



دانشگاه خواجه نصیرالدین طوسی
دانشکده برق - گروه مخابرات

تمرین درس رباتیک دوره کارشناسی ارشد

رشته مهندسی مخابرات

عنوان

تمرین رباتیک

نگارش

علیرضا امیری

مهر ۱۴۰۳

فصل ۱

پاسخ سوالات سری دوم

۱.۱ پاسخ سوال ۱ - الف

۱.۱.۱ پاسخ بخش اول

در این بخش، از آنجا که تنها موقعیت نهایی ابزار مورد نیاز است و جهت یابی آن خواسته نشده است، می توانیم با استفاده از معادله ی (۲.۶۱) موقعیت نهایی را به دست آوریم

$$\mathbf{A_P} = \mathbf{A_{P_{O_B}}} + \mathbf{A_{R_B}} \cdot \mathbf{B_P}$$

برای این منظور، با تقسیم معادلات به دو بخش نوک ابزار و نقطه ی A و ترکیب این دو، می توانیم در نهایت موقعیت نوک ابزار را نسبت به مبدا مختصات به دست آوریم. در بخش اول که مربوط به پیدا کردن مختصات نقطه ی A در مختصات مبدا است، ابتدا ماتریس دوران حول محور ثابت Z به اندازه ی θ و ماتریس دوران حول محور متغیر y به اندازه ی ϕ را تعریف می کنیم.

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_z(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

با در اختیار داشتن ماتریس های دوران، می توانیم با ضرب آن در موقعیت ابتدایی نقطه ی A، موقعیت جدید نقطه ی A را به دست بیاوریم.

$$A_0 = \begin{pmatrix} 0 \\ 0 \\ L_{1234} \end{pmatrix}$$

$$P_A = \begin{pmatrix} L_{1234} \cos(\phi) \sin(\theta) \\ L_{1234} \sin(\phi) \sin(\theta) \\ L_{1234} \cos(\theta) \end{pmatrix}$$

در اینجا باید توجه شود که چون دوران حول محور Z، دوران حول محور متغیر است، بنابراین این ماتریس از سمت چپ ضرب شده است

۲.۱.۱ پاسخ بخش دوم

در بخش بعد، مانند آنچه که در این بخش انجام شد تلاش می شود موقعیت نوک ابزار ربات جراح نسبت به موقعیت جدید نقطه ی A به دست می آید.

برای تعریف ماتریس های دوران مربوط به این بخش خواهیم داشت:

$$R_y(\theta_d) = \begin{bmatrix} \cos(\theta_d) & 0 & \sin(\theta_d) \\ 0 & 1 & 0 \\ -\sin(\theta_d) & 0 & \cos(\theta_d) \end{bmatrix}$$

$$R_z(\phi_d) = \begin{bmatrix} \cos(\phi_d) & -\sin(\phi_d) & 0 \\ \sin(\phi_d) & \cos(\phi_d) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\alpha + \psi) = \begin{bmatrix} \cos(\alpha + \psi) & 0 & \sin(\alpha + \psi) \\ 0 & 1 & 0 \\ -\sin(\alpha + \psi) & 0 & \cos(\alpha + \psi) \end{bmatrix}$$

از ضرب این ماتریس ها با رعایت قاعده ی ضرب از سمت چپ برای دوران های حول محور متحرک در موقعیت اولیه ی نقطه ی ابزار نسبت به نقطه ی A خواهیم داشت:

$$R_{\gamma} = R_y(\alpha + \psi) \cdot R_z(\phi_d) \cdot R_y(\theta_d)$$

$$R_{\gamma} = \begin{pmatrix} \cos(\alpha + \psi) \cos(\phi_d) \cos(\theta_d) - \sin(\alpha + \psi) \sin(\theta_d) & -\cos(\alpha + \psi) \sin(\phi_d) & \sin(\alpha + \psi) \cos(\theta_d) + \cos(\alpha + \psi) \cos(\phi_d) \sin(\theta_d) \\ \cos(\theta_d) \sin(\phi_d) & \cos(\phi_d) & \sin(\phi_d) \sin(\theta_d) \\ -\cos(\alpha + \psi) \sin(\theta_d) - \sin(\alpha + \psi) \cos(\phi_d) \cos(\theta_d) & \sin(\alpha + \psi) \sin(\phi_d) & \cos(\alpha + \psi) \cos(\theta_d) - \sin(\alpha + \psi) \cos(\phi_d) \sin(\theta_d) \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix}$$

$$A_{P_D} = R_{\gamma} \cdot \mathbf{d} = \begin{pmatrix} d (\sin(\alpha + \psi) \cos(\theta_d) + \cos(\alpha + \psi) \cos(\phi_d) \sin(\theta_d)) \\ d \sin(\phi_d) \sin(\theta_d) \\ d (\cos(\alpha + \psi) \cos(\theta_d) - \sin(\alpha + \psi) \cos(\phi_d) \sin(\theta_d)) \end{pmatrix}$$

در نهایت پس از به دست آوردن مختصات نقطه ی A نسبت به مرجع و مختصات نقطه ی ابزار نسبت به A، می توانیم مختصات خالص ابزار به مرجع را با استفاده از جمع این دو بردار به دست بیاوریم:

$$O_P = A_{PD} + O_{PA} = \begin{pmatrix} d(\sin(\alpha + \psi) \cos(\theta_d) + \cos(\alpha + \psi) \cos(\phi_d) \sin(\theta_d)) + L_{1234} \cos(\phi) \sin(\theta) \\ L_{1234} \sin(\phi) \sin(\theta) + d \sin(\phi_d) \sin(\theta_d) \\ L_{1234} \cos(\theta) + d(\cos(\alpha + \psi) \cos(\theta_d) - \sin(\alpha + \psi) \cos(\phi_d) \sin(\theta_d)) \end{pmatrix}$$

برای محاسبات فوق که به صورت نمادین در متلب انجام شده اند، قابلیت جایگذاری عددی برای پارامترهای مورد استفاده نیز طراحی شده است تا بتوان با استفاده از داده های عددی، موقعیت دقیق ابزار را نیز محاسبه کرد. همچنین، با مراجعه به مقاله ی مربوط به ربات جراح چشم آرش در [۱] مشاهده می شود که روابط مورد استفاده برای تعیین موقعیت ابزار این ربات به صورت زیر تعریف شده است که تفاوت هایی با مقادیر به دست آمده به روش بالا دارد:

$$O_P = \begin{pmatrix} d(\sin(\alpha + \psi) \cos(\theta) \cos(\phi) + \cos(\alpha + \psi) \sin(\theta) \cos(\psi)) + L_{1234} \sin(\theta) \\ d \sin(\phi) \sin(\psi + \alpha) \\ d(-\sin(\theta) \cos(\phi) \sin(\psi + \alpha) + \cos(\theta) \cos(\psi + \alpha)) + L_{1234} \cos(\theta) \end{pmatrix}$$

۳.۱.۱ پاسخ سوال ۲

۱.۳.۱.۱ حل به روش بستر حلقه

در این روش حل، با در نظر گرفتن مختصات ابزار ربات نسبت به مبدا مختصات، O یک بار به صورت مستقیم و با نام گذاری های

$$[x_e, y_e, \theta_e]$$

و یک بار نیز با در نظر گرفتن هر یک از مفصل ها و در نهایت با برابر قرار دادن مقادیر به دست آمده از این دو روش، مختصات نهایی را پیدا می کنیم. برای این منظور خواهیم داشت:

$$x_e = d_1 + a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2)$$

$$y_e = a_3 \sin(\theta_2 + \theta_3) + a_2 \sin(\theta_2)$$

$$\theta_e = \theta_2 + \theta_3$$

۲.۳.۱.۱ حل به روش دناویت هارتنبرگ

برای حل این مسئله به روش دناویت هارتنبرگ، لازم است ابتدا پارامترهای دناویت هارتنبرگ برای هر یک از مفصل‌ها مشخص شده و در ادامه، با پیاده سازی برنامه ی نرم افزاری و اجرای آن، مختصات نقطه حاصل از تبدیل به دست آمده محاسبه شود. برای اجرای نرم افزاری، از تابع موجود در پایگاه گیت هاب این درس استفاده شده است.

از آنجا که مفصل اول از نوع کشویی است، بنابراین تنها یک پارامتر d_1 دارد. دو مفصل چرخشی بعدی، هر یک دارای یک مقدار a و یک زاویه چرخش θ می باشند. بنابراین، پارامترهای دناویت هارتنبرگ به صورت زیر محاسبه می شود:

$$T = \begin{pmatrix} 1^\circ \cos(\theta_2 + \theta_3) & -1^\circ \sin(\theta_2 + \theta_3) & \circ & 1^\circ a_3 \cos(\theta_2 + \theta_3) + 1^\circ a_2 \cos(\theta_2) \\ 1^\circ \sin(\theta_2 + \theta_3) & 1^\circ \cos(\theta_2 + \theta_3) & \circ & 1^\circ a_3 \sin(\theta_2 + \theta_3) + 1^\circ a_2 \sin(\theta_2) \\ \circ & \circ & 1^\circ & 1^\circ d_1 \\ \circ & \circ & \circ & 1^\circ \end{pmatrix}$$

مشاهده می شود که مقادیر به دست آمده برای x_e و y_e به طور مشخص با نتایج به دست آمده از روش بسترار حلقه مطابقت دارند. همچنین برای محاسبه ی زاویه ی θ مطابق رابطه ی زیر خواهیم داشت:

$$\theta = \arccos\left(\frac{\text{trace}_A - 1}{2}\right)$$

$$\theta = \frac{18^\circ \arccos(1^\circ \cos(\theta_2 + \theta_3))}{\pi}$$

مشخص است که با ساده سازی این عبارت، به جواب به دست آمده در قسمت پیشین خواهیم رسید.

۴.۱.۱ پاسخ سوال ۳

۵.۱.۱ پاسخ سوال ۴

۱. در این بخش ابتدا یکه بودن هر یک از بردارهای u و v و w را بررسی کرده و مقدار مجهول را با استفاده از این رابطه مشخص می کنیم.

$$1 = \sqrt{x^2 + 0.6046^2 + 0.0977^2} \Rightarrow x = \pm 0.7905$$

$$1 = \sqrt{0.3864^2 + 0.3686^2 + z^2} \Rightarrow z = \pm 0.8454$$

$$1 = \sqrt{0.4752^2 + y^2 + 0.5250^2} \Rightarrow y = \pm 0.7060$$

با به دست آوردن مقادیر ممکن برای درایه های مجهول، لازم است علامت آنها نیز به درستی مشخص شود. برای این کار، با محاسبه ی دترمینان ماتریس دوران حاصل از هر یک از این پارامترها، می توانیم علامت صحیح پارامترها را در حالتی که دترمینان برابر ۱ باشد به دست آوریم.

```
1.3.1: Defining unknown elements in 1
rotation matrix R
```

2

```
A = [0.7905, -0.3864, 0.4752; 3
```

```
0.6046, 0.3686, -0.7060; 4
```

```
0.0977, 0.8454, 0.5250] 5
```

```
det_A = det(A); 6
```

7

```
disp('Determinant of matrix A:'); disp(8
```

```
det_A)
```

در نتیجه:

$$x = +0.7905, \quad y = -0.7060, \quad z = +0.8454$$

• محاسبه زوایای اوایلر حول محور ثابت:

1.3.2: Inverse Rpry

```
% Define the rotation matrix R with numerical
    values
```

```
disp(R)
```

```
% Calculate beta
```

```
beta = atan2(-R(3,1), sqrt(R(1,1)^2 + R(2,1)^2));
```

```
% Calculate gamma
```

```
gamma = atan2(R(2,1) / cos(beta), R(1,1) / cos(beta));
```

```
% Calculate alpha
```

```
alpha = atan2(R(3,2) / cos(beta), R(3,3) / cos(beta));
```

```
% Display the calculated angles in radians
```

```
fprintf('Calculated angles: alpha = %.2f rad, beta
```



```

= %.2f rad, gamma = %.2f rad\n', alpha, beta,
gamma);
18
% Optionally, convert radians to degrees
19
alpha_deg = rad2deg(alpha);
20
beta_deg = rad2deg(beta);
21
gamma_deg = rad2deg(gamma);
22
23
% Display the angles in degrees
24
fprintf('Calculated angles: alpha = %.2f deg, beta
25
= %.2f deg, gamma = %.2f deg\n', alpha_deg,
beta_deg, gamma_deg);

```

$$\alpha = 58.16^\circ, \quad \beta = -5.61^\circ, \quad \gamma = +37.41^\circ$$

• محاسبه زوایای اویلر حول محور متحرک: Ruvw

```

1
1.3.3: Inversw Ruvw
2
We calculate Euler angles (alpha, beta,
3
gamma) from a given rotation matrix R.
4
% Define the rotation matrix R with
5
numerical values
6
disp(R)
7
% Calculate beta

```

```

beta = atan2(R(1,3), sqrt(R(1,1)^2 + R(1,2)
    ^2));
9

% Calculate alpha
10
alpha = atan2(-R(2,3) / cos(beta), R(3,3) /
    cos(beta));
11
12
% Calculate gamma
13
gamma = atan2(-R(1,2) / cos(beta), R(1,1) /
    cos(beta));
14
15
% Display the calculated angles in radians
16
fprintf('Calculated angles: alpha = %.2f
    rad, beta = %.2f rad, gamma = %.2f rad\n
    ', alpha, beta, gamma);
17
18
% Optionally, convert radians to degrees
19
alpha_deg = rad2deg(alpha);
20
beta_deg = rad2deg(beta);
21
gamma_deg = rad2deg(gamma);
22
23
% Display the angles in degrees
24
fprintf('Calculated angles: alpha = %.2f
    deg, beta = %.2f deg, gamma = %.2f deg\n
    ', alpha_deg, beta_deg, gamma_deg);
25

```

$$\alpha = 53.36^\circ, \quad \beta = 28.37^\circ, \quad \gamma = 26.05^\circ$$

• محاسبه زوایای اویلر حول محور متحرک Rvwv:

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
1.3.4: Inverse Rvwv
In this section we calculate the Euler
    angels given a Rotation Matrix in the
    system of vvw.
% Define the rotation matrix R with
    numerical values
disp(R)

% Calculate beta
beta = atan2(sqrt(R(3,1)^2 + R(3,2)^2),R
    (3,3));

% Calculate alpha
alpha = atan2(R(2,3) / sin(beta), R(1,3) /
    sin(beta));

% Calculate gamma
gamma = atan2(R(3,2) / sin(beta), -R(3,1) /
    sin(beta));

% Display the calculated angles in radians
```

```

fprintf('Calculated angles: alpha = %.2f 17
      rad, beta = %.2f rad, gamma = %.2f rad\n
      ', alpha, beta, gamma);
18
% Optionally, convert radians to degrees 19
alpha_deg = rad2deg(alpha); 20
beta_deg = rad2deg(beta); 21
gamma_deg = rad2deg(gamma); 22
23
% Display the angles in degrees 24
fprintf('Calculated angles: alpha = %.2f 25
      deg, beta = %.2f deg, gamma = %.2f deg\n
      ', alpha_deg, beta_deg, gamma_deg);

```

$$\alpha = -56.06^\circ, \quad \beta = 58.33^\circ, \quad \gamma = 96.59^\circ$$

• محاسبه زوایای اویلر حول محور متحرک: R_{wuw}

```

1
1.3.5: Inverse Rwuw 2
In this section we calculate the Euler angels given
a Rotation Matrix in the system of wuw.
% Define the rotation matrix R with numerical 4
values
disp(R) 5
% Calculate beta 6

```

```
beta = atan2(sqrt(R(3,1)^2 + R(3,2)^2),R(3,3));      7
                                                    8
% Calculate alpha                                     9
alpha = atan2(R(1,3) / sin(beta), -R(2,3) / sin(    10
    beta));
                                                    11
% Calculate gamma                                     12
gamma = atan2(R(3,1) / sin(beta), R(3,2) / sin(beta) 13
    ));
                                                    14
% Display the calculated angles in radians            15
fprintf('Calculated angles: alpha = %.2f rad, beta    16
    = %.2f rad, gamma = %.2f rad\n', alpha, beta,
    gamma);
                                                    17
% Optionally, convert radians to degrees             18
alpha_deg = rad2deg(alpha);                          19
beta_deg = rad2deg(beta);                             20
gamma_deg = rad2deg(gamma);                           21
                                                    22
% Display the angles in degrees                      23
fprintf('Calculated angles: alpha = %.2f deg, beta    24
    = %.2f deg, gamma = %.2f deg\n', alpha_deg,
    beta_deg, gamma_deg);
```

$$\alpha = 33.94^\circ, \quad \beta = 58.33^\circ, \quad \gamma = 6.59^\circ$$

۲. محاسبه ی زاویه و محور دوران معادل:

1.3.6: Inverse Screw

% Step 1: Define the rotation matrix R (example)

disp(R)

*% Step 2: Calculate the rotation angle theta using the
trace of R*

theta = acos((trace(R) - 1) / 2);

*% Step 3: Check if sin(theta) is zero (to avoid
division by zero)*

if sin(theta) == 0

error('Singularity encountered: sin(theta) is zero.
Unable to compute the axis.');

end

*% Step 4: Calculate the components of the screw axis
= [sx, sy, sz]*

sx = (R(3,2) - R(2,3)) / (2 * sin(theta));

sy = (R(1,3) - R(3,1)) / (2 * sin(theta));

sz = (R(2,1) - R(1,2)) / (2 * sin(theta));

```

% Step 5: Display the calculated screw axis and rotation angle
fprintf('Calculated rotation angle (theta) in radians:
%.4f\n', theta);
fprintf('Calculated rotation angle (theta) in degrees:
%.4f\n', rad2deg(theta));
fprintf('Screw axis vector s = [sx, sy, sz]: [%.4f, %.4f, %.4f]\n', sx, sy, sz);

% Optional: Normalize the screw axis (if you want the unit vector)
s_magnitude = sqrt(sx^2 + sy^2 + sz^2);
s_normalized = [sx, sy, sz] / s_magnitude;
fprintf('Normalized screw axis vector s: [%.4f, %.4f, %.4f]\n', s_normalized(1), s_normalized(2), s_normalized(3));

%% Step 6: Calculate the theta vector (theta_x, theta_y, theta_z)
theta_x = theta * sx;
theta_y = theta * sy;
theta_z = theta * sz;

% Step 7: Display the theta vector
fprintf('Theta vector = [theta_x, theta_y, theta_z]:

```

```
[% .4f, % .4f, % .4f] \n', theta_x, theta_y, theta_z);
```

$$\theta = ۳۰/۰۰^{\circ},$$

$$\mathbf{S} = [s_x, s_y, s_z] : [۰/۸۲۵۵, ۰/۲۰۰۹, ۰/۵۲۷۳]$$

$$\vec{\theta} = [\theta_x, \theta_y, \theta_z] : [۱/۰۰۸۵, ۰/۲۴۵۴, ۰/۶۴۴۲]$$

۳. محاسبه نمایش چهارگان ماتریس دوران R:

```

1
2 % Step 1: Define the rotation matrix R (example)
3 disp(R)
4
5 % Step 2: Calculate the scalar part of the quaternion (e4)
6 e4 = 0.5 * sqrt(1 + R(1,1) + R(2,2) + R(3,3));
7
8 % Step 3: Calculate the vector components of the
9 quaternion
10 e1 = (R(3,2) - R(2,3)) / (4 * e4); % e1
11 e2 = (R(1,3) - R(3,1)) / (4 * e4); % e2
12 e3 = (R(2,1) - R(1,2)) / (4 * e4); % e3
13
14 % Step 4: Display the quaternion as a vector [e1, e2, e3, e4]
15 quaternion_from_R = [e1, e2, e3, e4];
```


disp('Quaternion vector [e1, e2, e3, e4]:');	15
disp(quaternion_from_R);	16

$$\vec{e} = [\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4] = [0.4735, 0.1152, 0.3024, 0.8192]$$

۶.۱.۱ پاسخ سوال ۵

با در اختیار داشتن ماتریس دوران، میتوان با استفاده از ویژگی های مختلف این ماتریس اقدام به پیدا کردن المان های مجهول کرد. یکی از راه ها، محاسبه ی ترانهاده و وارون ماتریس و برابر قرار دادن المان های متناظر است که به دلیل پیچیدگی زیاد این روش، از ویژگی دیگری استفاده می شود. به عنوان جایگزین، با استفاده از روابط ضرب خارجی ستون های ماتریس دوران چنان که در بخش پایین قرار داده شده است استفاده می کنیم.

$$\mathbf{u} \times \mathbf{v} = \mathbf{w},$$

$$\mathbf{v} \times \mathbf{w} = \mathbf{u}, \quad (1.1)$$

$$\mathbf{w} \times \mathbf{u} = \mathbf{v}$$

ماتریس R مطابق صورت سوال به شکل زیر تعریف می شود:

$$R = \begin{pmatrix} \cos(t) & a & b \\ \sin(t) & \frac{\sqrt{2}k \cos(t)}{2} & c \\ 0 & -\frac{\sqrt{2}k \sin(t)}{2} & d \end{pmatrix}$$

با قرار دادن ستون های این ماتریس به عنوان $\mathbf{u}, \mathbf{v}, \mathbf{w}$ و محاسبه ی ضرب خارجی آنها خواهیم داشت:

$$\mathbf{u} \times \mathbf{v} = \begin{pmatrix} -\frac{\sqrt{\gamma} k \sin(t)^{\gamma}}{\gamma} \\ \frac{\sqrt{\gamma} k \sin(\gamma t)}{\gamma} \\ -\frac{\sqrt{\gamma} k \sin(t)^{\gamma}}{\gamma} - a \sin(t) + \frac{\sqrt{\gamma} k}{\gamma} \end{pmatrix}$$

$$\mathbf{v} \times \mathbf{w} = \begin{pmatrix} \frac{\sqrt{\gamma} k (d \cos(t) + c \sin(t))}{\gamma} \\ -a d - \frac{\sqrt{\gamma} b k \sin(t)}{\gamma} \\ a c - \frac{\sqrt{\gamma} b k \cos(t)}{\gamma} \end{pmatrix}$$

$$\mathbf{w} \times \mathbf{u} = \begin{pmatrix} -d \sin(t) \\ d \cos(t) \\ b \sin(t) - c \cos(t) \end{pmatrix}$$

با برابر قرار دادن این ماتریس ها با ستون های ماتریس دوران و حل معادلات به دست آمده به ازای مقادیر

a,b,c,d,k به جواب های زیر دست می یابیم.

$$\mathbf{a} = \begin{pmatrix} -\frac{\sin(t)}{\sqrt{\sin(t)^{\gamma}+1}} \\ \frac{\sin(t)}{\sqrt{\sin(t)^{\gamma}+1}} \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} -\frac{\sin(t)^{\gamma}}{\sqrt{\sin(t)^{\gamma}+1}} \\ \frac{\sin(t)^{\gamma}}{\sqrt{\sin(t)^{\gamma}+1}} \end{pmatrix}$$

$$\mathbf{c} = \begin{pmatrix} \frac{\sin(\gamma t)}{\gamma \sqrt{\sin(t)^{\gamma}+1}} \\ -\frac{\sin(\gamma t)}{\gamma \sqrt{\sin(t)^{\gamma}+1}} \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} \frac{1}{\sqrt{\sin(t)^{\gamma}+1}} \\ -\frac{1}{\sqrt{\sin(t)^{\gamma}+1}} \end{pmatrix}$$

$$\mathbf{k} = \begin{pmatrix} \frac{\sqrt{2}}{\sqrt{\sin(t)^4 + 1}} \\ -\frac{\sqrt{2}}{\sqrt{\sin(t)^4 + 1}} \end{pmatrix}$$

۷.۱.۱ پاسخ سوال ۶

با در اختیار داشتن مقادیر زوایای اویلر، می توان ماتریس دوران و مشتق آن را به دست آورده و با استفاده از المان های این ماتریس ها، سرعت زاویه ای و ماتریس تبدیل آن را به دست آورد. برای این منظور، مطابق فرایند زیر عمل می کنیم.

```

%% Section 1: Define the Euler Angles and Rotation Matrices
for UVW
% Define symbolic angles alpha, beta, gamma for the Euler
angles
syms alpha beta gamma real
syms alpha_dot beta_dot gamma_dot real % Time derivatives
of the angles

% Step 1: Define symbolic rotation matrices for each axis
% Rotation about the x-axis (Ru(alpha))
Ru_alpha = [1, 0, 0;
0, cos(alpha), -sin(alpha);
0, sin(alpha), cos(alpha)];

% Rotation about the y-axis (Rv(beta))
Rv_beta = [cos(beta), 0, sin(beta);
0, 1, 0;

```

```

-sin(beta), 0, cos(beta)]; 15
16
% Rotation about the z-axis (Rw(gamma)) 17
Rw_gamma = [cos(gamma), -sin(gamma), 0; 18
sin(gamma), cos(gamma), 0; 19
0, 0, 1]; 20

```

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

$$R_{v_\beta} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

$$R_{w_\gamma} = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

آنگاه ماتریس دوران حاصل از سه دوران حول محورهای اوایلر از ضرب سه ماتریس فوق به دست می آید:

$$R_{uvw} = \begin{pmatrix} \cos(\beta) \cos(\gamma) & -\cos(\beta) \sin(\gamma) & \sin(\beta) \\ \cos(\alpha) \sin(\gamma) + \cos(\gamma) \sin(\alpha) \sin(\beta) & \cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma) & -\cos(\beta) \sin(\alpha) \\ \sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta) & \cos(\gamma) \sin(\alpha) + \cos(\alpha) \sin(\beta) \sin(\gamma) & \cos(\alpha) \cos(\beta) \end{pmatrix}$$

در ادامه و با محاسبه ی مشتق این ماتریس، سرعت زاویه ای ω را محاسبه می کنیم.

```

%% Section 1: Define the Euler Angles and 1
Rotation Matrices for UVW

```

```

% Define symbolic angles alpha, beta, gamma
% for the Euler angles

syms alpha beta gamma real          3
syms alpha_dot beta_dot gamma_dot real % 4
% Time derivatives of the angles

% Step 1: Define symbolic rotation matrices
% for each axis

% Rotation about the x-axis (Ru(alpha))  7
Ru_alpha = [1,          0,          0;  8
0, cos(alpha), -sin(alpha);  9
0, sin(alpha),  cos(alpha)];  10
% Rotation about the y-axis (Rv(beta))  12
Rv_beta = [cos(beta),  0, sin(beta);  13
0,          1, 0;  14
-sin(beta), 0, cos(beta)];  15
% Rotation about the z-axis (Rw(gamma))  17
Rw_gamma = [cos(gamma), -sin(gamma), 0;  18
sin(gamma),  cos(gamma), 0;  19
0,          0,          1];  20

```

$$\begin{pmatrix} -\dot{\beta} \cos(\gamma) \sin(\beta) - \dot{\gamma} \cos(\beta) \sin(\gamma) & \dot{\beta} \sin(\beta) \sin(\gamma) - \dot{\gamma} \cos(\beta) \cos(\gamma) & \dot{\beta} \cos(\beta) \\ \dot{\gamma} \sigma_{\gamma} - \dot{\alpha} \sigma_{\gamma} + \dot{\beta} \cos(\beta) \cos(\gamma) \sin(\alpha) & -\dot{\alpha} \sigma_{\gamma} - \dot{\gamma} \sigma_{\gamma} - \dot{\beta} \cos(\beta) \sin(\alpha) \sin(\gamma) & \dot{\beta} \sin(\alpha) \sin(\beta) - \dot{\alpha} \cos(\alpha) \cos(\beta) \\ \dot{\alpha} \sigma_{\gamma} + \dot{\gamma} \sigma_{\gamma} - \dot{\beta} \cos(\alpha) \cos(\beta) \cos(\gamma) & \dot{\alpha} \sigma_{\gamma} - \dot{\gamma} \sigma_{\gamma} + \dot{\beta} \cos(\alpha) \cos(\beta) \sin(\gamma) & -\dot{\alpha} \cos(\beta) \sin(\alpha) - \dot{\beta} \cos(\alpha) \sin(\beta) \end{pmatrix}$$

where

$$R_{uw} = \sigma_{\gamma} = \sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta)$$

$$\sigma_{\gamma} = \cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma)$$

$$\sigma_{\gamma} = \cos(\gamma) \sin(\alpha) + \cos(\alpha) \sin(\beta) \sin(\gamma)$$

$$\sigma_{\gamma} = \cos(\alpha) \sin(\gamma) + \cos(\gamma) \sin(\alpha) \sin(\beta)$$

$$\Omega = \begin{pmatrix} \dot{\alpha} + \dot{\gamma} \sin(\beta) \\ \dot{\beta} \cos(\alpha) - \dot{\gamma} \cos(\beta) \sin(\alpha) \\ \dot{\beta} \sin(\alpha) + \dot{\gamma} \cos(\alpha) \cos(\beta) \end{pmatrix}$$

در گام آخر، ماتریس تبدیل E با استفاده از تبدیل ژاکوبین و کمی ساده سازی به شکل زیر به دست می آید:

```
% Construct the matrix E by taking the Jacobian of Omega 1
    with respect to [alpha_dot; beta_dot; gamma_dot]
E_matrix = jacobian(Omega_vec, [alpha_dot, beta_dot, 2
    gamma_dot]);
3
% Simplify the matrix E 4
E_matrix = simplify(E_matrix); 5
6
% Display the matrix E(alpha, beta, gamma) 7
disp('Matrix E(alpha, beta, gamma):'); 8
disp(E_matrix); 9
```

$$E = \begin{pmatrix} 1 & 0 & \sin(\beta) \\ 0 & \cos(\alpha) & -\cos(\beta) \sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \cos(\beta) \end{pmatrix}$$

۸.۱.۱ پاسخ سوال ۷

۲.۱ توصیف مکان

۱.۲.۱ پاسخ سوال ۱

۱. سنسورهای موقعیت:

- سنسورهای خطی: این حسگرها می‌توانند موقعیت یک نقطه خاص را در راستای یک محور خطی اندازه‌گیری کنند. به عنوان مثال، می‌توان از سنسورهای مقاومتی یا پتانسیومتر استفاده کرد.

۲. انکودرهای چرخشی:

- این حسگرها می‌توانند زاویه‌های چرخشی مفاصل ربات را اندازه‌گیری کنند. با استفاده از اطلاعات زاویه‌ای و موقعیت‌های اولیه، می‌توان به طور غیرمستقیم موقعیت نقطه مورد نظر را محاسبه کرد. این روش معمولاً در ترکیب با مدل‌های سینماتیکی بازوهای رباتیک استفاده می‌شود.

۳. سنسورهای لیزری:

- لیزرهای فاصله‌سنج: این حسگرها می‌توانند فاصله تا یک نقطه مشخص را اندازه‌گیری کنند و به این ترتیب می‌توانند موقعیت را در فضای سه‌بعدی محاسبه کنند.

۴. سنسورهای دوربین:

- دوربین‌های استریو: با استفاده از دوربین‌های استریو و تحلیل تصویر، می‌توان عمق و موقعیت نقاط را در فضای سه‌بعدی اندازه‌گیری کرد.

- دوربین‌های $RGB-D$: این دوربین‌ها، اطلاعات رنگی و عمق را به طور همزمان ارائه می‌دهند و به تعیین موقعیت نقاط در فضای سه‌بعدی کمک می‌کنند.

۵. سنسورهای مادون قرمز:

- این سنسورها می‌توانند با اندازه‌گیری زمان رفت و برگشت نور مادون قرمز، فاصله تا یک نقطه مشخص را محاسبه کنند.

در مجموع، هر یک از سنسورهایی که برای تعیین موقعیت استفاده می‌شوند را می‌توان در بازوهای رباتیک مورد استفاده قرار داد. اما استفاده از انکودر ها برای اندازه‌گیری زوایای بازوی ربات، و استفاده از دوربین‌های با کیفیت تصویر بالا و لیزر ها برای تشخیص موقعیت ابزار ربات نیز از جمله مواردی است که در کاربرد های رباتیک مورد استفاده قرار می‌گیرند.

۲.۲.۱ پاسخ سوال ۲

علاوه بر دستگاه دکارتی، سایر دستگاه‌های مختصات نظیر دستگاه استوانه‌ای و کروی نیز می‌توانند مورد استفاده قرار بگیرند. انتخاب دستگاه مختصات مورد استفاده، وابسته به ساختار حرکت ربات دارد. به عنوان مثال، برای ارزیابی حرکت کره‌ای چشم، استفاده از مختصات کروی سهولت بیشتری دارد. اما برای استفاده در ربات‌های صفحه‌ای، مطابق با ساختار آنان، از دستگاه دکارتی استفاده می‌شود. علاوه بر این دستگاه‌ها، استفاده از دستگاه مختصات همگن که یک محور بعد w مازاد بر مختصات دکارتی دارد نیز استفاده می‌شود. مزیت استفاده از این دستگاه مختصات، در بررسی حرکات انتقالی و دورانی جسم است.

۳.۲.۱ پاسخ سوال ۳

۱. در این قسمت برای به دست آوردن مختصات نقطه در دستگاه A پس از انجام دوران، باید ابتدا ماتریس دوران حاصل از دوران‌های ذکر شده محاسبه شده و سپس با ضرب این ماتریس در موقعیت نقطه در دستگاه B ، مختصات جدید به دست آید.

```
% Define symbolic angles alpha, beta, and gamma
```



```

syms alpha beta gamma                                     2
                                                         3
% Step 1: Define symbolic rotation matrices for each    4
axis
% Rotation about the z-axis (Rz(gamma))                5
Rz_gamma = [cos(gamma), -sin(gamma), 0;                 6
sin(gamma),  cos(gamma), 0;                             7
0,           0,           1];                             8
                                                         9
% Rotation about the y-axis (Ry(beta))                10
Ry_beta = [cos(beta),  0, sin(beta);                    11
0,           1, 0;                                       12
-sin(beta),  0, cos(beta)];                             13
                                                         14
% Rotation about the x-axis (Rx(alpha))                15
Rx_alpha = [1, 0,           0;                          16
0, cos(alpha), -sin(alpha);                             17
0, sin(alpha),  cos(alpha)];                             18
                                                         19
% Display each rotation matrix                        20
disp('Rotation matrix around Z-axis (Rz(gamma)):');     21
disp(Rz_gamma);                                          22
                                                         23
disp('Rotation matrix around Y-axis (Ry(beta)):');      24
disp(Ry_beta);                                          25

```

```

26
disp('Rotation matrix around X-axis (Rx(alpha)):'); 27
disp(Rx_alpha); 28
29
% Step 2: Multiply the matrices to get the final 30
rotation matrix
Rpry = simplify(Rz_gamma * Ry_beta * Rx_alpha); 31
32
% Display the final rotation matrix symbolically 33
disp('Final rotation matrix Rpry (alpha, beta, gamma): 34
);
disp(Rpry); 35
% Step 3: Assign numerical values to alpha, beta, gamma 36
and compute the final numeric matrix
alpha_val = deg2rad(30); % 30 degrees 37
beta_val = deg2rad(60); % 60 degrees 38
gamma_val = deg2rad(0); % 0 degrees 39
40
% Substitute numerical values into the final rotation 41
matrix
Rpry_numeric = double(subs(Rpry, {alpha, beta, gamma}, 42
{alpha_val, beta_val, gamma_val}));
43
% Display the final numerical result 44
fprintf('Final rotation matrix with numerical values (45

```

```

alpha = %.2f , beta = %.2f, gamma = %.2f):\n',
alpha_val, beta_val, gamma_val);
disp(Rpry_numeric);
R_num = Rpry_numeric;

```

$$R_{rpy} = \begin{pmatrix} \cos(\beta) \cos(\gamma) & \cos(\gamma) \sin(\alpha) \sin(\beta) - \cos(\alpha) \sin(\gamma) & \sin(\alpha) \sin(\gamma) + \cos(\alpha) \cos(\gamma) \sin(\beta) \\ \cos(\beta) \sin(\gamma) & \cos(\alpha) \cos(\gamma) + \sin(\alpha) \sin(\beta) \sin(\gamma) & \cos(\alpha) \sin(\beta) \sin(\gamma) - \cos(\gamma) \sin(\alpha) \\ -\sin(\beta) & \cos(\beta) \sin(\alpha) & \cos(\alpha) \cos(\beta) \end{pmatrix}$$

با جایگذاری مقادیر α ، β و γ در ماتریس فوق خواهیم داشت:

$$R_{rpy} = \begin{pmatrix} 0.5000 & 0.4330 & 0.7500 \\ 0 & 0.8660 & -0.5000 \\ -0.8660 & 0.2500 & 0.4330 \end{pmatrix}$$

در نهایت با جایگذاری مختصات ابتدایی نقطه در دستگاه B و تنظیم ماتریس انتقال با مقادیر صفر، مختصات نهایی نقطه در دستگاه B را به دست می آوریم

```

% Define numerical values for the translation vector P1
(Example values)
P_num = [0; 0; 0]; % Translation along x, y, and z
% Define numerical values for the point p_B in local
frame B (Example values)
p_B_num = [1; 2; 3]; % Coordinates of the point in
frame B

```

6	
7	<i>% Step 6: Calculate the position in global frame A</i>
	<i>numerically</i>
8	<code>P_in_A_numeric = P_num + R_num * p_B_num;</code>
9	
10	<i>% Step 7: Display the numerical result</i>
11	<code>disp('Position of the point in the global coordinate</code>
	<code>system (numerical):');</code>
12	<code>disp(P_in_A_numeric);</code>

در نهایت، مختصات نقطه ی P در دستگاه A به صورت زیر به دست می آید.

$$A_P = \begin{pmatrix} ۳٫۶۱۶۰ \\ ۰٫۲۳۲۱ \\ ۰٫۹۳۳۰ \end{pmatrix}$$

۲. در این قسمت، مشابه مرحله ی قبل عمل می کنیم. با این تفاوت که برای به دست آوردن ماتریس دوران، باید تغییرات زاویه را نسبت به محور های متحرک در نظر بگیریم. در این بخش برای سادگی، از تبدیل Ruvw استفاده می کنیم.

1	<i>% Define symbolic angles alpha, beta, and gamma</i>
2	<code>syms alpha beta gamma</code>
3	
4	<i>% Step 1: Define symbolic rotation matrices for each</i>
	<i>axis</i>
5	<i>% Rotation about the x-axis (Ru(alpha))</i>
6	<code>Ru_alpha = [1, 0, 0;</code>

```

0, cos(alpha), -sin(alpha); 7
0, sin(alpha), cos(alpha)]; 8
9
% Rotation about the y-axis (Rv(beta)) 10
Rv_beta = [cos(beta), 0, sin(beta); 11
0, 1, 0; 12
-sin(beta), 0, cos(beta)]; 13
14
% Rotation about the z-axis (Rw(gamma)) 15
Rw_gamma = [cos(gamma), -sin(gamma), 0; 16
sin(gamma), cos(gamma), 0; 17
0, 0, 1]; 18
19
% Display each rotation matrix 20
disp('Rotation matrix around X-axis (Ru(alpha)):' ); 21
disp(Ru_alpha); 22
23
disp('Rotation matrix around Y-axis (Rv(beta)):' ); 24
disp(Rv_beta); 25
26
disp('Rotation matrix around Z-axis (Rw(gamma)):' ); 27
disp(Rw_gamma); 28
29
% Step 2: Multiply the matrices to get the final 30
rotation matrix

```

$$R_{uvw} = \begin{pmatrix} \cos(\beta) \cos(\gamma) & -\cos(\beta) \sin(\gamma) & \sin(\beta) \\ \cos(\alpha) \sin(\gamma) + \cos(\gamma) \sin(\alpha) \sin(\beta) & \cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma) & -\cos(\beta) \sin(\alpha) \\ \sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta) & \cos(\gamma) \sin(\alpha) + \cos(\alpha) \sin(\beta) \sin(\gamma) & \cos(\alpha) \cos(\beta) \end{pmatrix}$$
$$R_{uvw} = \begin{pmatrix} \circ_{/}9329 & \circ_{/}670 & -\circ_{/}3540 \\ -\circ_{/}1999 & \circ_{/}9137 & -\circ_{/}3538 \\ \circ_{/}2997 & \circ_{/}4008 & \circ_{/}8657 \end{pmatrix}$$

```
% Define numerical values for the translation vector P1
    (Example values)

P_num = [0; 0; 0]; % Translation along x, y, and z

% Define numerical values for the point p_B in local
    frame B (Example values)

p_B_num = [1; 2; 3]; % Coordinates of the point in
```

```

    frame B
6
% Step 6: Calculate the position in global frame A
7
    numerically
P_in_A_numeric = P_num + R_num * p_B_num;
8
9
% Step 7: Display the numerical result
10
disp('Position of the point in the global coordinate
11
    system (numerical):');
disp(P_in_A_numeric);
12

```

در نهایت، مختصات نقطه ی P در دستگاه A به صورت زیر به دست می آید.

$$\mathbf{A}_P = \begin{pmatrix} ۰/۰۰۵۰ \\ ۰/۵۶۶۰ \\ ۳/۶۹۸۶ \end{pmatrix}$$

۳. عبارت پیش ضرب به معنای ضرب ماتریس ها از سمت چپ می باشد. بنابراین، در محاسبه ی ماتریس دوران، ابتدایی ترین دوران α در سمت راست نوشته می شود و ماتریس دومین دوران β از سمت راست در آن ضرب می شود و همین قضیه برای دوران γ نیز صادق است. برای دوران های حول محور های ثابت، از پیش ضرب استفاده می شود. حال آنکه برای محاسبه ی دوران حول محور های ثابت، از پس ضرب استفاده می شود؛ به این صورت که ماتریس اولین دوران α نوشته شده و سپس ماتریس دومین دوران β در سمت راست آن نوشته می شود.

کتاب نامه

Hei- R. Sadeghi, H. Bataleblu, A. Khorambakht, R. Dindarloo, M.R. Hassani, A. [۷]
dynamic and Kinematic al., et Khorasani, A. Hojati, N.S. Hasani, P. Iranfar, A. dari,
Con- International RSI th ۹ ۲۰۲۱ in ,*positioning micro Toward asist. arash of analysis*
IEEE. , ۲۰۲۱ , ۶۵-۵۹ pp. , (ICRoM) *Mechatronics and Robotics on ference*