



Department of
Computer Engineering

به نام خدا



Amirkabir University of Technology
(Tehran Polytechnic)

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر
اصول علم ربات

تمرین سری دوم

نام و نام خانوادگی	محسن حسینی
شماره دانشجویی	۹۷۳۱۰۱۶
تاریخ ارسال گزارش	۱۴۰۱/۰۲/۱۹

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

- سوال ۱ - ماتریس تبدیل همگن ۳
- سوال ۲ - نمایش در فریم ۴
- سوال ۳ - Kinematics ۵
- سناریو - گام اول ۶
- سناریو - گام دوم ۸

سوال ۱ – ماتریس تبدیل همگن

باید از فریم ۰ به ۳ برسیم:

$$T_{03} = T_{01}T_{12}T_{23}$$

$$T_{01} = -R_{01}P_1 = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_{12} = -R_{12}P_2 = \begin{pmatrix} \cos(\beta) & -\sin(\beta) & 0.3 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_{23} = -R_{23}P_3 = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0.4 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

سوال ۲ – نمایش در فریم

ابتدا باید معکوس ماتریس R را محاسبه کنیم:

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

حال داریم:

$$\begin{bmatrix} P_b \\ 1 \end{bmatrix} = R_{ba} \times \begin{bmatrix} P_a \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ -2 \\ 1 \end{bmatrix} \rightarrow P_b = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}$$

سوال ۳ – Kinematics

ابتدا عمل Forward Kinematics را انجام می‌دهیم:

$$T_{03} = T_{01}T_{12}T_{23}$$

$$T_{01} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{12} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{23} = \begin{bmatrix} 1 & 0 & 0 & d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{03} = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & -\cos(\theta_1)\sin(\theta_2) & \sin(\theta_1) & d_3\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_1)\cos(\theta_2) & -\sin(\theta_1)\sin(\theta_2) & \cos(\theta_1) & d_3\sin(\theta_1)\cos(\theta_2) \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & a - d_3\sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

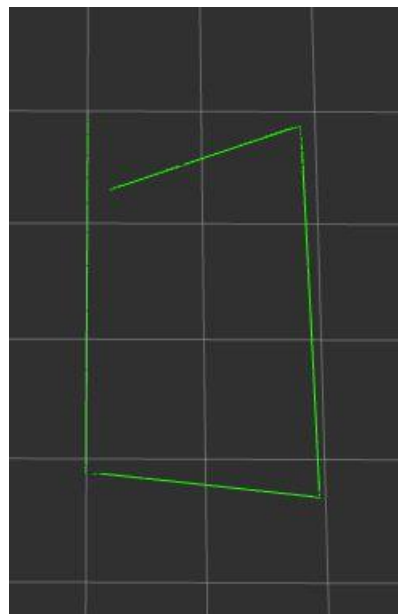
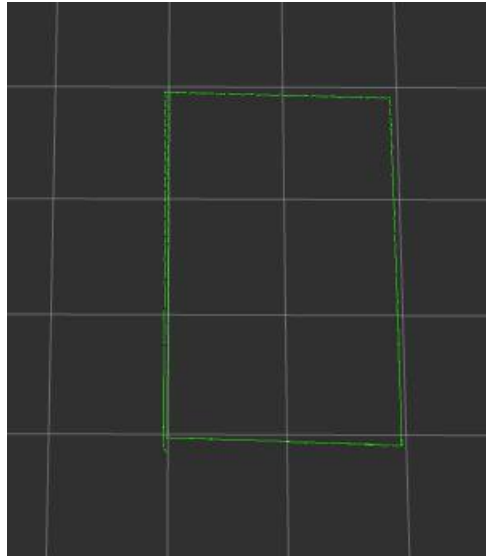
حال با اینورس گرفتن از ماتریس T_{03} به Inverse Kinematics می‌رسیم.

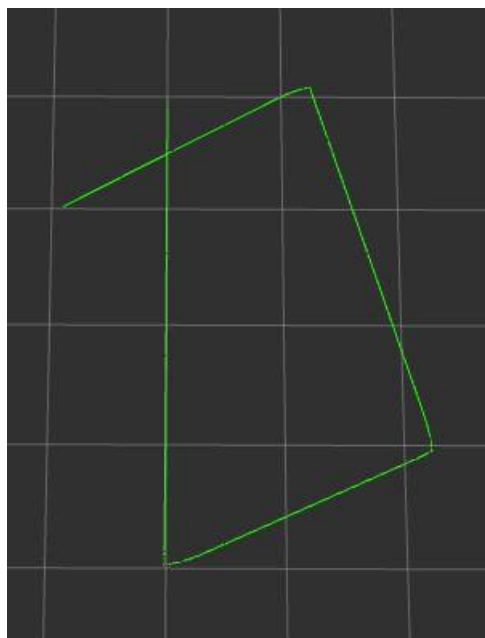
$$T_{30} = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & -\cos(\theta_1)\sin(\theta_2) & -\sin(\theta_2) & a\sin(\theta_2) - d \\ -\cos(\theta_1)\sin(\theta_2) & -\sin(\theta_1)\sin(\theta_2) & -\cos(\theta_2) & a\cos(\theta_2) \\ -\sin(\theta_2) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

سناریو – گام اول

برای بهتر دیده شدن ابعاد مستطیل را به ۲ و ۳ تغییر داده‌ام.

۲- شکل‌های زیر در شبیه‌ساز Rvis به ترتیب برای سرعت‌های ۰.۲ و ۰.۴ و ۰.۶ تهیه شده است.

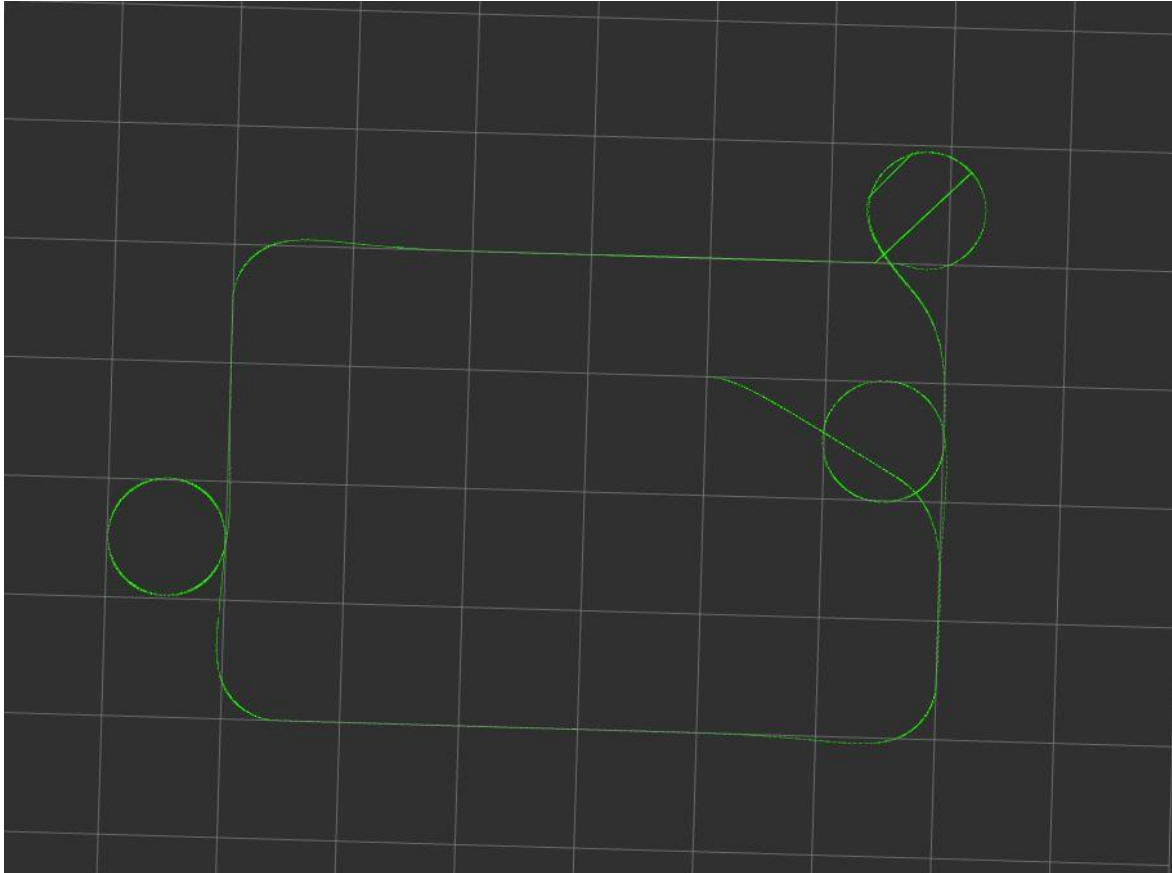




۳- بله همانطور که در شکل‌های بالا مشاهده می‌شود، با افزایش سرعت خطا بسیار افزایش می‌یابد. دلیل این امر هم وقفه‌هایی است که در ROS وجود دارد. بدین معنی که از زمانی که ما دستور اتمام حرکت را می‌دهیم، چند میلی ثانیه زمان می‌برد که این دستور اجرا شود و هرچه سرعت بالاتر باشد خطای بیشتری تولید می‌شود.

سناریو – گام دوم

۱- در شکل زیر خروجی شبیه‌ساز Rviz را مشاهده می‌کنیم. دلیل وجود دایره‌های اضافی را نتوانستم مشخص کنم.



۲- ضرایبی که مورد استفاده قرار گرفتند به شرح زیر هستند:

```
kp_distance = 4
ki_distance = 0.002
kd_distance = 4

kp_angle = 0.5
```

ضریب k_i باید بسیار کوچک باشد تا overshoot کمتری داشته باشیم و ارور ثابت به تدریج برطرف شود.

ضریب k_p و k_d از فرمول $K_d \leq \frac{\sqrt{K_p}}{2}$ بدست می‌آید.

۳- از آنجایی که خطای ما (فاصله) همیشه مثبت است پس با افزایش k_p و k_i ، سرعت افزایش می‌یابد و افزایش k_d باعث کاهش سرعت می‌شود (خطا در حال کم شدن است). به طور کلی افزایش سرعت باعث انحراف بیشتر از مسیر و افزایش خطا می‌شود.