## پیاده‌سازی کنترلر PI

در این قسمت همانند تمرین قبل، یک package شامل دو node برای حرکت و مانیتور ربات ساختم و در گره مربوط به حرکت ربات، یک PI Controller را پیاده کردم تا سرعت و زاویه را به روز رسانی کند.

به روزرسانی زاویه:

theta\_star = math.atan2(delta\_y, delta\_x)

    alpha = theta\_star - self.theta

    if alpha > math.pi:

      alpha -= 2 \* math.pi

    if alpha < -math.pi:

      alpha += 2 \* math.pi

    self.vel.angular.z = self.k\_theta \* alpha

به روزرسانی سرعت:

e = math.sqrt(delta\_x \*\* 2 + delta\_y \*\* 2)

    e\_p = e - self.d\_star

    if len(self.E) < self.window:

      self.E.append(e\_p)

    else:

      self.E.pop(0)

      self.E.append(e\_p)

    v = (self.k\_p \* e\_p) + (self.k\_i \* np.array(self.E).sum())

    self.vel.linear.x = v

**پارامترهایی که در پیاده‌سازی استفاده شده‌اند** عبارتند از:

* k\_theta که همان ضریب زاویه است که هم‌واره برابر 0.۴۵ قرار داده شده‌است.
* k\_p که همان ضریب اپسیلون (خطای در لحظه) است که در ابتدای حرکت برابر 0.۱۵ در نظر گرفته شده و با نزدیک شدن ربات به مسیر اصلی‌ای که باید در آن حرکت کند به 0.۴۳ تغییر می‌کند.
* k\_i که همان ضریبی است که کنار انتگرال (همان مجموع خطاهای گذشته) قرار می‌گیرد که در ابتدای حرکت برابر 0.۰۲ در نظر گرفته شده و با نزدیک شدن ربات به مسیر اصلی‌ای که باید در آن حرکت کند به 0.۱۷ تغییر می‌کند.
* window که اندازه‌ی پنجره‌ای است که برای جمع کردن خطاهای گذشته (برای محاسبه‌ی انتگرال) در نظر گرفته می‌شود که مقدار آن ۵ در نظر گرفته شده‌است به این معنی که فقط ۵ خطای آخر را به حساب می‌آورد.
* d\_star که در صورت تمرین به عنوان ds داده شده و به معنای فاصله‌ای است که اگر ربات در آن نسبت به هدف قرار گیرد، هدف بعدی جایگزین هدف کنونی می‌شود.

**مقادیری که برای k\_p (به عنوان ضریب p) و هم‌چنین k\_i (به عنوان ضریب i)** انتخاب شده بر این اساس بوده که اهمیت فاصله‌ی کنونی ربات تا هدف نسبت به خطاهای گذشته اهمیت بیشتری داشته و با نزدیک‌شدن ربات به هدف باید سرعت کمی کاهش می‌یافته یعنی k\_p مقداری بیش‌تر از k\_i دارد.

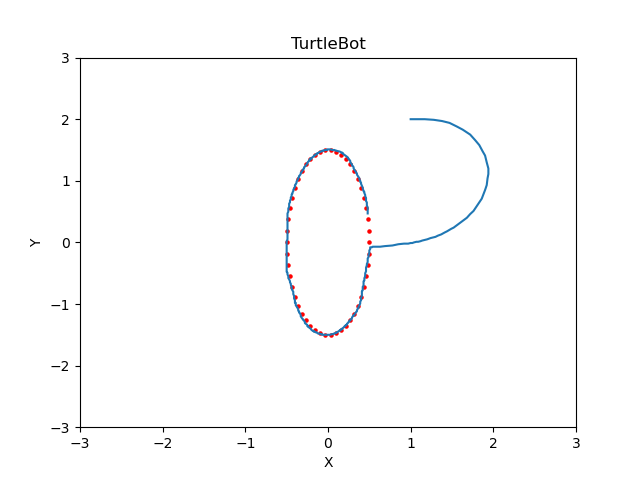
اگر ضریب i خیلی کم شود، باعث می‌شود با نزدیک شدن ربات به هدف دیگر حرکتی نکند و از طرفی اگر این ضریب خیلی زیاد شود هم در زمان عوض شدن هدف و پیچیدن موجب انحراف ربات از مسیر می‌شود.

اگر ضریب p کم شود، باعث می‌شود ربات به صورت کلی آرام‌تر حرکت کند و اگر زیاد شود هم به خاطر عوض شدن خطاهایی که درون پنجره در نظر گرفتیم باعث نوسان سرعت در ربات می‌گردد.

**خطایی** که برای برای حرکت ربات در مسیر در نظر گرفته شده نیز به این صورت است که فاصله‌ی ربات در هر لحظه نسبت به نزدیک‌ترین نقطه‌ی مسیر بررسی می‌شود و میانگین تمامی این خطاها به عنوان خطای کلی گزارش می‌شود.

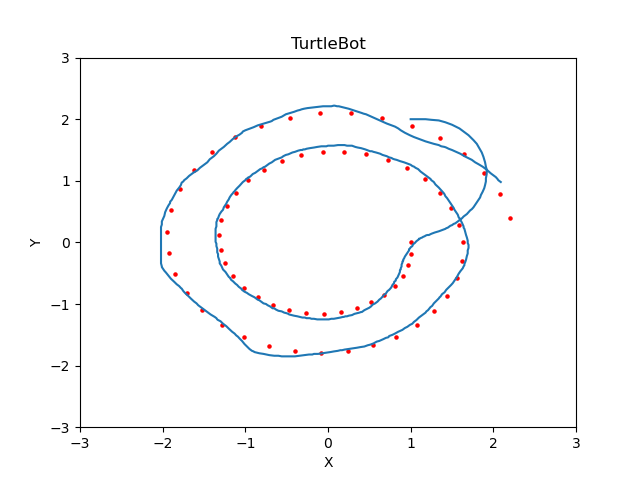
## مسیر حرکت ربات

حال به سراغ مسیری که ربات (با شروع از مکان (1, 2)) برای دو مسیر تعریف شده در تمرین طی کرده است، می‌رویم:

* حرکت روی مسیر بیضی که قطر بزرگ آن ۳ (روی محور y) و قطر کوچک آن ۱ (روی محور x)

خطایی که حرکت ربات در این مسیر داشته برابر ۰.۱۶ اندازه‌گیری شده است.

* حرکت روی مسیر مارپیچی که growth factor آن برابر ۰.۱ در نظر گرفته شده



خطایی که حرکت ربات در این مسیر داشته نیز برابر ۰.۱۰ اندازه‌گیری شده است.