

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر
اصول علم ربات

تمرین سری

نام و نام خانوادگی	سید عارف طباطبایی
شماره دانشجویی	9831040
تاریخ ارسال گزارش	11 خرداد 1402

فهرست گزارش سوالات (لطفاً پس از تکمیل گزارش، این فهرست را به روز کنید.)

سوال 1 – گام 1 3

سوال ۲ – گام 2 8

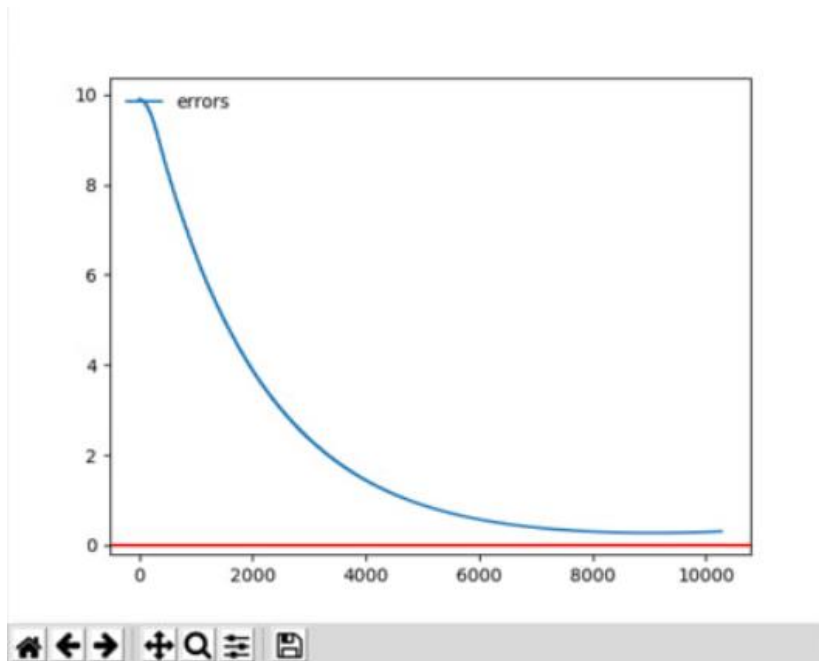
سوال 3 – گام 3 12

سوال 1 – گام 1

بخش 1:

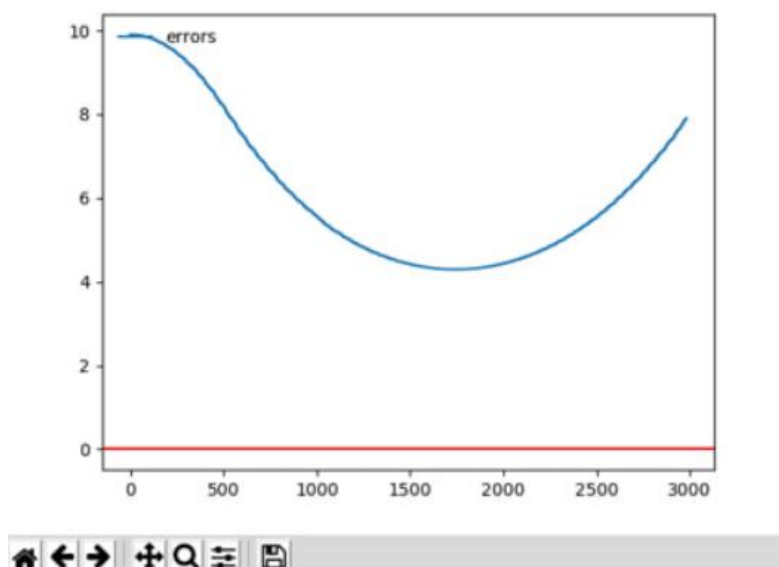
برای تنظیم مقادیر PID به صورت زیر عمل می‌کنیم:

برای ضریب $k-p$ مقدار 0.1 مناسب بود و نمودار زیر مشاهده شد:

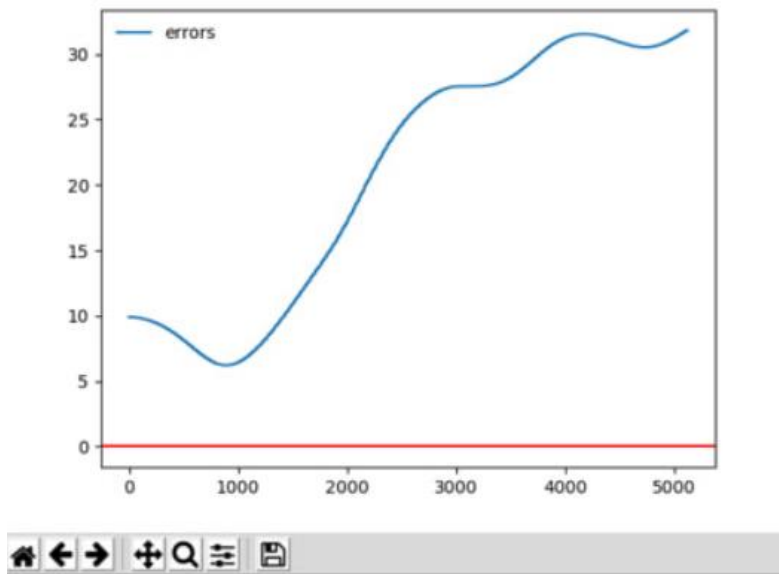


این ضریب در مقادیر بزرگ‌تر، باعث نوسان در نمودار می‌شد:

برای مقدار 0.2 داریم:



و برای مقدار 0.5 داریم:



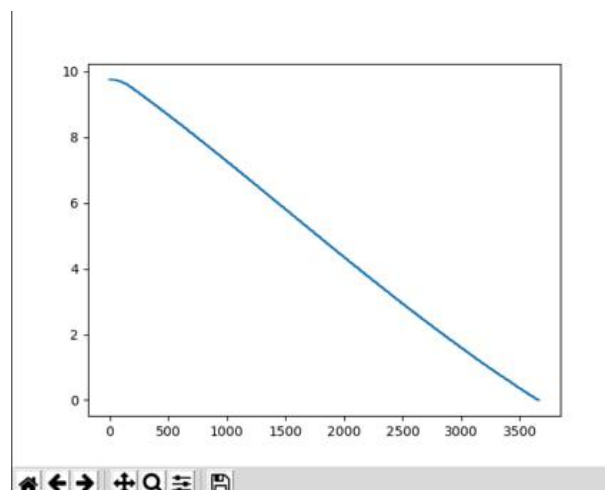
ضریب $k-d$ را افزایش داده شد تا نوسان کم شود اما اختلاف زیادی با مقصد داشتیم. از طرفی به ازای $k-p$ با مقدار 0.1، با کم شدن ارور (یا همان فاصله از مقصد) سرعت خطی نیز کاهش می‌یابد و مشاهده می‌شود که ربات انتهای مسیر با سرعت خیلی کمی به سمت مقصد در حال حرکت است. در نتیجه مقدار ضریب $k-i$ را افزایش می‌دهیم تا با جمع خطاهای قبلی، سرعت را افزایش دهد. در نهایت به ضرایب زیر رسیدیم:

$$k_p = 0.1$$

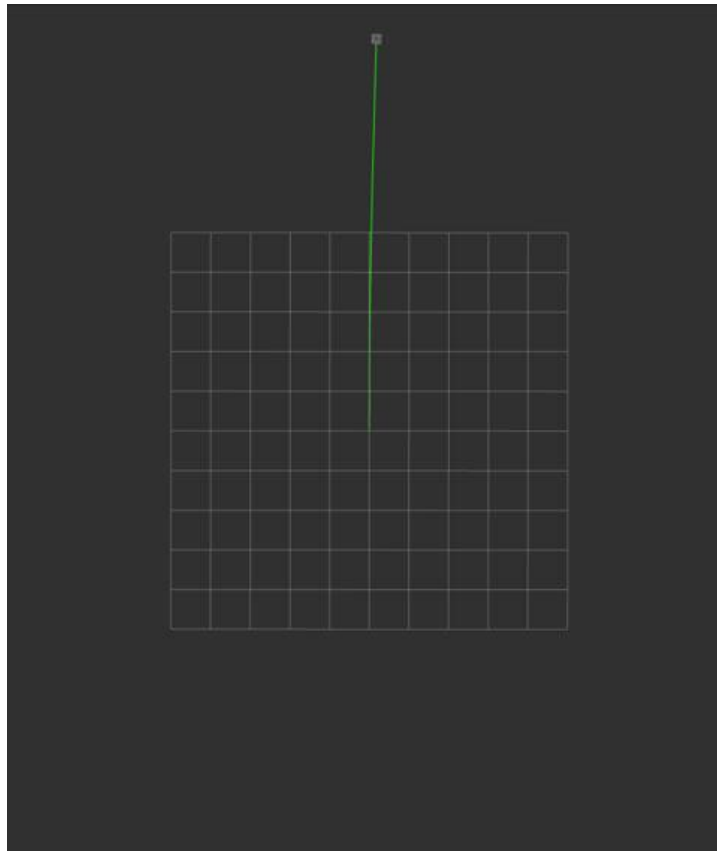
$$k_i = 0.005$$

$$k_d = 5$$

با این ضرایب، نمودار زیر مشاهده شد:

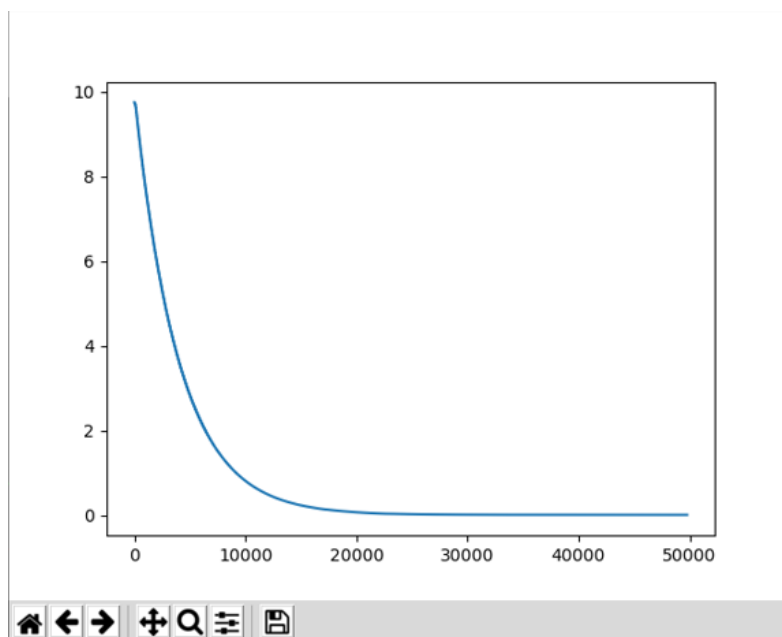


همچنین، مسیر حرکت را می‌توانید در تصویر زیر مشاهده کنید:

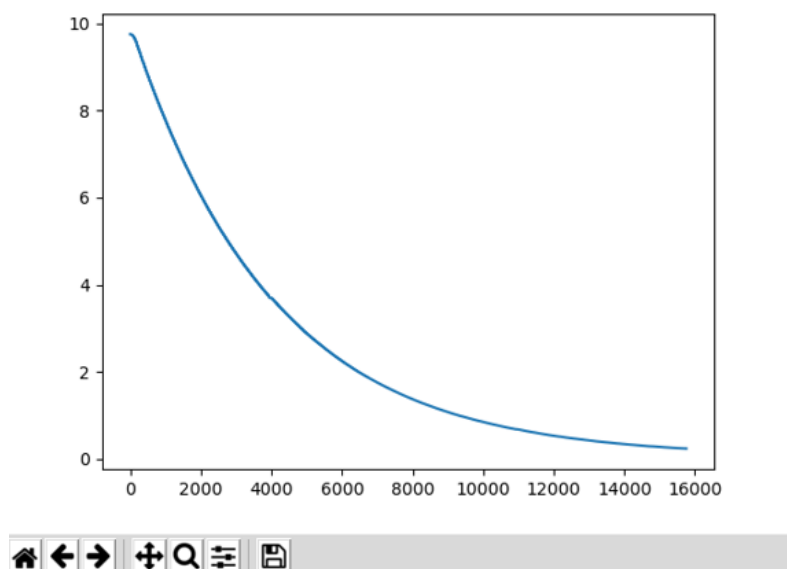


مقایسه P، PD و PID:

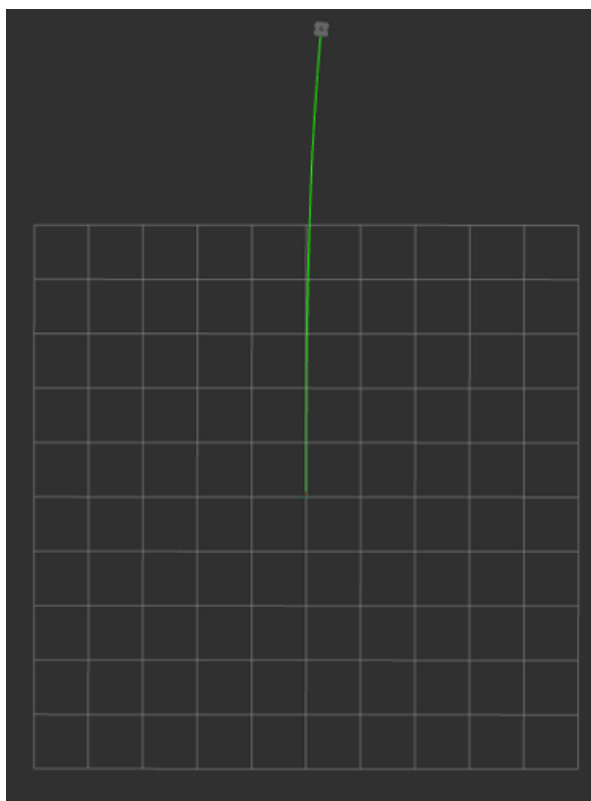
در حالت کنترلر P، سرعت خیلی کم است.



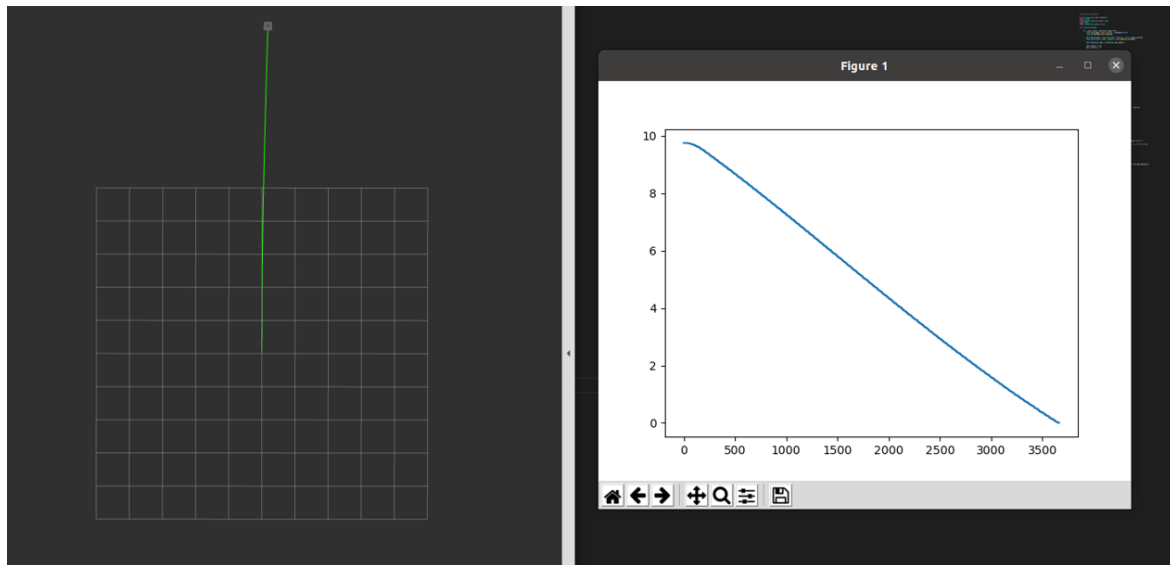
در حالت کنترلر PD، سرعت در ابتدا بیشتر می‌شود، اما باز هم انتهای مسیر سرعت کم است و همچنین ربات دچار انحراف می‌شود:



انحراف مسیر:



اما در حالت کنترلر PID داریم:



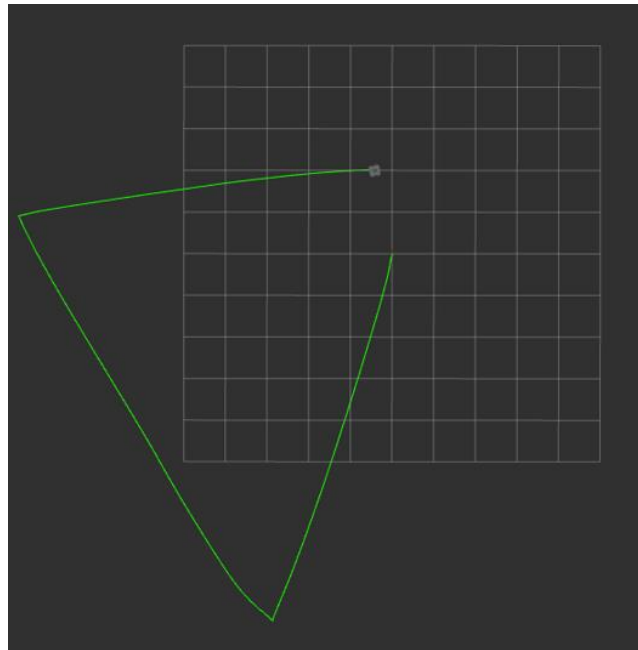
بخش 2:

برای این بخش، نیاز به اعمال کنترلر PID بر روی سرعت زاویه ای نیز داریم. از این رو، مشابه با سرعت خطی عمل می‌کنیم و به ضرایب زیر می‌رسیم:

```
# angular velocity
self.kp_a = 0.1
self.ki_a = 0
self.kd_a = 5
self.Iz = 0
self.previous_a_error = 0.0

# linear velocity
self.kp_l = 0.1
self.ki_l = 0.005
self.kd_l = 1
self.Ix = 0
self.previous_l_error = 0.0
```

نمودار نهایی:



سوال ۲ – گام 2

نحوه تنظیم آرایه از مختصات برای هر شکل:

```
def rectangle(self):
```

```
    X1 = 0
```

```
    Y1 = 0
```

```
    Y2 = 0
```

```
    X2 = X1 + self.width
```

```
    X3 = X1 + self.width
```

```
    Y3 = Y1 + self.height
```

```
    Y4 = Y1 + self.height
```

```
    X4 = 0
```

```
    return ([X1, X2, X3, X4]), ([Y1, Y2, Y3, Y4])
```

```
def star(self):
```

```
    X = [0, 1.5, 0.25, 0.75, -0.5, -1.75, -1.25, -2.5, -1, -0.5]
```

```
    Y = [0, 0, -1, -2.5, -1.5, -2.5, -1, 0, 0, 1.5]
```

```
    return X, Y
```

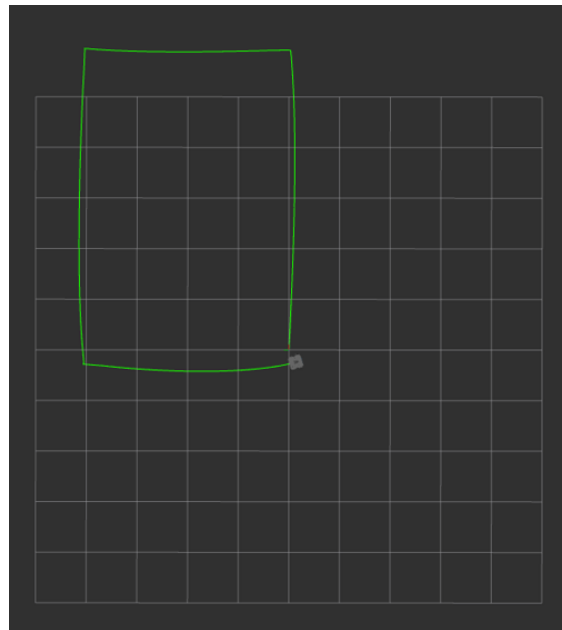


```
def spiral(self):
    a = 0.17
    k = tan(a)
    X , Y = [] , []

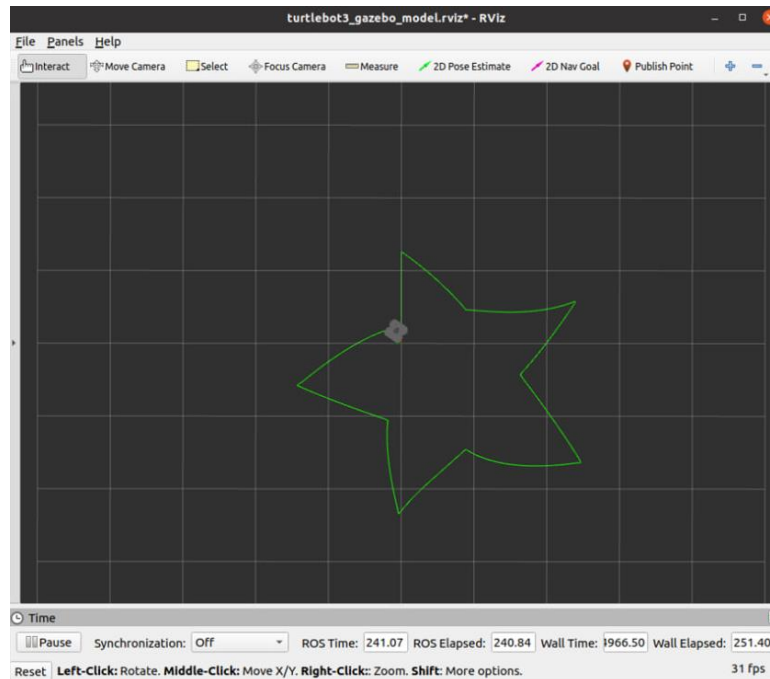
    for i in range(150):
        t = i / 20 * pi
        dx = a * exp(k * t) * cos(t)
        dy = a * exp(k * t) * sin(t)
        X.append(dx)
        Y.append(dy)
    return np.array(X), np.array(Y)
```

نمودار های رسم شده توسط Rviz برای هر شکل:

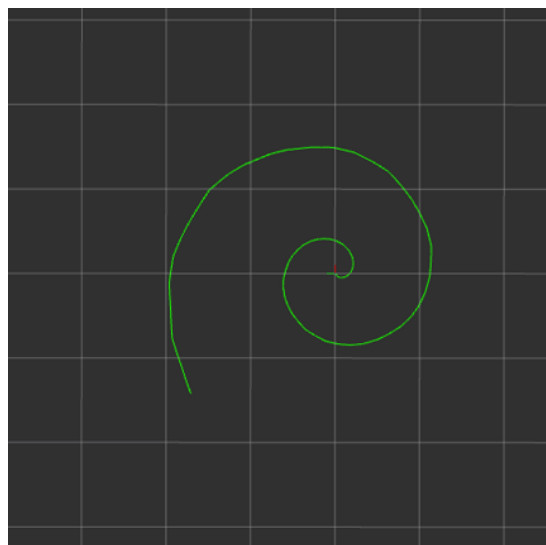
مستطیل:



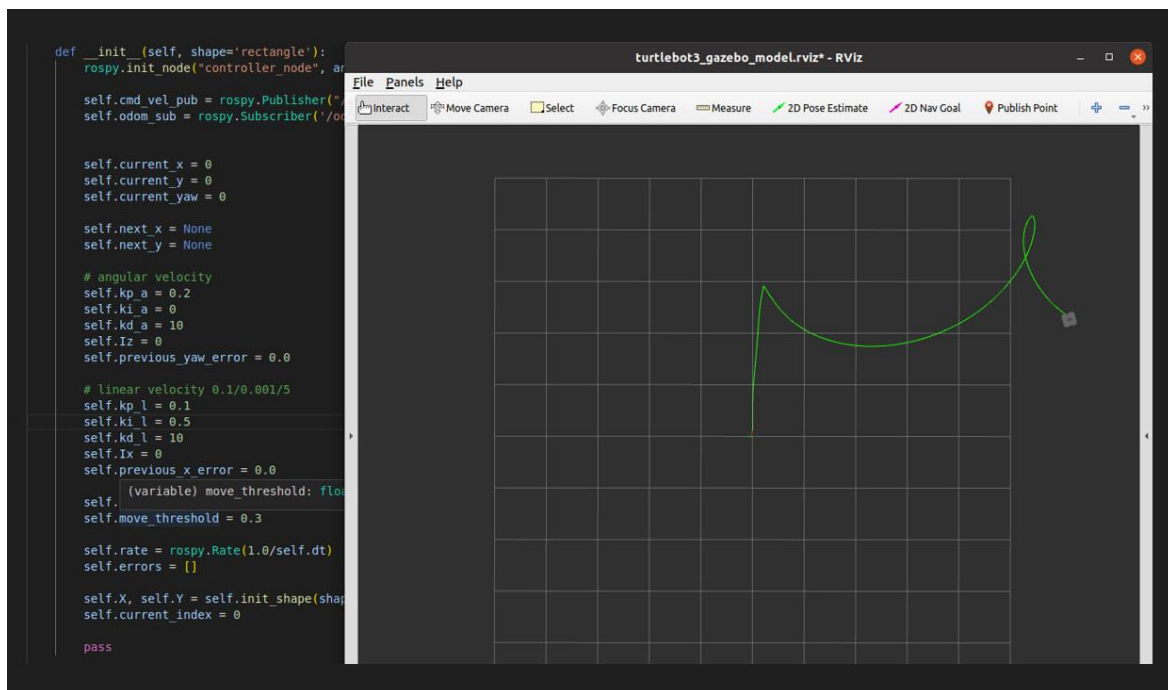
ستاره:



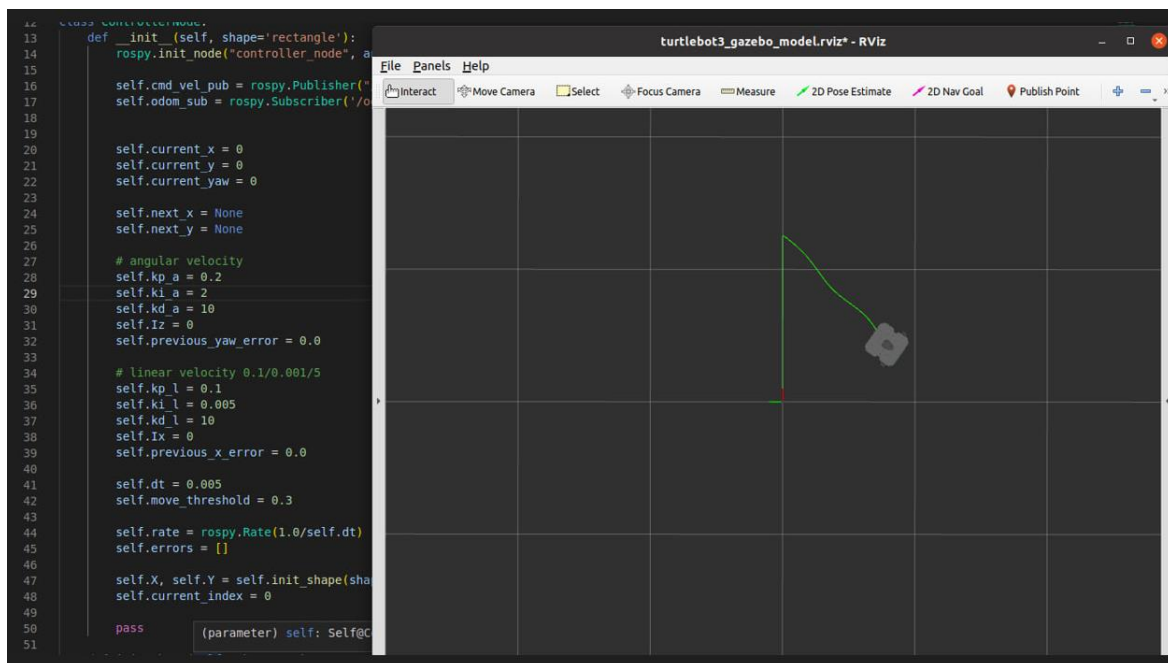
مارپیچ:



مشکلی که در این قسمت پیش آمد، این بود که سرعت حرکت ربات از سرعت رسم نمودار حرکت آن توسط Rviz بیشتر بود و از این رو ربات را در شکل بالا مشاهده نمی کنید. تغییر مقادیر ضرایب کنترلر PID را بر روی شکل ستاره بررسی می کنیم: افزایش k_i در سرعت خطی باعث می شود سرعت ربات در هنگام رسیدن به مقصد زیاد شود و ربات از کنترل خارج شود:



افزایش k_i در سرعت زاویه ای باعث می شود ربات دچار نوسان شود:



دیگر ضرایب نیز رفتار مشابهی خواهند داشت.

سوال 3 – گام 3

در این قسمت، با استفاده از پیام های سنسور LiDAR مینیمم فاصله با دیوار یافت شده و سپس به عنوان خروجی برگردانده می شود. باید توجه داشت به دلیل این که ربات ما در سمت راست دیوار است، به همین دلیل تنها نیاز است که بازه 0 تا 180 درجه در خروجی این سنسور را بررسی کرد که مربوط به ناحیه سمت چپی ربات می شود.

در ادامه بسته به میزان خطا که در اینجا همان فاصله مجاز از دیوار می باشد، سرعت زاویه ای ربات مشخص می شود. در اینجا برای چرخش بهتر ربات، در صورتی که سرعت زاویه ای ربات از حدی بیشتر شود، سرعت خطی ربات $4/1$ می شود تا ربات چرخش بهتری داشته باشد.

یکی از مشکلات کم بودن سرعت واکنش به مقدار خطا حین دوران در گوشه های دیوار (هنگام چرخش) است. برای رفع این مشکل مقدار k_i را برای سرعت زاویه ای افزایش می دهیم.

مقادیر کنترلی:

```
self.angular_k_i = 0.1  
self.angular_k_p = 0.6  
self.angular_k_d = 7
```

نمودار حرکت:

