi}_2}{2} \\ \dot{y} &= \frac{r\dot{\phi}_1 - r\dot{\phi}_2}{d} \end{aligned}$$

These are labeled as (1) and (2). Then, equations (1) and (2) are combined to solve for $\dot{\phi}_1$ and $\dot{\phi}_2$:

$$\begin{aligned} \dot{\phi}_1 &= \frac{2\dot{x} + \dot{y}d}{2r} \\ \dot{\phi}_2 &= \frac{2\dot{x} - \dot{y}d}{2r} \end{aligned}$$

The final equations for $\dot{\phi}_1$ and $\dot{\phi}_2$ are circled in red.

```
# question 2 function:
```

```
def inverse_kinematic(x_dot, teta_dot, d, r):  
    phi1_dot = ( (2 * x_dot) - (teta_dot * d) ) / (2 * r)  
    phi2_dot = ( (2 * x_dot) + (teta_dot * d) ) / (2 * r)  
    return [phi1_dot, phi2_dot]
```

```

pos_x = []
pos_y = []
c = 0
t0 = robot.getTime()
# 1:
leftMotor.setVelocity(inverse_kinematic(5, 0, 1, 2)[0])
rightMotor.setVelocity(inverse_kinematic(5, 0, 1, 2)[1])
# 2:
# leftMotor.setVelocity(inverse_kinematic(0, 3, 1, 2)[0])
# rightMotor.setVelocity(inverse_kinematic(0, 3, 1, 2)[1])

while robot.step(TIME_STEP) != -1:
    temp = gps.getValues()
    pos_x.append(round(temp[0], 3))
    pos_y.append(round(temp[2], 3))
    c = c + 1
    t1 = robot.getTime()
    if t1 - t0 > 13:
        break;
plt.plot(pos_x, pos_y)
plt.show()

```