



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 اول

۱.۱ توصیف جهت گیری

۱.۱.۱ پاسخ سوال ۱

۲.۱.۱ پاسخ سوال ۲

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

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

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

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

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 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:

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

• محاسبه زوایای اوپلر 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
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);

```

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

• محاسبه زوایای اویلر حول محور متحرک: Ruvw

```

1
2 1.3.3: Inversw Ruvw
3 We calculate Euler angles (alpha, beta,
4   gamma) from a given rotation matrix R.
5
6 % Define the rotation matrix R with
7   numerical values
8
9 disp(R)
10
11 % Calculate beta
12 beta = atan2(R(1,3), sqrt(R(1,1)^2 + R(1,2)
13   ^2));

```

```

9
% Calculate alpha 10
alpha = atan2(-R(2,3) / cos(beta), R(3,3) 11
    cos(beta));
12
% Calculate gamma 13
gamma = atan2(-R(1,2) / cos(beta), R(1,1) 14
    cos(beta));
15
% Display the calculated angles in radians 16
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);

```

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

- محاسبه زوایای اوایلر حول محور متحرک: R_{wvw}

```

1.3.4: Inverse Rvwv
In this section we calculate the Euler
    angles 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));
% 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);

```

```

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

```



```

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
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 s14
    = [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, %.42
        f, %.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_29
    , theta_z)
theta_x = theta * sx; 30
theta_y = theta * sy; 31
theta_z = theta * sz; 32
33
% Step 7: Display the theta vector 34
fprintf('Theta vector = [theta_x, theta_y, theta_z]: 35
        [%.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
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];
16 disp('Quaternion vector [e1, e2, e3, e4]:');
17 disp(quaternion_from_R);

```

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

۵.۱.۱ پاسخ سوال ۵

۶.۱.۱ پاسخ سوال ۶

۷.۱.۱ پاسخ سوال ۷