



دانشگاه خواجه نصیرالدین طوسی  
دانشکده برق - گروه مکترونیک

تمرین درس رباتیک دوره کارشناسی ارشد

رشته مهندسی مکترونیک

عنوان

تمرین رباتیک

نگارش

علیرضا امیری

مهر ۱۴۰۳

## فصل ۱

### پاسخ سوالات سری اول

#### ۱.۱ توصیف جهت گیری

##### ۱.۱.۱ پاسخ سوال ۱

- ژيروسکوپ

ژيروسکوپ سرعت زاویه‌ای (نرخ چرخش) را براساس اثر کوریولیس اندازه‌گیری می‌کند. ژيروسکوپ‌ها معمولاً داده‌های جهت‌گیری را در محورهای ثابت  $(X, Y, Z)$  با انتگرال‌گیری از سرعت زاویه‌ای در طول زمان ارائه می‌کنند. این داده‌ها می‌توانند با استفاده از زاویه‌های اولیه بیان شوند. با این حال، استفاده از ژيروسکوپ‌ها به‌تنهایی ممکن است باعث خطای انباشته در طول زمان شود.

- شتاب‌سنج

شتاب‌سنج شتاب حرکت را نسبت به گرانش زمین اندازه‌گیری کرده و تخمینی از شیب بر اساس کشش گرانش بر روی محورهای سنسور فراهم می‌کند. بنابراین می‌توان گفت خروجی آن در محورهای ثابت  $(X, Y, Z)$  قرار دارد و هنگامی که در حالت سکون است، زوایای شیب را نسبت به بردار گرانش فراهم می‌کند.

- مگنومتر

این حسگر قدرت و جهت میدان مغناطیسی زمین را برای محاسبه جهت گیری نسبت به شمال مغناطیسی اندازه گیری می کند. مگنتومتر داده های جهت گیری را به صورت زاویه های اوایلر (مخصوصاً زاویه yaw) ارائه می دهد، زمانی که با داده های شتاب سنج ترکیب می شود تا یک جهت گیری کامل در ۳ محور ایجاد شود.

### • واحد اندازه گیری اینرسی (IMU)

این حسگر که به طور گسترده برای تعیین جهت گیری جسم مورد استفاده قرار می گیرد، ترکیبی از شتاب سنج،ژیروسکوپ و گاهی اوقات مگنتومتر برای ارائه تصویر کامل تری از حرکت و جهت گیری یک جسم دارد. IMU ها معمولاً از محوره های ثابت استفاده می کنند و می توانند خروجی به صورت زاویه های اوایلر ارائه دهند. برخی ها IMU از کواترنیون ها استفاده می کنند که از مسائل هم محوری و قفل شدن گیمبال در زوایای اوایلر اجتناب می کند.

### • حسگر شیب (انکلی نومتر)

حسگر شیب زاویه شیب نسبت به گرانش را تشخیص می دهد. خروجی این حسگر به طور معمول در محوره های ثابت با توجه به گرانش است، و زوایای اوایلر را به طور مستقیم ارائه می دهد.

## ۲.۱.۱ پاسخ سوال ۲

سنسور IMU به دلیل مجهز بودن به حسگر های شتاب سنج، می تواند در حالت سکون زوایای گردش اوایلر را محاسبه کند. به طور خاص، شتاب سنج نیروی گرانش را که در محوره های X و Y حس می کند اندازه گیری کرده و از این داده ها برای محاسبه زوایای شیب استفاده می کند.

علاوه بر این، با در اختیار داشتن ژیروسکوپ، که سسرعت زاویه ای را به دست می دهد، می توان زاویه ی جسم را به دست آورد. برای این کار لازم است با در اختیار داشتن زاویه ی اولیه ی نقطه ی مورد نظر و با انتگرال گیری از سرعت زاویه ای، زاویه را محاسبه کرد.

## ۳.۱.۱ پاسخ سوال ۳

وجود نویز در اندازه گیری ماتریس دوران، منجر به ذخیره ی اشتباه مقدار عددی بعضی از درایه ها خواهد شد. بنابراین، برای اصلاح ماتریس دوران، باید بر اساس ویژگی های اساسی این ماتریس، مقادیر اشتباه را پیدا و تصحیح کرد. از جمله ی این ویژگی ها می توان به این موارد اشاره کرد:

۱. یکامتعامد

۲. دترمینان برابر ۱ باشد.

۳. مقدار وارون ماتریس با ترانهاده ی ماتریس برابر باشد.

۴. یک مقدار ویژه برابر ۱ داشته باشد.

## ۴.۱.۱ پاسخ سوال ۴

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

$$1 = \sqrt{x^2 + 0.6046^2 + 0.977^2} \Rightarrow x = \pm 0.7905$$

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

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

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 پارامتر ها 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) 9

```

در نتیجه:

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

• محاسبه زوایای اوایلر حول محور ثابت:

```

1.3.2: Inverse Rpry 1
2
3
4
% Define the rotation matrix R with numerical 5
values
disp(R) 6
% Calculate beta 7
beta = atan2(-R(3,1), sqrt(R(1,1)^2 + R(2,1)^2)); 8
9
% Calculate gamma 10

```

```

gamma = atan2(R(2,1) / cos(beta), R(1,1) / cos(beta)11
    ));
12
% Calculate alpha 13
alpha = atan2(R(3,2) / cos(beta), R(3,3) / cos(beta)14
    ));
15
% Display the calculated angles in radians 16
fprintf('Calculated angles: alpha = %.2f rad, beta17
    = %.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, beta25
    = %.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

1.3.3: Inversw Ruvw

We calculate Euler angles (alpha, beta, gamma) from a given rotation matrix R.

*% Define the rotation matrix R with numerical values*

disp(R)

*% Calculate beta*

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

*% Calculate alpha*

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

*% Calculate gamma*

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

*% Display the calculated angles in radians*

fprintf('Calculated angles: alpha = %.2f rad, beta = %.2f rad, gamma = %.2f rad\n', alpha, beta, gamma);

```

% 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);

```

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

• محاسبه زوایای اوایلر حول محور متحرک: Rvwv

```

1
1.3.4: Inverse Rvwv 2
In this section we calculate the Euler 3
    angels given a Rotation Matrix in the
    system of wvw.
% Define the rotation matrix R with 4
    numerical values
disp(R) 5
6
% Calculate beta 7
beta = atan2(sqrt(R(3,1)^2 + R(3,2)^2),R 8
    (3,3));

```



```

9
10 % Calculate alpha
11 alpha = atan2(R(2,3) / sin(beta), R(1,3) /
    sin(beta));
12
13 % Calculate gamma
14 gamma = atan2(R(3,2) / sin(beta), -R(3,1) /
    sin(beta));
15
16 % Display the calculated angles in radians
17 fprintf('Calculated angles: alpha = %.2f
    rad, beta = %.2f rad, gamma = %.2f rad\n
    ', alpha, beta, gamma);
18
19 % Optionally, convert radians to degrees
20 alpha_deg = rad2deg(alpha);
21 beta_deg = rad2deg(beta);
22 gamma_deg = rad2deg(gamma);
23
24 % Display the angles in degrees
25 fprintf('Calculated angles: alpha = %.2f
    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 اوایلر حول محور متحرک:  $R_{wuw}$

```

1
2
1.3.5: Inverse  $R_{wuw}$ 
3
In this section we calculate the Euler angels given
4
a Rotation Matrix in the system of  $wuw$ .
5
% Define the rotation matrix R with numerical
6
values
7
disp(R)
8
% Calculate beta
9
beta = atan2(sqrt(R(3,1)^2 + R(3,2)^2), R(3,3));
10
11
% Calculate alpha
12
alpha = atan2(R(1,3) / sin(beta), -R(2,3) / sin(
13
    beta));
14
15
% Calculate gamma
16
gamma = atan2(R(3,1) / sin(beta), R(3,2) / sin(beta
17
    ));
18
19
% Display the calculated angles in radians
20
fprintf('Calculated angles: alpha = %.2f rad, beta
21
    = %.2f rad, gamma = %.2f rad\n', alpha, beta,
22
    gamma);
23
24
% Optionally, convert radians to degrees
25
alpha_deg = rad2deg(alpha);
26

```

```

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
1.3.6: Inverse Screw 2
% Step 1: Define the rotation matrix R (example) 3
disp(R) 4
5
% Step 2: Calculate the rotation angle theta using the 6
trace of R
theta = acos((trace(R) - 1) / 2); 7
8
% Step 3: Check if sin(theta) is zero (to avoid 9
division by zero)
if sin(theta) == 0 10
error('Singularity encountered: sin(theta) is zero. 11
      Unable to compute the axis.');
```

```

end
12
13
% Step 4: Calculate the components of the screw axis
14
    = [sx, sy, sz]
sx = (R(3,2) - R(2,3)) / (2 * sin(theta));
15
sy = (R(1,3) - R(3,1)) / (2 * sin(theta));
16
sz = (R(2,1) - R(1,2)) / (2 * sin(theta));
17
18
% Step 5: Display the calculated screw axis and
19
    rotation angle
fprintf('Calculated rotation angle (theta) in radians:
20
    %.4f\n', theta);
fprintf('Calculated rotation angle (theta) in degrees:
21
    %.4f\n', rad2deg(theta));
fprintf('Screw axis vector s = [sx, sy, sz]: [%.4f, %.4f,
22
    %.4f]\n', sx, sy, sz);
23
% Optional: Normalize the screw axis (if you want the
24
    unit vector)
s_magnitude = sqrt(sx^2 + sy^2 + sz^2);
25
s_normalized = [sx, sy, sz] / s_magnitude;
26
fprintf('Normalized screw axis vector s: [%.4f, %.4f,
27
    %.4f]\n', s_normalized(1), s_normalized(2),
    s_normalized(3));
28

```

```

%% Step 6: Calculate the theta vector (theta_x, theta_y, theta_z)
    , 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);

```

$$\theta = 30.00^\circ,$$

$$\mathbf{S} = [s_x, s_y, s_z] : [0.8255, 0.2009, 0.5273]$$

$$\vec{\theta} = [\theta_x, \theta_y, \theta_z] : [1.0085, 0.2454, 0.6442]$$

۳. محاسبه نمایش چهارگان ماتریس دوران R:

```

1
% Step 1: Define the rotation matrix R (example) 2
disp(R) 3
4
% Step 2: Calculate the scalar part of the quaternion 5(
    e4)
e4 = 0.5 * sqrt(1 + R(1,1) + R(2,2) + R(3,3)); 6
7
% Step 3: Calculate the vector components of the 8

```

```

    quaternion
e1 = (R(3,2) - R(2,3)) / (4 * e4);    % e1          9
e2 = (R(1,3) - R(3,1)) / (4 * e4);    % e2          10
e3 = (R(2,1) - R(1,2)) / (4 * e4);    % e3          11
                                         12
% Step 4: Display the quaternion as a vector [e1, e2, 13
e3, e4]
quaternion_from_R = [e1, e2, e3, e4];  14
disp('Quaternion vector [e1, e2, e3, e4]:');  15
disp(quaternion_from_R);                16

```

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

### ۵.۱.۱ 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 ترانهاد 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:

$$\mathbf{u} \times \mathbf{v} = \begin{pmatrix} -\frac{\sqrt{2} k \sin(t)}{2} \\ \frac{\sqrt{2} k \sin(t)}{2} \\ -\frac{\sqrt{2} k \sin(t)}{2} - a \sin(t) + \frac{\sqrt{2} k}{2} \end{pmatrix}$$

$$\mathbf{v} \times \mathbf{w} = \begin{pmatrix} \frac{\sqrt{2} k (d \cos(t) + c \sin(t))}{2} \\ -a d - \frac{\sqrt{2} b k \sin(t)}{2} \\ a c - \frac{\sqrt{2} b k \cos(t)}{2} \end{pmatrix}$$

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

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.

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

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

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

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

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

### ۶.۱.۱ پاسخ سوال ۶

با در اختیار داشتن مقادیر زوایای اوپلر، 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 2
    angles
syms alpha beta gamma real                                     3
syms alpha_dot beta_dot gamma_dot real % Time derivatives4
    of the angles
                                                                    5
% Step 1: Define symbolic rotation matrices for each axis 6
% 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
                                         11
% 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
                                         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 اویلر 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  $\omega$  را محاسبه می کنیم.

```

%% Section 1: Define the Euler Angles and 1
Rotation Matrices for UVW
% Define symbolic angles alpha, beta, gamma 2
for the Euler angles

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

5
% Step 1: Define symbolic rotation matrices 6
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
11
% 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

```

```

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

```

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

where

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

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

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

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

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

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

```

1 % Construct the matrix E by taking the Jacobian of Omega
  with respect to [alpha_dot; beta_dot; gamma_dot]
E_matrix = jacobian(Omega_vec, [alpha_dot, beta_dot,
2 gamma_dot]);
3
4 % Simplify the matrix E
E_matrix = simplify(E_matrix);
5

```

	6
<code>% Display the matrix E(alpha, beta, gamma)</code>	7
<code>disp('Matrix E(alpha, beta, gamma):');</code>	8
<code>disp(E_matrix);</code>	9

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

۷.۱.۱ 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 دوران های 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      1
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); 46
47
R_num = Rpry_numeric; 48

```

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

با جایگذاری مقادیر  $\alpha$ ،  $\beta$  و  $\gamma$  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 2

```



```

3
4 % Define numerical values for the point p_B in local
   frame B (Example values)
5 p_B_num = [1; 2; 3]; % Coordinates of the point in
   frame B
6
7 % Step 6: Calculate the position in global frame A
   numerically
8 P_in_A_numeric = P_num + R_num * p_B_num;
9
10 % Step 7: Display the numerical result
11 disp('Position of the point in the global coordinate
   system (numerical):');
12 disp(P_in_A_numeric);

```

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 % Define symbolic angles alpha, beta, and gamma
2 syms alpha beta gamma

```

```

3
% Step 1: Define symbolic rotation matrices for each 4
axis
% Rotation about the x-axis (Ru(alpha)) 5
Ru_alpha = [1, 0, 0; 6
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
Ruvw = simplify(Ru_alpha * Rv_beta * Rw_gamma); 31
32
% Display the final rotation matrix symbolically 33
disp('Final rotation matrix Rpry (alpha, beta, gamma): 34
);
disp(Ruvw); 35

```

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

با جایگذاری مقادیر  $\alpha$ ،  $\beta$  و  $\gamma$  در ماتریس فوق خواهیم داشت:

$$R_{uvw} = \begin{pmatrix} 0.9329 & 0.0670 & -0.3540 \\ -0.1999 & 0.9137 & -0.3538 \\ 0.2997 & 0.4008 & 0.8657 \end{pmatrix}$$

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 Pl
(Example values)

```

```

P_num = [0; 0; 0]; % Translation along x, y, and z 2
3
% Define numerical values for the point p_B in local 4
frame B (Example values)
p_B_num = [1; 2; 3]; % Coordinates of the point in 5
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  $\alpha$  در سمت راست نوشته می شود و ماتریس دومین دوران  $\beta$  از سمت راست در آن ضرب می شود و همین قضیه برای دوران  $\gamma$  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  $\alpha$  نوشته شده و سپس ماتریس دومین دوران  $\beta$  در

سمت راست آن نوشته می شود.