



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
مکاترونیک

گزارش مینی پروژه اول

| | |
|--------------------|---------------|
| نام و نام خانوادگی | کامیار رحمانی |
| شماره دانشجویی | ۸۱۰۱۹۹۴۲۲ |
| تاریخ ارسال گزارش | ۱۴۰۲/۱/۲۰ |

فهرست گزارش سوالات

- ۳.....MPU6050 بخش اول : کار با سنسور
- ۳.....roll,pitch,yaw محاسبه پارامترهای طبیعی : e و ϕ از روی زوایای
- ۵.....Quaternions محاسبه پارامترهای طبیعی : e و ϕ از روی
- ۷..... q, q_0 محاسبه پارامترهای خطی :
- ۸..... q, q_0 محاسبه ماتریس دوران از روی پارامترهای خطی :
- ۹.....Vpython بخش دوم : استفاده از کتابخانه

بخش اول : کار با سنسور MPU6050

محاسبه پارامترهای طبیعی : e و ϕ از روی زوایای roll, pitch, yaw

ابتدا تابعی پیاده سازی کردم که پارامترهای طبیعی را از روی زوایای roll و pitch و yaw محاسبه کند.

سنسور MPU6050 زوایای گفته شده را در اختیار ما قرار می دهد و من ارائه شناخته شده XYZ را در دستگاه مختصات ثابت یا Global بدست آوردم :

$$Q = Q_z * Q_y * Q_x$$

همچنین از درس هر کدام از ماتریس های دوران بالا را می دانیم :

$$Q_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad Q_y = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix}$$

$$Q_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

که زاویه roll همان زاویه دوران حول محور x و زاویه pitch زاویه دوران حول محور y و زاویه yaw زاویه دوران حول محور z است.

سپس از روی ماتریس Q پارامترهای طبیعی را استخراج کردم :

$$\text{vect}(\mathbf{A}) \equiv \mathbf{a} \equiv \frac{1}{2} \begin{bmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{bmatrix}, \quad \text{tr}(\mathbf{A}) \equiv a_{11} + a_{22} + a_{33} \quad \cos \phi = \frac{\text{tr}(\mathbf{Q}) - 1}{2}$$

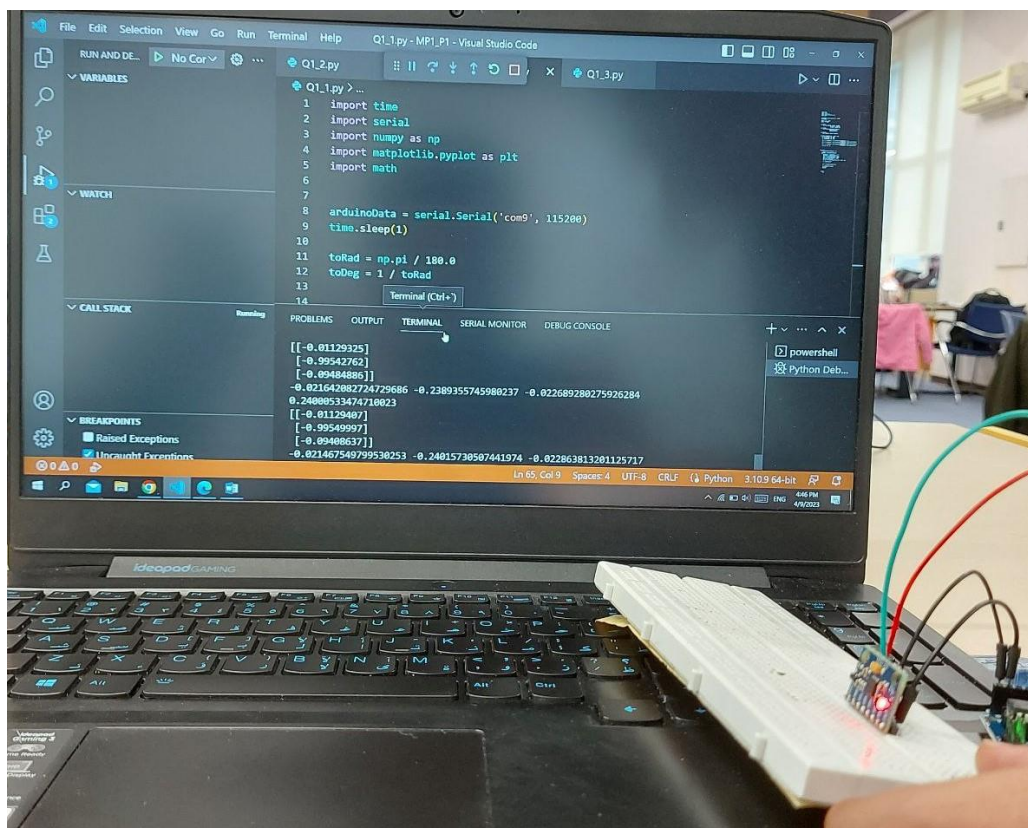
$$\text{vect}(\mathbf{Q}) = \text{vect}(\sin \phi \mathbf{E}) = \sin \phi \mathbf{e}$$

خروجی:

در شکل زیر خروجی را برای یک دوران خاص مشاهده می کنیم :

در شکل زیر خروجی ها به ترتیب زیر است:(یادم رفت اسامی رو پرینت کنم!!)

- زوایای دوران yaw, pitch, roll بر حسب رادیان
- زاویه دوران phi بر حسب رادیان
- راستای دوران



توقع داریم محور دوران، بردار یکه شود که همانطور که می بینیم که همینطور است.

محاسبه پارامترهای طبیعی: e و ϕ از روی Quaternions

طبق درس میدانیم که quaternion همان پارامترهای اویلر رودریگز هستند:

$$\underline{\vec{r}} = \sin\left(\frac{\phi}{2}\right) \underline{\vec{e}} \quad \text{و} \quad \underline{r_0} = \cos\left(\frac{\phi}{2}\right)$$

خروجی کد برداری به صورت زیر خواهد بود که از روی درایه آخر زاویه دوران را بدست می آوریم و سپس از روی سه درایه اول و زاویه دوران محور دوران را نیز بدست می آوریم.

$$\vec{\lambda} = \begin{bmatrix} r \\ r_0 \end{bmatrix}$$

برای اینکه در کد پایتون Quaternion ها در خروجی نمایش داده شوند بخش زیر را به کد اضافه

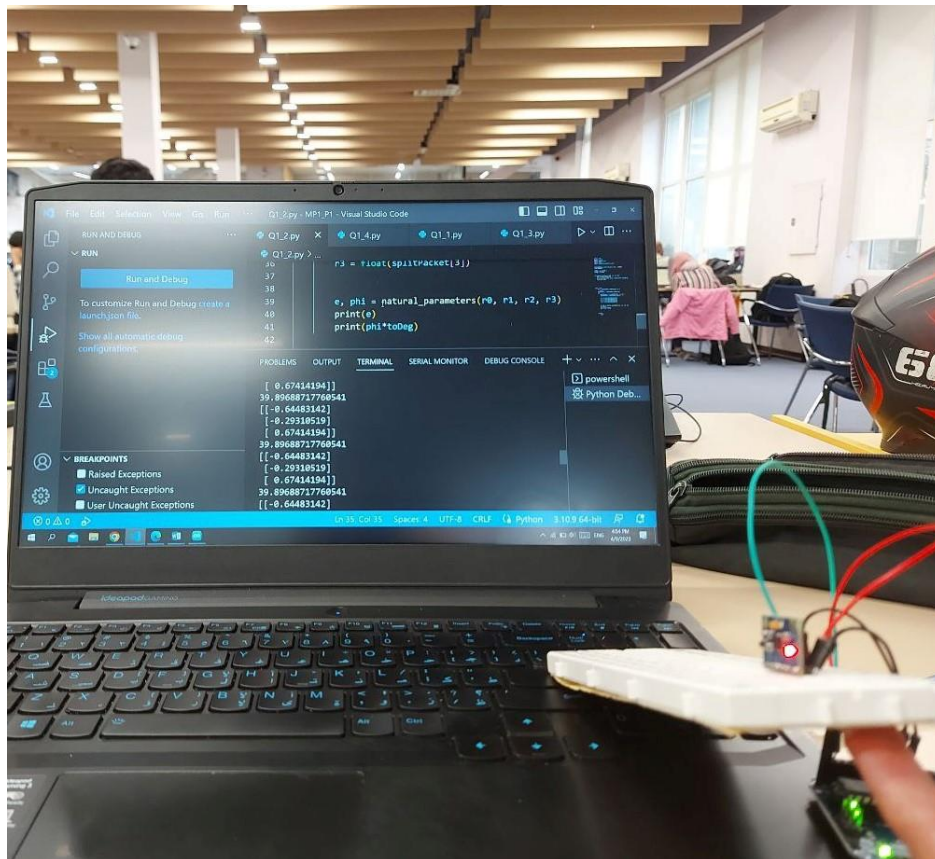
کردم:

```
r0 = float(splitPacket[0])
r1 = float(splitPacket[1])
r2 = float(splitPacket[2])
r3 = float(splitPacket[3])
```

خروجی :

در شکل زیر خروجی ها به ترتیب زیر است:(بادم رفت اسامی رو پرینت کنم!!)

- محور دوران e
- زاویه دوران ϕ بر حسب درجه



همانطور که می بینیم در این جا نیز محور دوران برداری یکه شده است.

محاسبه پارامترهای خطی : q, q_0

در درس آموختیم که :

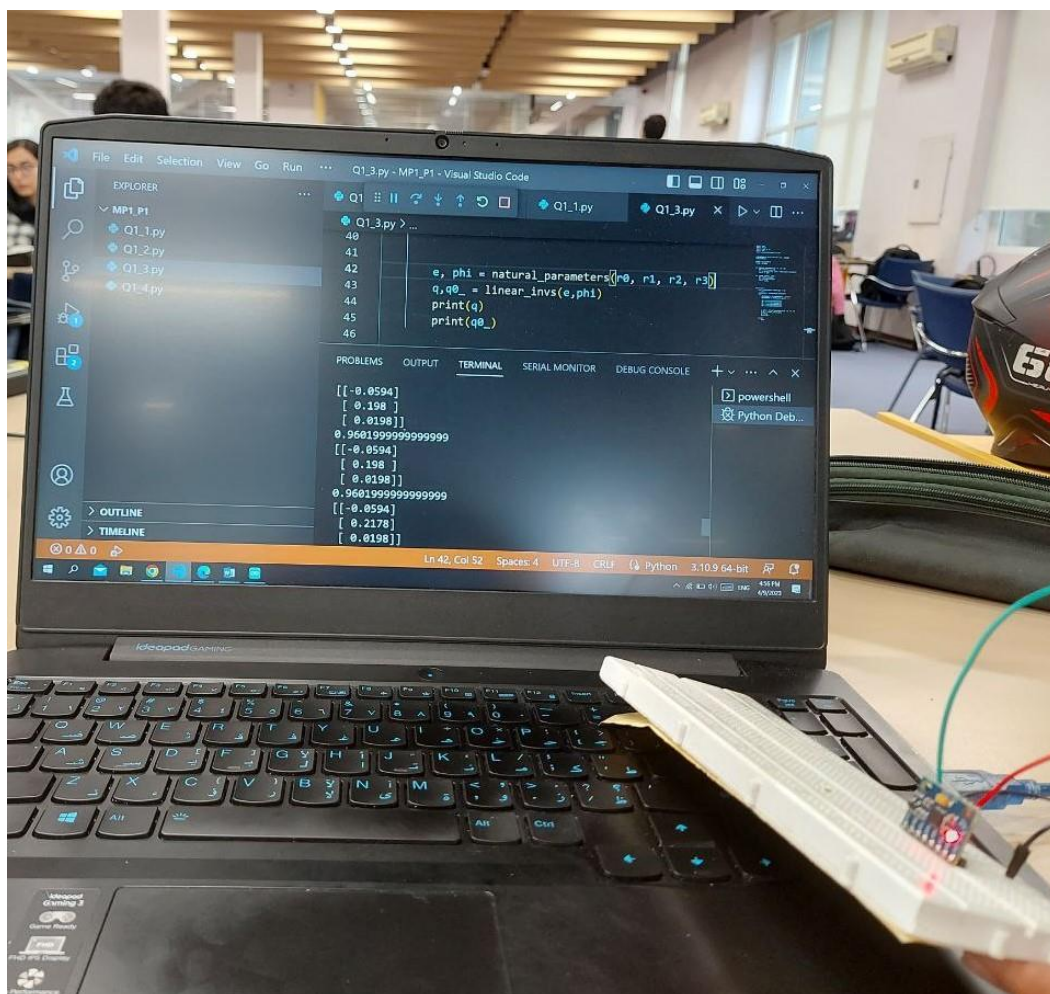
$$q_0 = \cos(\varphi)$$

$$\vec{q} = \vec{e} \sin(\varphi)$$

با استفاده از پارامترهای طبیعی که در قسمت قبل بدست آوردیم پارامترهای خطی را محاسبه می کنیم:

در شکل زیر ترتیب خروجی ها بدین ترتیب است :

- بردار q
- q_0



محاسبه ماتریس دوران از روی پارامترهای خطی: q, q_0

در درس دیدیم که با استفاده از رابطه زیر می توانیم از روی پارامترهای خطی ماتریس دوران را محاسبه کنیم

$$Q = q_0 \frac{1}{2} + \bar{Q} + \frac{\vec{q} \vec{q}^T}{1+q_0}$$

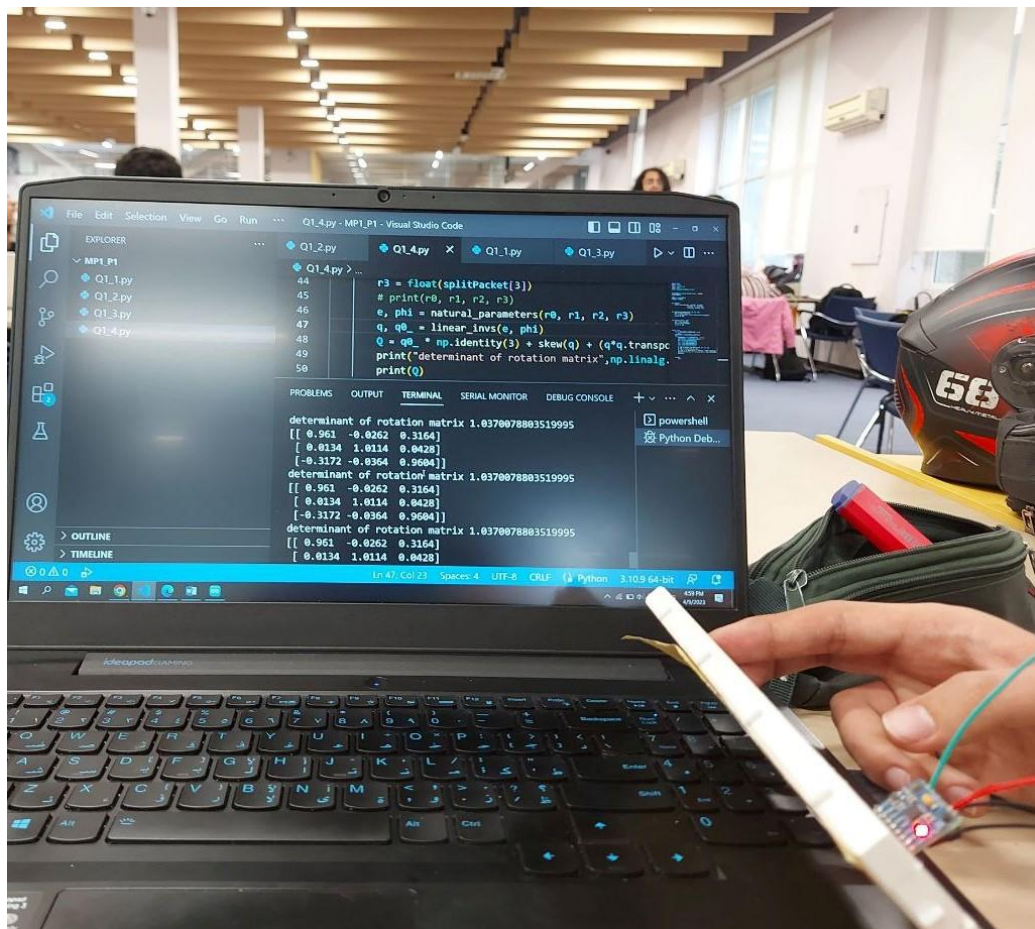
ماتریس دوران را با پارامترهای خطی q, q_0

در فرمول بالا \bar{Q} همان $CPM(q)$ است.

خروجی :

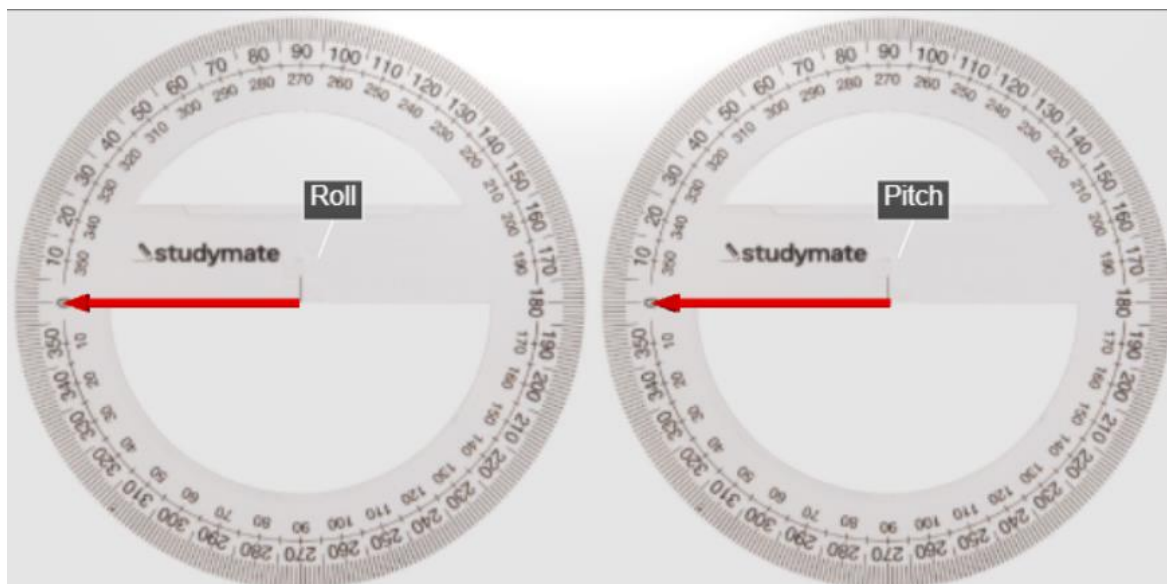
در شکل زیر ماتریس دوران و هم چنین دترمینان آن را با توجه به فرمول بالا حساب کردم.

همانطور که در شکل می بینیم دترمینان ماتریس دوران با تقریب خوبی برابر یک شده است و این به منزله کالیبره بودن سنسور است.



بخش دوم : استفاده از کتابخانه Vpython

در ابتدا با استفاده از ابزار های box,arrow و همچنین عکسی از نقاله که از اینترنت گرفتم دو زاویه سنج را طراحی کردم :



نتیجه پس از شبیه سازی :

