دانشگاه خواجه نصیرالدین طوسی

دانشکده برق- گروه مکاترونیک

تمرین درس رباتیک دوره کارشناسی ارشد

رشته مهندسی مکاترونیک

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

- **ژیروسکوپ**

ژیروسکوپ سرعت زاویه‌ای (نرخ چرخش) را براساس اثر کوریولیس اندازه‌گیری می‌کند. ژیروسکوپ‌ها 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 اویلر بیان شوند. با این حال، استفاده از ژیروسکوپ‌ها به‌تنهایی ممکن است باعث خطای انباشته در طول زمان شود.

- **شتاب‌سنج**

شتاب سنج شتاب حرکت را نسبت به گرانش زمین اندازه‌گیری کرده و تخمینی از شیب بر اساس کشش گرانش بر روی محورهای سنسور فراهم می‌کند. بنابراین می توان 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.

- **مگنتومتر**

این حسگر قدرت و جهت میدان مغناطیسی زمین را برای محاسبه جهت‌گیری نسبت به شمال مغناطیسی اندازه‌گیری می‌کند. مگنتومتر داده‌های جهت‌گیری را به صورت زاویه‌های اویلر (مخصوصاً زاویه yaw) ارائه می‌دهد، زمانی که با داده‌های شتاب‌سنج ترکیب می‌شود تا یک جهت‌گیری کامل در ۳ محور ایجاد شود.

- **واحد اندازه‌گیری اینرسی (IMU)**

این حسگر که به طور گسترده برای تعیین جهتگیری جسم مورد استفاده قرار می گیرد، ترکیبی از شتاب‌سنج، ژیروسکوپ و گاهی اوقات مگنتومتر برای ارائه تصویر کامل‌تری از حرکت و جهت‌گیری یک جسم دارد.

IMU ها معمولاً از محورهای ثابت استفاده می‌کنند و می‌توانند خروجی به صورت زاویه‌های اویلر ارائه دهند. برخی هاIMU از کواترنیون‌ها استفاده می‌کنند که از مسائل هم‌محوری و قفل‌شدن گیمبال در زوایای اویلر اجتناب می‌کند.

- **حسگر شیب (انکلی‌نومتر)**

حسگر شیب زاویه شیب نسبت به گرانش را تشخیص می‌دهد. خروجی این حسگر به طور معمول در محورهای ثابت با توجه به گرانش است، و زوایای اویلر را به طور مستقیم را ارائه می‌دهد.

۲.۱.۱ پاسخ سوال ۲

سنسور IMU به دلیل مجهز بودن به حسگر های شتابسنج، می تواند در حالت سکون زوایای گردش اویلر را محاسبه کند. به طور خاص، شتاب‌سنج نیروی گرانش را که در محورهای X و Y حس می‌کند اندازه‌گیری کرده و از این داده‌ها برای محاسبه زوایای شیب استفاده می‌کند.

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، v و w را بررسی کرده و مقدار مجهول را با استفاده از این رابطه مشخص می کنیم.

$$۱ = \sqrt{x^۲ + ۰/۶۰۴۶^۲ + ۰/۰۹۷۷^۲} \Rightarrow \quad x = \pm ۰/۷۹۰۵$$

$$۱ = \sqrt{۰/۳۸۶۴^۲ + ۰/۳۶۸۶^۲ + z^۲} \Rightarrow \quad z = \pm ۰/۸۴۵۴$$

$$۱ = \sqrt{۰/۴۷۵۲^۲ + y^۲ + ۰/۵۲۵۰^۲} \Rightarrow \quad y = \pm ۰/۷۰۶۰$$

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
   rotation matrix R

A = [0.7905, -0.3864, 0.4752;
0.6046, 0.3686, -0.7060;
0.0977, 0.8454, 0.5250]
det_A = det(A);

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

در نتیجه:

$$x = +۰/۷۹۰۵, \quad y = -۰/۷۰۶۰, \quad z = +۰/۸۴۵۴$$

- محاسبه زوایای اویلر 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);

% Optionally, convert radians to degrees
alpha_deg = rad2deg(alpha);
beta_deg = rad2deg(beta);
gamma_deg = rad2deg(gamma);

% Display the angles in degrees
fprintf('Calculated angles: alpha = %.2f deg, beta = %.2f deg, gamma = %.2f deg\n', alpha_deg, beta_deg, gamma_deg);
```

$$\alpha = ۵۸٫۱۶^\circ, \quad \beta = -۵٫۶۱^\circ, \quad \gamma = +۳۷٫۴۱^\circ$$

- 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
alpha_deg = rad2deg(alpha);
beta_deg = rad2deg(beta);
gamma_deg = rad2deg(gamma);

% Display the angles in degrees
fprintf('Calculated angles: alpha = %.2f deg, beta = %.2f deg, gamma = %.2f deg\n', alpha_deg, beta_deg, gamma_deg);
```

$$\alpha = ۵۳/۳۶^\circ, \quad \beta = ۲۸/۳۷^\circ, \quad \gamma = ۲۶/۰۵^\circ$$

- محاسبه زوایای اویلر حول محور متحرک :Rwvw

```

1.3.4: Inverse Rwvw
In this section we calculate the Euler angels given a Rotation Matrix in the system of wvw.
% 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));
```

```matlab

% 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 rad, beta = %.2f rad, gamma = %.2f rad\n', alpha, beta, gamma);

% Optionally, convert radians to degrees
alpha_deg = rad2deg(alpha);
beta_deg = rad2deg(beta);
gamma_deg = rad2deg(gamma);

% Display the angles in degrees
fprintf('Calculated angles: alpha = %.2f deg, beta = %.2f deg, gamma = %.2f deg\n', alpha_deg, beta_deg, gamma_deg);
```

$$\alpha = -۵۶٫۰۶^\circ, \quad \beta = ۵۸٫۳۳^\circ, \quad \gamma = ۹۶٫۵۹^\circ$$

- محاسبه زوایای اویلر حول محور متحرک Rwuw:

```

1.3.5: Inverse Rwuw
In this section we calculate the Euler angels given a Rotation Matrix in the system of wuw.
% 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(1,3) / sin(beta), -R(2,3) / sin(beta));

% Calculate gamma
gamma = atan2(R(3,1) / sin(beta), R(3,2) / sin(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
alpha_deg = rad2deg(alpha);
```

```
beta_deg = rad2deg(beta);
gamma_deg = rad2deg(gamma);

% Display the angles in degrees
fprintf('Calculated angles: alpha = %.2f deg, beta = %.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 s
    = [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, %.4
    f, %.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);
```

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

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

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

۳. محاسبه نمایش چهارگان ماتریس دوران R:

```

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

% Step 2: Calculate the scalar part of the quaternion (e4)
e4 = 0.5 * sqrt(1 + R(1,1) + R(2,2) + R(3,3));

% Step 3: Calculate the vector components of the
```

```
    quaternion
e1 = (R(3,2) - R(2,3)) / (4 * e4);    % e1
e2 = (R(1,3) - R(3,1)) / (4 * e4);    % e2
e3 = (R(2,1) - R(1,2)) / (4 * e4);    % e3

% Step 4: Display the quaternion as a vector [e1, e2,
    e3, e4]
quaternion_from_R = [e1, e2, e3, e4];
disp('Quaternion vector [e1, e2, e3, e4]:');
disp(quaternion_from_R);
```

$$\vec{\epsilon} = [\epsilon_{۱}, \epsilon_{۲}, \epsilon_{۳}, \epsilon_{۴}] = [۰٫۴۷۳۵, ۰٫۱۱۵۲, ۰٫۳۰۲۴, ۰٫۸۱۹۲]$$

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 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.

$$\begin{aligned} \mathbf{u} \times \mathbf{v} &= \mathbf{w}, \\ \mathbf{v} \times \mathbf{w} &= \mathbf{u}, \\ \mathbf{w} \times \mathbf{u} &= \mathbf{v} \end{aligned} \tag{۱.۱}$$

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

$$R = \begin{pmatrix} \cos(t) & a & b \\ \sin(t) & \frac{\sqrt{۲}\,k\,\cos(t)}{۲} & c \\ ۰ & -\frac{\sqrt{۲}\,k\,\sin(t)}{۲} & d \end{pmatrix}$$

با قرار دادن ستون های این ماتریس به عنوان u، v w و محاسبه ی ضرب خارجی آنها خواهیم داشت:

$$\mathbf{u} \times \mathbf{v} = \begin{pmatrix} -\frac{\sqrt{۲}\,k\sin(t)^{۲}}{۲} \\ \frac{\sqrt{۲}\,k\,\sin(۲\,t)}{۴} \\ -\frac{\sqrt{۲}\,k\sin(t)^{۲}}{۲} - a\,\sin(t) + \frac{\sqrt{۲}\,k}{۲} \end{pmatrix}$$

$$\mathbf{v} \times \mathbf{w} = \begin{pmatrix} \frac{\sqrt{۲}\,k\,(d\,\cos(t)+c\,\sin(t))}{۲} \\ -a\,d - \frac{\sqrt{۲}\,b\,k\,\sin(t)}{۲} \\ a\,c - \frac{\sqrt{۲}\,b\,k\,\cos(t)}{۲} \end{pmatrix}$$

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

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.

$$\mathbf{a} = \begin{pmatrix} -\frac{\sin(t)}{\sqrt{\sin(t)^{۲}+۱}} \\ \frac{\sin(t)}{\sqrt{\sin(t)^{۲}+۱}} \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} -\frac{\sin(t)^{۲}}{\sqrt{\sin(t)^{۲}+۱}} \\ \frac{\sin(t)^{۲}}{\sqrt{\sin(t)^{۲}+۱}} \end{pmatrix}$$

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

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

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

۶.۱.۱ 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)];

% Rotation about the z-axis (Rw(gamma))
Rw_gamma = [cos(gamma), -sin(gamma), 0;
sin(gamma),  cos(gamma), 0;
0,           0,           1];
```

$$R_{u_\alpha} = \begin{pmatrix} ۱ & ۰ & ۰ \\ ۰ & \cos(\alpha) & -\sin(\alpha) \\ ۰ & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$R_{v_\beta} = \begin{pmatrix} \cos(\beta) & ۰ & \sin(\beta) \\ ۰ & ۱ & ۰ \\ -\sin(\beta) & ۰ & \cos(\beta) \end{pmatrix}$$

$$R_{w_\gamma} = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & ۰ \\ \sin(\gamma) & \cos(\gamma) & ۰ \\ ۰ & ۰ & ۱ \end{pmatrix}$$

آنگاه ماتریس دوران حاصل از سه دوران حول محور های 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 ω را محاسبه 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)];
```

```matlab

% Rotation about the z-axis (Rw(gamma))
Rw_gamma = [cos(gamma), -sin(gamma), 0;
sin(gamma),  cos(gamma), 0;
0,           0,          1];
```

$$R_{uvw.} = \begin{pmatrix} -\dot{\beta}\,\cos(\gamma)\,\sin(\beta) - \dot{\gamma}\,\cos(\beta)\,\sin(\gamma) & \dot{\beta}\,\sin(\beta)\,\sin(\gamma) - \dot{\gamma}\,\cos(\beta)\,\cos(\gamma) & \dot{\beta}\,\cos(\beta) \\ \dot{\gamma}\,\sigma_2 - \dot{\alpha}\,\sigma_1 + \dot{\beta}\,\cos(\beta)\,\cos(\gamma)\,\sin(\alpha) & -\dot{\alpha}\,\sigma_3 - \dot{\gamma}\,\sigma_4 - \dot{\beta}\,\cos(\beta)\,\sin(\alpha)\,\sin(\gamma) & \dot{\beta}\,\sin(\alpha)\,\sin(\beta) - \dot{\alpha}\,\cos(\alpha)\,\cos(\beta) \\ \dot{\alpha}\,\sigma_4 + \dot{\gamma}\,\sigma_3 - \dot{\beta}\,\cos(\alpha)\,\cos(\beta)\,\cos(\gamma) & \dot{\alpha}\,\sigma_2 - \dot{\gamma}\,\sigma_1 + \dot{\beta}\,\cos(\alpha)\,\cos(\beta)\,\sin(\gamma) & -\dot{\alpha}\,\cos(\beta)\,\sin(\alpha) - \dot{\beta}\,\cos(\alpha)\,\sin(\beta) \end{pmatrix}$$

where

$$\sigma_1 = \sin(\alpha)\,\sin(\gamma) - \cos(\alpha)\,\cos(\gamma)\,\sin(\beta)$$

$$\sigma_2 = \cos(\alpha)\,\cos(\gamma) - \sin(\alpha)\,\sin(\beta)\,\sin(\gamma)$$

$$\sigma_3 = \cos(\gamma)\,\sin(\alpha) + \cos(\alpha)\,\sin(\beta)\,\sin(\gamma)$$

$$\sigma_4 = \cos(\alpha)\,\sin(\gamma) + \cos(\gamma)\,\sin(\alpha)\,\sin(\beta)$$

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

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

```matlab
% 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,
    gamma_dot]);

% Simplify the matrix E
E_matrix = simplify(E_matrix);
```

```
% Display the matrix E(alpha, beta, gamma)
disp('Matrix E(alpha, beta, gamma):');
disp(E_matrix);
```

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

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 می شوند را می توان در بازوهای رباتیک مورد استفاده قرار داد. اما استفاده از انکودر ها برای اندازه گیری زوایای بازوی ربات، و استفاده از دوربین های 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 می گیرند.

۲.۲.۱ پاسخ سوال ۲

علاوه بر دستگاه دکارتی، سایر دستگاه های مختصات نظیر دستگاه استوانه ای و کروی نیز می توانند مورد استفاده قرار بگیرند. انتخاب دستگاه مختصات مورد استفاده، وابسته به ساختار حرکت ربات دارد. به عنوان مثال، برای ارزیابی حرکت کره ی چشم، استفاده از مختصات کروی سهولت بیشتری دارد. اما برای استفاده در ربات های صفحه ای، مطابق با ساختار آنان، از دستگاه دکارتی استفاده می شود. علاوه بر این دستگاه ها، استفاده از دستگاه مختصات همگن که یک محور بعد w مازاد بر مختصات دکارتی دارد نیز استفاده می شود. 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
syms alpha beta gamma

% Step 1: Define symbolic rotation matrices for each axis
% Rotation about the z-axis (Rz(gamma))
Rz_gamma = [cos(gamma), -sin(gamma), 0;
sin(gamma),  cos(gamma), 0;
0,           0,          1];

% Rotation about the y-axis (Ry(beta))
Ry_beta = [cos(beta),  0, sin(beta);
0,           1, 0;
-sin(beta),  0, cos(beta)];

% Rotation about the x-axis (Rx(alpha))
Rx_alpha = [1, 0,           0;
0, cos(alpha), -sin(alpha);
0, sin(alpha),  cos(alpha)];

% Display each rotation matrix
```

```
disp('Rotation matrix around Z-axis (Rz(gamma)):');
disp(Rz_gamma);

disp('Rotation matrix around Y-axis (Ry(beta)):');
disp(Ry_beta);

disp('Rotation matrix around X-axis (Rx(alpha)):');
disp(Rx_alpha);

% Step 2: Multiply the matrices to get the final
    rotation matrix
Rpry = simplify(Rz_gamma * Ry_beta * Rx_alpha);

% Display the final rotation matrix symbolically
disp('Final rotation matrix Rpry (alpha, beta, gamma):'
    );
disp(Rpry);
% Step 3: Assign numerical values to alpha, beta, gamma
     and compute the final numeric matrix
alpha_val = deg2rad(30);  % 30 degrees
beta_val = deg2rad(60);   % 60 degrees
gamma_val = deg2rad(0);  % 0 degrees

% Substitute numerical values into the final rotation
    matrix
```

```
Rpry_numeric = double(subs(Rpry, {alpha, beta, gamma},
   {alpha_val, beta_val, gamma_val}));

% Display the final numerical result
fprintf('Final rotation matrix with numerical values (
   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:

$$R_{rpy} = \begin{pmatrix} ۰٫۵۰۰۰ & ۰٫۴۳۳۰ & ۰٫۷۵۰۰ \\ ۰ & ۰٫۸۶۶۰ & -۰٫۵۰۰۰ \\ -۰٫۸۶۶۰ & ۰٫۲۵۰۰ & ۰٫۴۳۳۰ \end{pmatrix}$$

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 P
   (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

% Step 6: Calculate the position in global frame A numerically
P_in_A_numeric = P_num + R_num * p_B_num;

% Step 7: Display the numerical result
disp('Position of the point in the global coordinate system (numerical):');
disp(P_in_A_numeric);
```

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

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

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.

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

```
% 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)];

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

% Display each rotation matrix
disp('Rotation matrix around X-axis (Ru(alpha)):' );
disp(Ru_alpha);

disp('Rotation matrix around Y-axis (Rv(beta)):' );
disp(Rv_beta);

```

```
disp('Rotation matrix around Z-axis (Rw(gamma)):' );
disp(Rw_gamma);

% Step 2: Multiply the matrices to get the final rotation matrix
Ruvw = simplify(Ru_alpha * Rv_beta * Rw_gamma);

% Display the final rotation matrix symbolically
disp('Final rotation matrix Rpry (alpha, beta, gamma):' );
disp(Ruvw);
```

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

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

$$R_{uvw} = \begin{pmatrix} ۰٫۹۳۲۹ & ۰٫۰۶۷۰ & -۰٫۳۵۴۰ \\ -۰٫۱۹۹۹ & ۰٫۹۱۳۷ & -۰٫۳۵۳۸ \\ ۰٫۲۹۹۷ & ۰٫۴۰۰۸ & ۰٫۸۶۵۷ \end{pmatrix}$$

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 P (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

% Step 6: Calculate the position in global frame A numerically
P_in_A_numeric = P_num + R_num * p_B_num;

% Step 7: Display the numerical result
disp('Position of the point in the global coordinate system (numerical):');
disp(P_in_A_numeric);
```

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

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

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.