

1) از آنجایی که تبدیل‌ها (دوران و انتقال) به صورت fixed frame بر روی A اعمال میشوند، نحوه صحیح محاسبه مختصات نقطه نهایی از اعمال تبدیل‌ها به طور برعکس (از آخر به اول) بدست می‌آید.

a) From the Homogeneous transform we got $P^O = R_A^O P^A + d_A^O$

That can be rewrite as $p^O = \begin{bmatrix} R_A^O & d_A^O \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^A \\ 1 \end{bmatrix}$

The changes to apply are: $R_{x_0} d R_{z_0}$ reversing $\rightarrow R_{z_0} d R_{x_0}$

$$p^O = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$$p^O = \begin{bmatrix} -2 \\ 3 \\ 7 \\ 1 \end{bmatrix}$$

(b) اگر تبدیل‌ها را به صورت برعکس اعمال کنیم (از آخری به اولی)، مختصات نقطه p به صورت زیر می‌شود.

$$p_{new}^O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$$p_{new}^O = \begin{bmatrix} -3 \\ -5 \\ 6 \\ 1 \end{bmatrix}$$

$$c) p^O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$$p^O = \begin{bmatrix} -4 \\ -5 \\ 3 \\ 1 \end{bmatrix}$$

با جواب به دست آمده در بخش a یکسان نبوده پس میتوان نتیجه گرفت که ترتیب اعمال تبدیل‌های اجرا شده بر روی یک فریم حائز اهمیت است.

(2) برخلاف سوال اول در این سوال تبدیل‌ها بر روی فریم فعلی اعمال شده پس از روش معمولی برای بدست آوردن مختصات نقطه p استفاده کرده و تبدیل‌ها را به ترتیب خودشان اعمال میکنیم.

The changes to apply are: $R_{x_0} d R_{z_0}$

$$a) p^O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$$p^O = \begin{bmatrix} -3 \\ -5 \\ 6 \\ 1 \end{bmatrix}$$

(b) اگر تبدیل‌ها را به صورت برعکس اعمال کنیم (از آخری به اولی)، مختصات نقطه p به صورت زیر می‌شود.

$$p_{new}^O = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

$$p_{new}^O = \begin{bmatrix} -2 \\ 3 \\ 7 \\ 1 \end{bmatrix}$$

(3) بله برعکس همدیگه اعمال میشه، تو سوال اول تبدیل‌ها در هر مرحله بر روی فریمی اعمال میشه که نتیجه تبدیل مرحله قبلی بوده (فریم ثابت A) که با استفاده از رابطه similarity transformation میتوان نشان داد که برای بدست آوردن مختصات نقطه p تبدیل‌ها باید به طور معکوس اعمال شوند، به طور مخالف در سوال دوم تبدیل‌ها نسبت به فریم فعلی که تبدیل‌ها را شروع به اعمال کردیم انجام میشوند به همین دلیل با ترتیب اصلی پیش رفته و به مختصات نقطه نهایی میرسیم.

(4)

- (a) برای مفصل‌های revolute در جهت محور دوران، مفصل‌های prismatic در جهت بازشوندگی مفصل
- (b) $\text{Offset\&angle}(d_i, \theta_i)$ که d_i میزان فاصله بازشوندگی مفصل prismatic بوده که در راستای محور Z ایجاد شده است و θ_i میزان بازشوندگی مفصل revolute نام است.
- (c) بله دو خط متناظر، که بی نهایت خط موازی عمود بر این دو خط از آنها می‌گذرد.
- (d) به صورت عمود بر هر دو محور Z مجاور و در راستای لینک بین آنها
- (e) لینک پارمترها که شامل a_i فاصله بازو بین دو مفصل که در راستای محور X می‌باشد و α_i که میزان پیچش بین دو مفصل که از روی تغییرات محور X بین دو مفصل با توجه به دوران حول محور Z مفصل آفست کمتر.
- (f) روش dh که ربات را بر اساس مفاصل آن به 4 پارامتر برای هر کدام توصیف می‌کند و روش Homogeneous Transformations که با توجه به فریم پایه pose، end-effector، را به صورت ترکیبی از ماتریس های 4×4 توصیف می‌کند.

(5)

پارامتر α میزان twist بین دو مفصل بوده که با توجه به محور X مفصل قبلی میزان چرخندگی محوره‌های Z نسبت به هم در نظر گرفته میشود، پارامتر a فاصله لینک بین مفصل فعلی و قبلی بوده که در راستای محور X ایجاد شده است (اندازه به وجود آمده بین دو محور Z مفصل‌ها)، پارامتر d فاصله به وجود آمد در راستای محور Z مفصل فعلی که میتواند توسط مفصل prismatic هم به وجود آمده باشد، پارامتر θ میزان چرخش مفصل revolute حول محور Z آن مفصل میباشد.

a)

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	90	0	l_2	θ_3

سطر اول: مفصل قبلی وجود نداشته و مفصل بر روی base قرار دارد پس مقدار α, a را 0 قرار میدهم. مفصل revolute است پس به میزان θ_1 دوران میکند و مقدار d ندارد.

سطر دوم: با در نظر گرفتن محور X مفصل 1 به سمت راست، محور Z بین دو مفصل چرخشی نداشته پس α نداریم، اما در راستای X لینکی بین دو مفصل وجود دارد پس a داریم، مفصل revolute است پس به میزان θ_2 دوران میکند و مقدار d ندارد.

سطر سوم: با در نظر گرفتن محور X مفصل 2 به سمت بالا، محور Z از مفصل 2 به 3 به اندازه 90 درجه چرخش داشته پس مقدار α داریم، اما فاصله لینک بین دو مفصل در راستای X مفصل قبلی نبوده پس a نداریم اما در راستای محور Z مفصل فعلی بوده پس مقدار d داریم، در نهایت به طور مشابه مفصل revolute است پس به میزان θ_3 دوران میکند.

b)

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	l_1	θ_1
2	90	l_3	0	θ_2
3	90	$\frac{\sqrt{2}}{2} l_4$	l_5^*	0
4	0	0	0	θ_4

سطر اول: : مفصل قبلی وجود نداشته و مفصل بر روی base قرار دارد پس مقدار α, a را 0 قرار میدهم، با در نظر گرفتن محور Z به سمت پایین، فاصله لینک بین دو مفصل در راستای Z بوده و مقدار d داریم، از طرفی مفصل revolute به اندازه θ_1 دوران میکند.

سطر دوم: با در نظر گرفتن محور X مفصل قبلی به سمت چپ، محور Z به اندازه 90 درجه چرخش داشته است و فاصله به وجود آمده بین دو مفصل (محور های Z آنها) در راستای X بوده و مقدار a داریم، در راستای Z فاصله ای وجود نداشته پس d نداریم و مفصل revolute به اندازه θ_2 دوران میکند.

سطر سوم: با در نظر گرفتن محور X مفصل قبلی به سمت بالا، محور Z به اندازه 90 درجه چرخش داشته است و فاصله به وجود آمده بین دو محور در راستای X به اندازه sin لینک میباشد (مقدار a)، مفصل 3م prismatic بوده و در راستای Z فاصله ایجاد میکند (مقدار d)، مقدار θ نداریم.

سطر چهارم: با در نظر گرفتن Z در راستای محور دوران (به سمت چپ) و محور X به سمت بالا هیچ پیچش یا فاصله چه در راستای X یا Z نداریم، پس مقادیر α, a, d برابر صفر میباشد و مفصل تنها مقدار θ_4 دوران میکند.

(6)

(a) ماتریس بدست آمده از مشتق ضمنی مفاصل ربات که به صورت روشی برای تبدیل، از نحوه جهت گیری مفاصل به مختصات end-effector است، این ماتریس وابسته از نحوه معرفی دستگاه ها بوده و با تبدیل ها متفاوت هر دستگاه Jacobian مختص به آن بدست می آید. این ماتریس نمایان گر سرعت خطی و زاویه مفاصل بوده و حالت ساده آن در دستگاه های کارترین برای سرعت خطی و دستگاه direction-cosine برای سرعت زاویه ای بیان میشود.

(b) در حالت کلی ماتریس Jacobian به صورت 6 سطر و n ستون (به تعداد مفاصل) بوده، که سه سطر بالایی آن مربوط به سرعت خطی که میتواند از باز و بسته شدن مفصل prismatic یا سرعت خطی به وجود آمده حاصل از دوران مفصل revolute باشد، سه سطر پایین این ماتریس مربوط به سرعت زاویه ای هست که مفاصل revolute تولید میکنند.

$$J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix}$$

با در نظر گرفتن $\varepsilon = 0$ برای مفصل های revolute و $\varepsilon = 1$ برای مفصل های prismatic داریم:
 نحوه بدست آوردن سرعت خطی:

$$v = \sum_{i=0}^n \varepsilon_i V_i + \bar{\varepsilon}(\Omega_i \times P_{in})$$

نحوه بدست آوردن سرعت زاویه‌ای:

$$\omega = \sum_{i=0}^n \bar{\varepsilon} \Omega_i$$

(c) میزان جابه‌جایی که یک نقطه در راستای محوری از مختصات در واحد زمان دارد را سرعت خطی و میزان دوران دستگاهی حول یک محورش را در واحد زمان سرعت زاویه‌ای گوئیم.

سوال‌های عملی

1)

a)

```
redha@juanFlatro:~$ cd ros2_ws/
redha@juanFlatro:~/ros2_ws$ . install/setup.bash
redha@juanFlatro:~/ros2_ws$ ros2 launch py_pubsub launch_file.yaml
[INFO] [launch]: All log files can be found below /home/redha/.ros/log/2023-04-1
9-14-13-20-555843-juanFlatro-82999
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [listener-1]: process started with pid [83000]
[listener-1] [INFO] [1681901032.541100345] [listener]: I heard: "S"
[listener-1] [INFO] [1681901033.029568200] [listener]: I heard: "E"
[listener-1] [INFO] [1681901033.529623863] [listener]: I heard: "N"
[listener-1] [INFO] [1681901034.029622207] [listener]: I heard: "D"
[listener-1] [INFO] [1681901034.528926112] [listener]: I heard: "-"
[listener-1] [INFO] [1681901035.029366178] [listener]: I heard: "H"
[listener-1] [INFO] [1681901035.529639713] [listener]: I heard: "E"
[listener-1] [INFO] [1681901036.029625687] [listener]: I heard: "L"
[listener-1] [INFO] [1681901036.529005619] [listener]: I heard: "P"
[listener-1] [INFO] [1681901037.027597038] [listener]: I heard: "I"
[INFO] [listener-1]: process has finished cleanly [pid 83000]
redha@juanFlatro:~/ros2_ws$
```

b)

```
redha@juanFlatro: ~/ros2_ws
[talker-1] [INFO] [1681906125.083410490] [talker]: Publishing: 'h'
[talker-1] [INFO] [1681906125.583412161] [talker]: Publishing: 'h'
[talker-1] [INFO] [1681906126.083424072] [talker]: Publishing: 'w'
[talker-1] [INFO] [1681906126.583406801] [talker]: Publishing: 'i'
[talker-1] [INFO] [1681906127.083407128] [talker]: Publishing: 'i'
[talker-1] [INFO] [1681906127.583414863] [talker]: Publishing: 'i'
[talker-1] [INFO] [1681906128.083339518] [talker]: Publishing: 'done'
[talker-1] [INFO] [1681906128.583398451] [talker]: Publishing: 'j'
[talker-1] [INFO] [1681906129.083419274] [talker]: Publishing: 'j'
[talker-1] [INFO] [1681906129.583458959] [talker]: Publishing: 'v'
[talker-1] [INFO] [1681906130.083348026] [talker]: Publishing: 'v'
[talker-1] [INFO] [1681906130.583413518] [talker]: Publishing: 'h'
[talker-1] [INFO] [1681906131.083419541] [talker]: Publishing: 'h'
[talker-1] [INFO] [1681906131.583452273] [talker]: Publishing: 'w'
[talker-1] [INFO] [1681906132.083446540] [talker]: Publishing: 'i'
[talker-1] [INFO] [1681906132.583352329] [talker]: Publishing: 'i'
[talker-1] [INFO] [1681906133.083405422] [talker]: Publishing: 'i'
[talker-1] [INFO] [1681906133.583417682] [talker]: Publishing: 'done'
[talker-1] [INFO] [1681906134.083410427] [talker]: Publishing: 'j'
[talker-1] [INFO] [1681906134.583437834] [talker]: Publishing: 'j'
[talker-1] [INFO] [1681906135.083411766] [talker]: Publishing: 'v'
[talker-1] [INFO] [1681906135.583434562] [talker]: Publishing: 'v'
[talker-1] [INFO] [1681906136.083472069] [talker]: Publishing: 'h'

redha@juanFlatro: ~/ros2_ws
redha@juanFlatro:~/ros2_ws$ ros2 param get /minimal_publisher test
Integer value is: 1
redha@juanFlatro:~/ros2_ws$ ros2 param get /talker test
Integer value is: 2
redha@juanFlatro:~/ros2_ws$
```

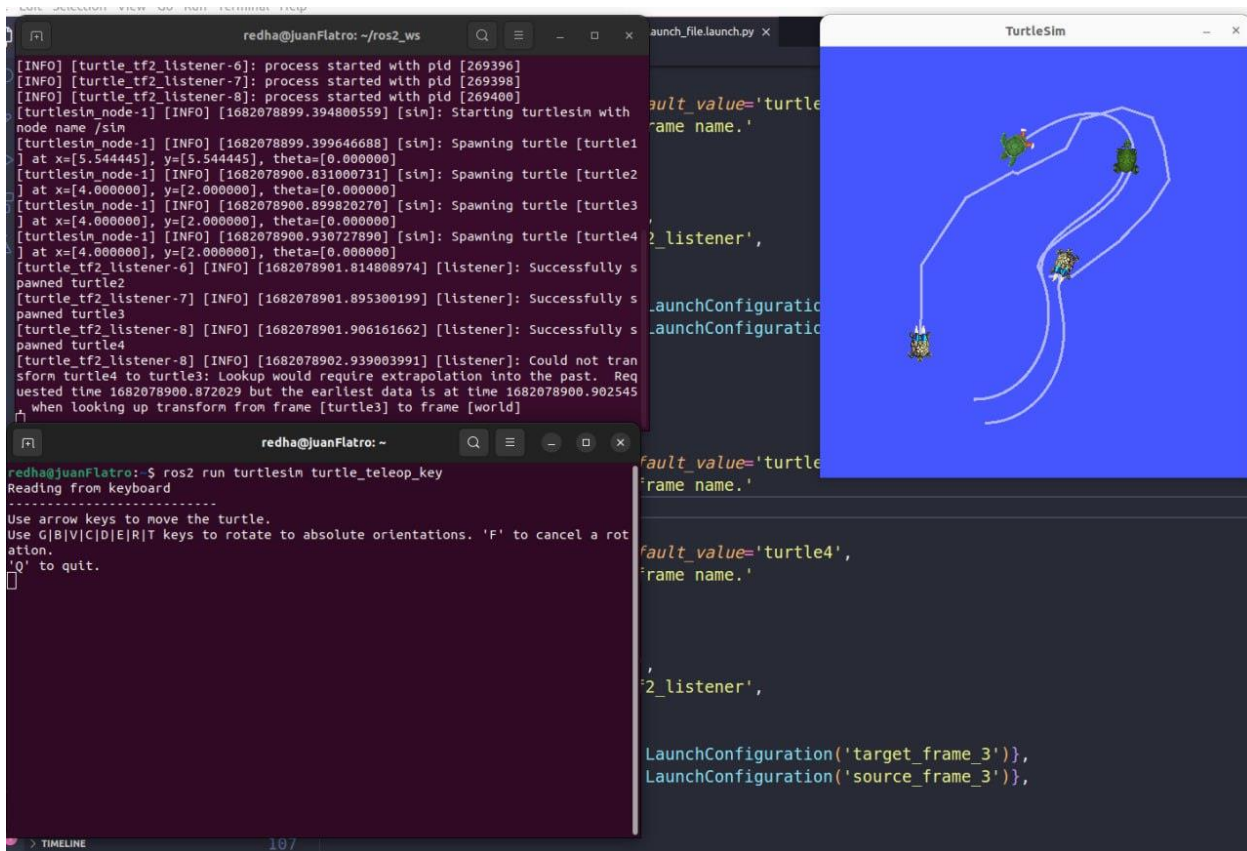
(c) کتابخانه tf2 با ایجاد درختی که هر node آنها نمایانگر فریمی از ربات است به طور بصری از ایجاد حلقه‌های بین فریم جلوگیری میکند به علاوه با استفاده از time-stamp ها اطمینانی حاصل میکند که تبدیل های محاسبه شده بین فریم ها دارای صحت است.

(d) روشی است که برای تعیین ارتباط static بین فریم های ربات که معمولا برای بدست آوردن اطلاعات از نقطه ثابتی از ربات نسبت به یک فریم مرسوم (مانند پایه) مورد استفاده قرار میگیرد، که با اعمال محدودیت هایی به transform_publisher میتوان شرایطی ایجاد کرد که دو فریم مورد بررسی نسبت به هم static شوند تا در صورت نبود static_transform_publisher بتوان از آن استفاده کرد.

(e) تبدیلات ایستا به طور کلی نسبت به مرجعی ثابت صورت میگیرند و طی زمان دچار تغییر نمیشوند از طرفی تبدیلات پویا همواره دستخوش تغییر بوده و وابسته به وضعیت فعلی بخش مورد بررسی است، با توجه به شی مورد بررسی و هدف کلی میتواند تبدیلات به صورت پویا یا ایستا انجام شود که آیا در طی جابه‌جایی و دوران‌ها شی ثابت میماند یا دچار تغییر میشود.

(f) کتابخانه tf2 میتواند با استفاده از اطلاعات موجود در درخت URDF یک درخت tf2 از رابطه بین مختصات فریم ها تشکیل دهد، به طور مخصوص از اطلاعات مربوط به پدر-فرزند بین لینک‌ها و مفصل‌ها که در درخت URDF به شکل سلسله مراتبی تعریف شده‌اند.

2)



3)

