به نام خدا

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

اصول علم ربات

تمرین سری دوم

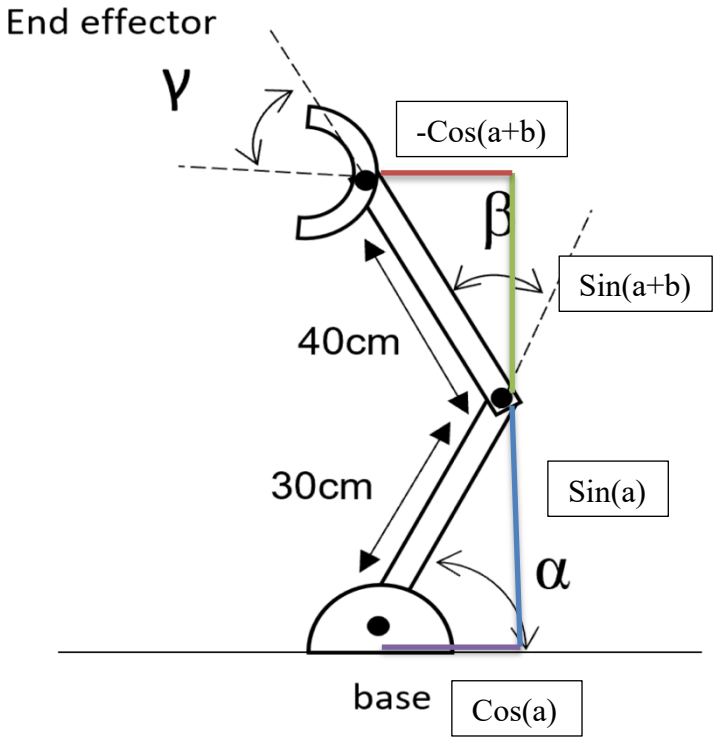
نام و نام خانوادگی	سارا تاجرنیا
شماره دانشجویی	۹۸۳۱۰۱۶
تاریخ ارسال گزارش	۱۴۰۱/۲/۱۳

فهرست گزارش سوالات

بخش تئوری

سوال 1 – مقادیر درجات

سوال اول: یک بازوی روباتی با سه درجه آزادی مطابق شکل را در نظر بگیرید. در این حالت تابع تبدیل همگن از $base$ به $effector$ را بر حسب مقادیر کنترلی α، β و γ به دست بیاورید .(بارم : 7امتیاز)



Alpha = a, beta = b, teta = t = a+b

X2 = 0.3cos(a) + 0.4cos(a+b)

Y2 = 0.3sin(a) + 0.4sin(a+b)

$$r = \begin{vmatrix} x \\ y \\ t \end{vmatrix} = \begin{bmatrix} 0.3\text{Cos}(a) + 0.4\ \text{Cos}(a+b) \\ 0.3\ \sin(a) + 0.4\ \text{Sin}(a+b) \\ a + b \end{bmatrix}$$

$$F\ (a+b) = \begin{bmatrix} \text{Cos}\ (a+b+y) & -\text{Sin}\ (a+b+y) & 0.3\text{Cos}(a) + 0.4\ \text{Cos}(a+b) \\ \text{Sin}\ (a+b+y) & \text{Cos}\ (a+b+y) & 0.3\ \sin(a) + 0.4\ \text{Sin}(a+b) \\ 0 & 0 & 1 \end{bmatrix}$$

سوال ۲ – مقادیر نقطه

سوال دوم: فرض کنید، جهت (orientation) فریم $\{b\}$ نسبت به فریم {a} با ماتریس زیر نمایش داده می‌شود:

$$R_{ab} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

و نقطه p در فریم {a} به صورت $(1, 2, 3)$ نمایش داده شود. مقادیر ان نقطه در فریم b را بیابید. (بارم : 3امتیاز)

$$\{b\} = \{a\} * R_{ab}^{-1} = (1 \quad 2 \quad 3) * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}^{-1} = (1 \quad 2 \quad 3) * -1 \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

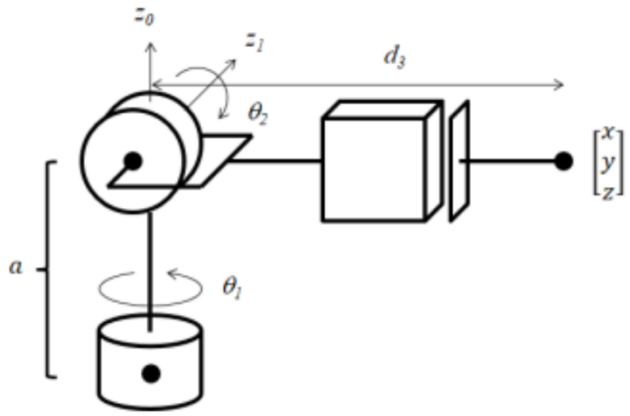
$\{b\} = (1 \quad -3 \quad 2)$

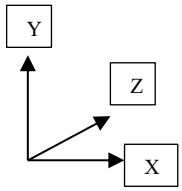
سوال 3 – kinematric

سوال سوم : شکل زیر را در نظر بگیرید، یکبار forward kinematic را برای شکل زیر و یکبار inverse kinematic آن را حل کنید. ($d_3 > 0$ رابطه نهایی را پارامتری به دست آورید.) (بارم : ۱۰ امتیاز)

نکته ۱: پایه شکل ثابت و فقط قابلیت چرخش دارد. (θ_1)

نکته ۲: مکعب موجود در شکل برای تغییر طول می‌باشد.





Forward kinematic:

X = sin(teta1) * sin(teta2) * d3

Y = a + cos(teta2) * d3

Z = cos(teta1) * sin(teta2) * d3

$$r = \begin{bmatrix} x \\ y \\ z \\ teta \end{bmatrix} = \begin{bmatrix} \sin(teta1) * \sin(teta2) * d3 \\ a + \cos(teta2) * d3 \\ \cos(teta1) * \sin(teta2) * d3 \\ teta1 + teta2 \end{bmatrix}$$

Inverse kinematic:

D3 = sqrt(a^2 + b^2)

Teta1 = arctan(a/b)

Teta2 = arctan((d3^2)/a) = arctan((a^2+b^2)/a)

بخش عملی

گام اول

بخش شبیه‌سازی

❖ شرح سناریو

در این سناریو می‌خواهیم با کمک **ROS** و مباحث کنترلی که یاد گرفته ایم، سیستمی برای ربات خود طراحی کنیم که مسیر دلخواه ما را گرفته و ربات را روی آن مسیر هدایت کند. سپس به تحلیل سیستم طراحی شده و ارزیابی درستی عملکرد آن می‌پردازیم.

❖ توضیح گام‌های انجام سناریو

گام اول (۲۰ امتیاز) : برای شروع، فرض کنید که ربات ما دو state دارد:

1. حرکت به سمت جلو با سرعت خطی ثابت
2. دوران 90 درجه به سمت چپ در حالت ایستاده (با سرعت زاویه‌ای دلخواه)

حال میخواهیم با کمک این state ها ربات را به گونه‌ای برنامه ریزی کنیم که بر روی یک مستطیل به مرکز فریم اصلی شبیه ساز حرکت کند. در این گام، ابتدا ربات باید به سمت نزدیکترین نقطه مستطیل حرکت کند و حرکت خود را روی مستطیل ادامه دهد، تا زمانی که برنامه متوقف شود. برنامه شما در این بخش باید قابلیت گرفتن سرعت خطی از ورودی (سمت کاربر) را داشته باشد (می‌توانید از تگ param در فایل ros launch خود استفاده کنید.). طول مستطیل خود را 6 و عرض مستطیل را 4 متر در نظر بگیرید. نقطه شروع ربات را بر روی مختصات (0,0) قرار دهید. سپس به ازای سرعت خطی های 0.2 متر بر ثانیه، 0.4 متر بر ثانیه و 0.8 متر بر ثانیه اجرا بگیرید و موارد زیر را در گزارش خود قرار دهید:

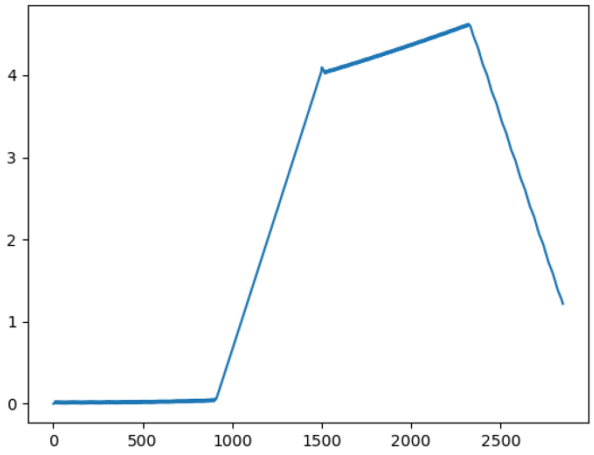
1. خطای انحراف از مسیر به ازای هر سرعت خطی
2. نمای شکل تولید شده از حرکت ربات در شبیه ساز Rviz به ازای هر سرعت خطی
3. عملکرد ربات به ازای سرعت‌های ذکر شده بررسی کنید. آیا افزایش سرعت باعث انحراف ربات شده است یا خیر؟ برای جواب بلی یا خیر خود، تحلیل ارائه دهید؟

همچنین برای محاسبه‌ی خطای انحراف از مسیر کافی است داخل یک آرایه نقاط شکل مورد نظر را نگه داشته و در هر زمانی که اطلاعات Pose خود را دریافت می کنید فاصله ربات را با نزدیکترین نقطه از مسیر محاسبه کنید. خطای انحراف را با استفاده از کتابخانه matplotlib ترسیم کنید.

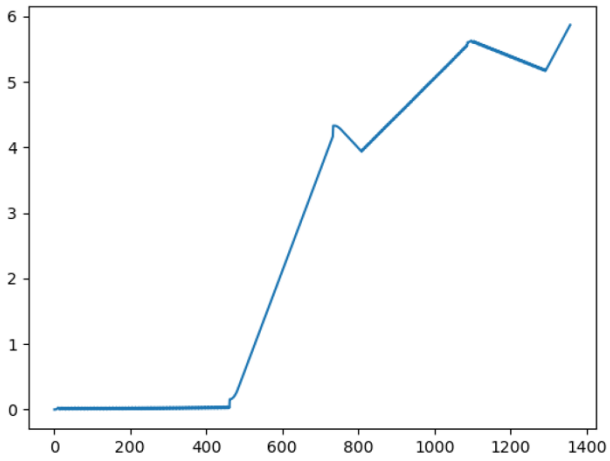
کد شکل مستطیل را از اینجا می‌توانید مشاهده کنید.

1. خطای انحراف ازمسیر به ازای هرسرعت خطی:

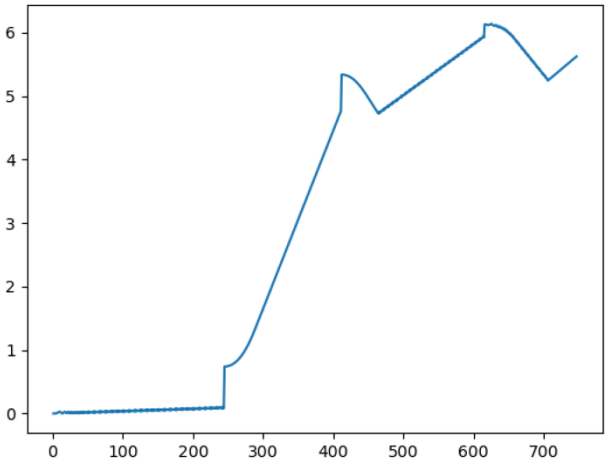
:0.2



:0.4

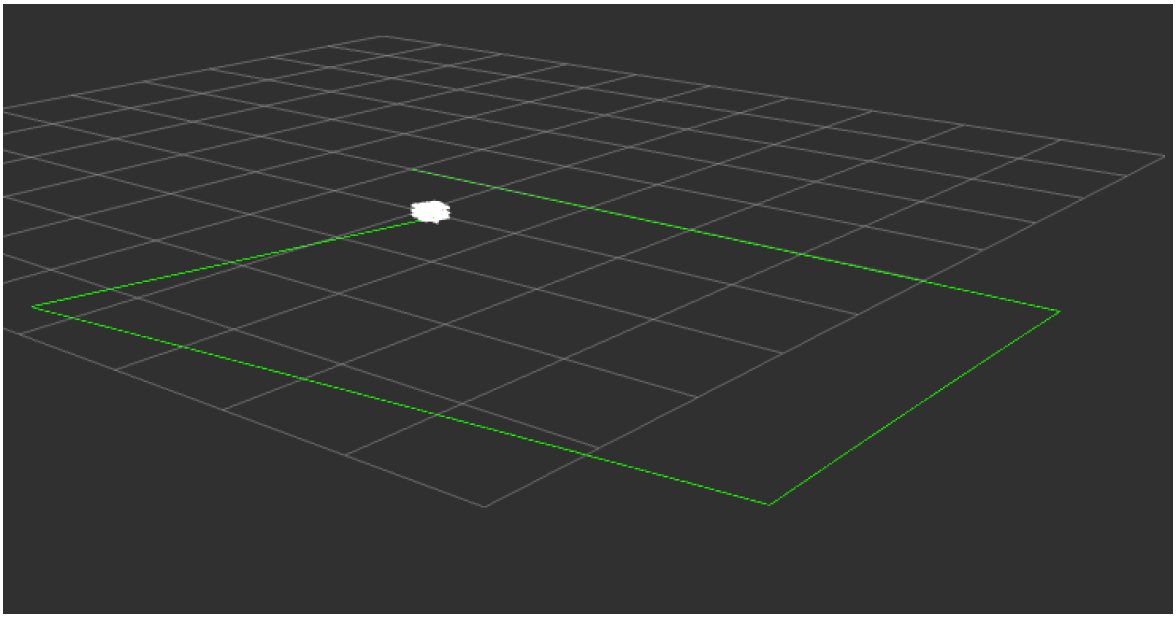


:0.8

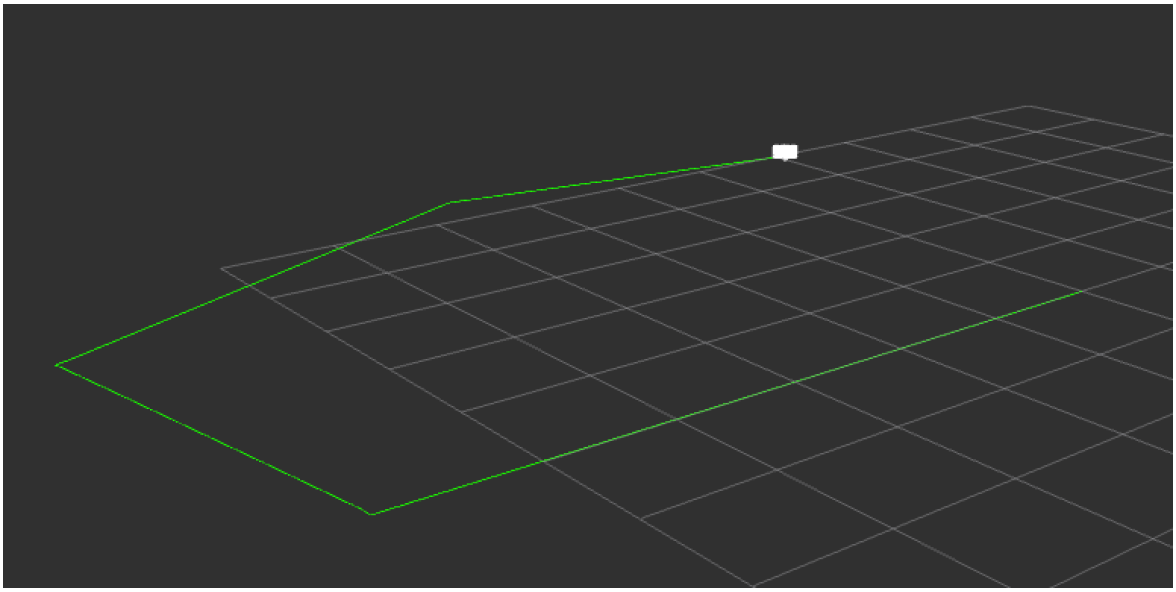


2. نمای شکل تولید شده از حرکت ربات در شبیه ساز Rviz به ازای هر سرعت خطی:

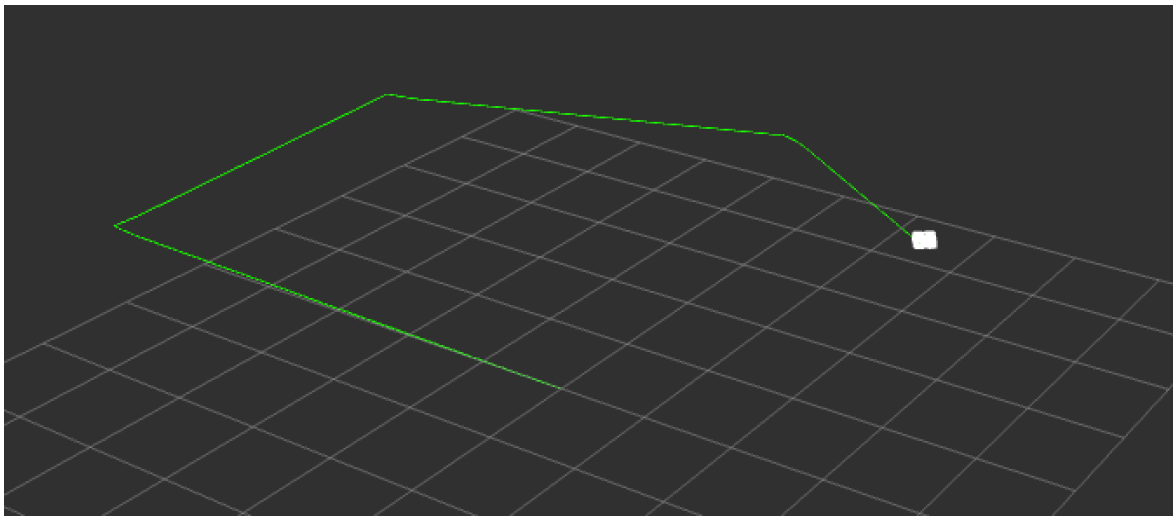
:0.2



:0.4



:0.8



3. به ازای افزایش سرعت خطی در ربات، ربات از مسیر اصلی خود منحرف میشود به طوری که در 0.8 گویا نمیتواند مستطیل بکشد دلیل این امر این است که در سرعت های بالاتر به نسبت سرعت ربات بیشتر طول میکشد تا توقف کامل کند به همین دلیل زاویه چرخش ربات خطای بالایی دارد و مستطیل تشکیل نمیدهد.

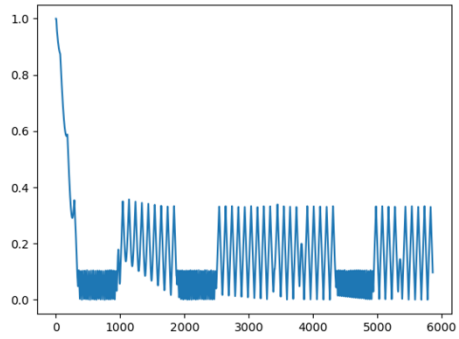
گام دوم

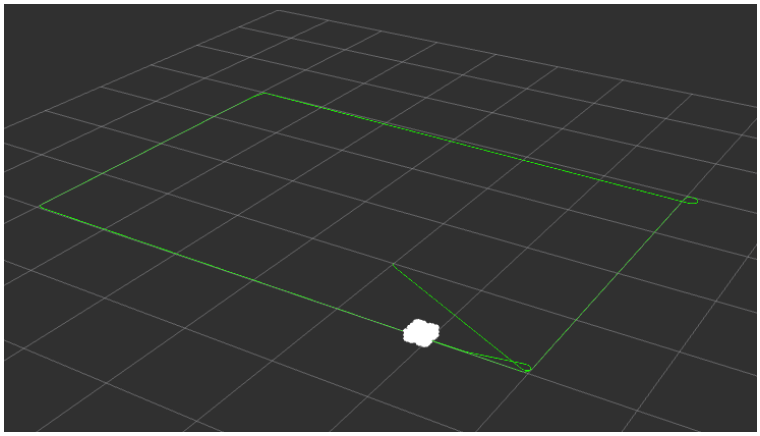
گام دوم (۴۰ امتیاز) : حال می‌خواهیم با روش‌های کنترلی، کنترل ربات را برای حرکت بر روی مسیر دلخواه خود، به دست بگیریم. برای این کار از کنترلر PID استفاده می‌کنیم. با PID در کلاس درس آشنا شده‌اید و اکنون سعی داریم در عمل از آن استفاده کنیم تا کنترل بهتری بر روی مسیر ربات خود داشته باشیم. این کنترلر باید به گونه‌ای طراحی شود که هر مسیر دلخواهی که در قالب یک آرایه ای از مختصات متوالی تعریف می‌شود را بتواند دنبال کند. مراحل زیر را برای پاسخ به این سوال دنبال کنید:

1. نمای شکل تولید شده از حرکت ربات در شبیه ساز Rviz را نشان دهید.
2. تشریح کنید که چه ضرایبی برای P، I و D پاسخ مناسبی را ارائه می‌کند. (ضرایب مناسب را به صورت تجربی به دست آورید)
3. درباره تاثیرات افزایش و کاهش هر کدام از ضرایب بحث کنید و چند نمونه آن را تشریح کنید. (برای این قسمت تحلیل کاهش و افزایش هر سه ضریب در گزارش نیاز است ولی برای ارزیابی حداقل دو مورد از تاثیرهای فوق را در شبیه‌ساز مستدل کنید کافی می‌باشد. این تاثیرها می‌توانند شامل سرعت ربات در نیل به مسیر هدف در صورت انحراف، خطای انحراف از مسیر، نوسان ربات در طی مسیر و ... باشند.)

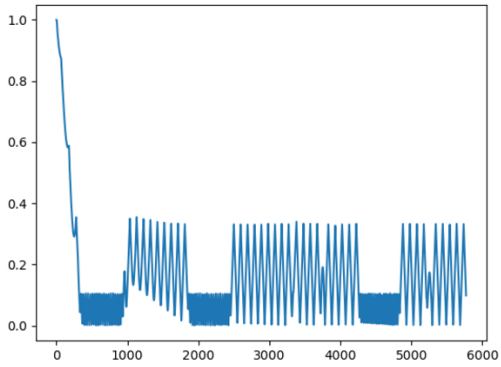
این گام را برای شکل توصیف شده در گام اول باید انجام دهید. نقطه شروع ربات را بر روی مختصات (1,1) قرار دهید.

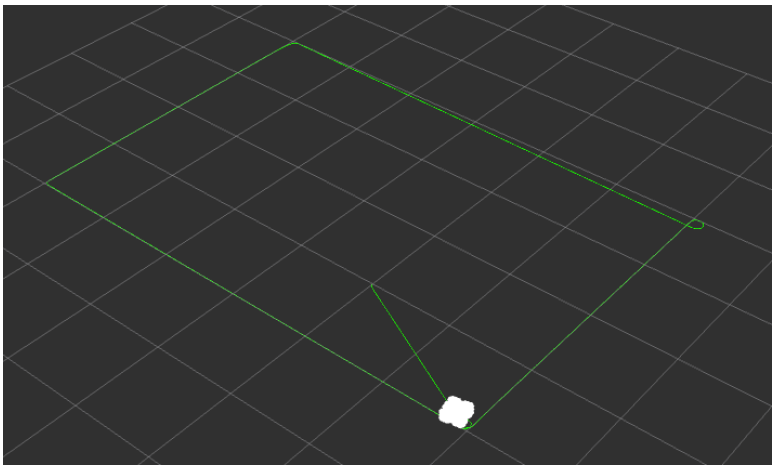
```
self.kp_distance = 10
self.ki_distance = 0.0
self.kd_distance = 0.5

self.kp_angle = 3
self.ki_angle = 0.03
self.kd_angle = 0.05
```





```
self.kp_distance = 15
self.ki_distance = 0.000001
self.kd_distance = 0.4

self.kp_angle = 3
self.ki_angle = 0.03
self.kd_angle = 0.05
```





P ⟵ تغییرات سرعت را کنترل میکند به طوری که اگر به هدف مورد نظر نزدیک باشیم سرعت را کم و اگر دور باشیم سرعت را زیاد میکند.

I ⟵ این ضریب برای کم کردن خطای به وجود آمده است به طوری که با گرفتن انتگرال در مسیرهایی که خطای بالایی دارند و چرخش زیادی در فضا دارند این خطا را از بین میبرد.

D ⟵ ضریب D به طوری به ما کمک میکند که مقدار فاصله با هدف را در نظر میگیرد و عدد آن بهتر از چیزی حدود 0.4 تا 0.6 باشد.

گام سوم

گام سوم (۲۰ امتیاز) : در این گام می‌خواهیم عملکرد کنترلر خود را برای شکل های دیگری آزمایش کنید. ربات خود را در مسیر های زیر قرار دهید:

1. مارپیچ لگاریتمی با $a = 0.17$
2. ترکیب دو نیم دایره، که برای کد آن به نکته دوم مراجعه کنید.
3. مارپیچ ارشمیدسی (Archimedean Spirals) با growth factor برابر 0.1
4. هشت ضلعی منتظم به ضلع دو

مرکز تمامی شکل ها نقطه ی (0,0) است و نقطه شروع ربات خود را نیز همان نقطه در نظر بگیرید. برای این بخش تنها کافی است که مسیر طی شده را در داخل شبیه ساز Rviz نمایش دهید. انتظار می‌رود دانشجویان ضرایب کنترلی مناسب را برای برآوردن مسیر خواسته شده توسط ربات در پیاده‌سازی لحاظ کرده باشند.

نکته۱: برای تمامی گام ها زمان اجرا را حداقل به گونه‌ای در نظر بگیرید که در داخل تصویر گرفته شده از مسیر طی شده شکل خواسته شده کاملا قابل تشخیص باشد.

نکته۲: می توانید برای تولید شکل‌های خواسته شده از کد قرار داده شده در اینجا استفاده کنید.

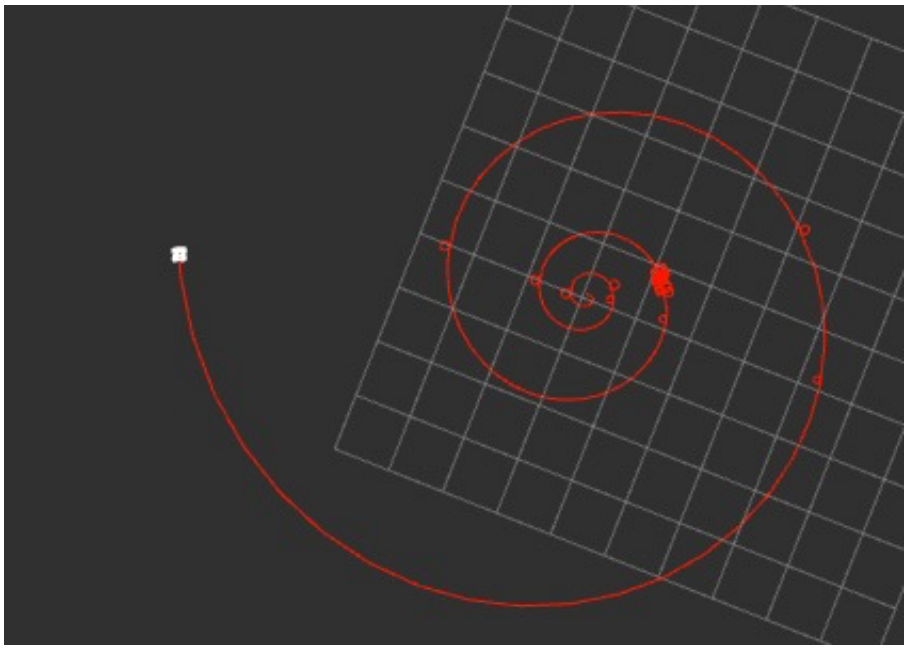
نکته۳: کد های مربوط به هر گام را به هر روشی که راحت‌تر هستید می‌توانید جدا کنید.

نکته۴: برای گام دوم و سوم کنترلری که برای سرعت خطی طراحی می‌کنید باید PID باشد، هرچند برای سرعت زاویه‌ای هر کنترلری را می‌توانید طراحی کنید.

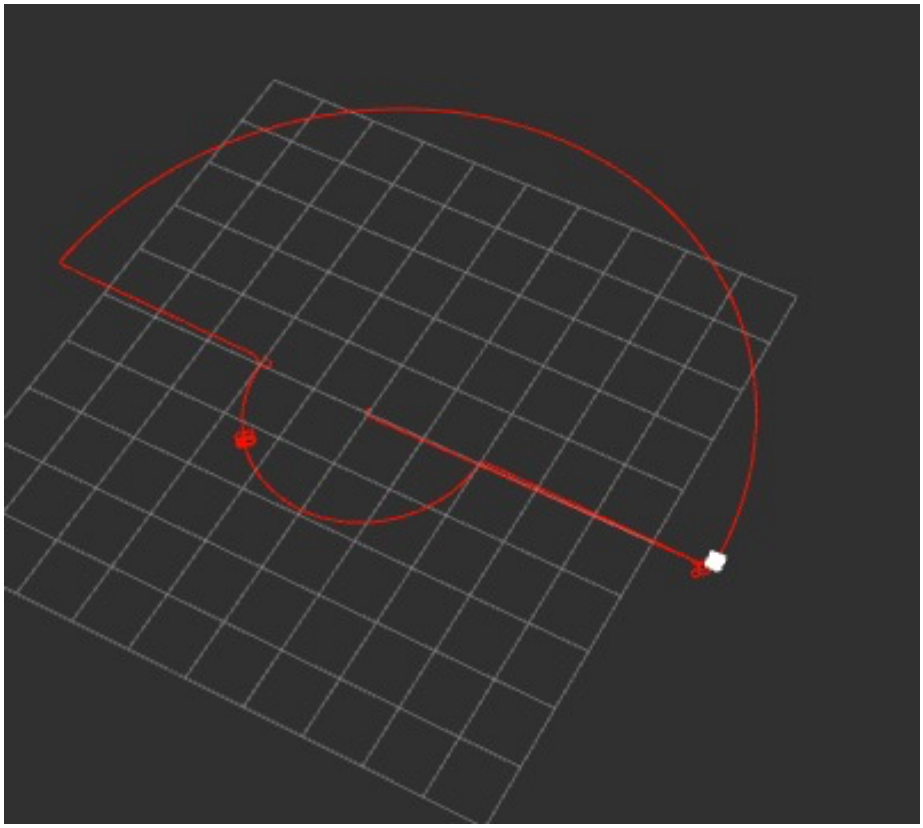
اشکال زیر با مقادیر زیر رسم شده است:

```
self.kp_angle = 2
self.ki_angle = 0.01
self.kd_angle = 0.03
self.kp_distance = 4
self.ki_distance = 0.0
self.kd_distance = 0.3
```

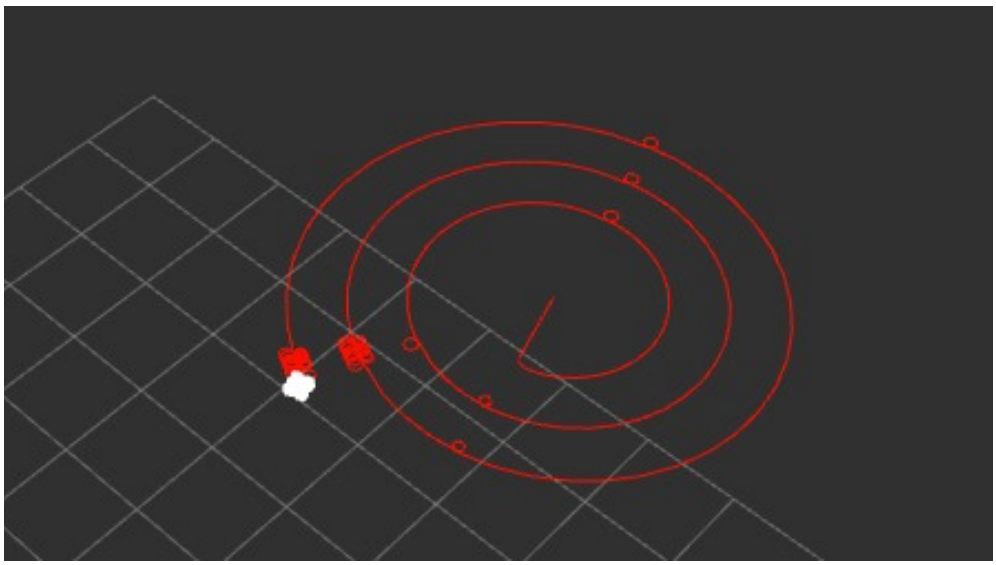
1 - مارپیچ لگاریتمی با $a = 0.17$



2 - ترکیب دو نیم دایره



3 - مارپیچ ارشمیدسی (ArchimedeanSpirals) با growthfactor برابر 0.1



4 - هشت ضلعی منتظم به ضلع دو