Name Solutions by KJK

Midterm Exam

MEAM 520, Introduction to Robotics
University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

November 5, 2013

You must take this exam independently, without assistance from anyone else. You may bring in a calculator and two 8.5"×11" sheets of notes for reference. Aside from these two pages of notes, you may not consult any outside references, such as the textbook or the Internet. Any suspected violations of Penn's Code of Academic Integrity will be reported to the Office of Student Conduct for investigation.

This exam consists of several problems. We recommend you look at all of the problems before starting to work. If you need clarification on any question, please ask a member of the teaching team. When you work out each problem, please show all steps and box your answer. On problems involving actual numbers, please keep your solution symbolic for as long as possible before converting to numbers at the end; this will make your work easier to follow and easier to grade. The exam is worth a total of 100 points, and partial credit will be awarded for the correct approach even when you do not arrive at the correct answer.

	Points	Score
Problem 1	20	
Problem 2	15	
Problem 3	10	
Problem 4	15	
Problem 5	40	
Total	100	

I agree to abide by the University of Pennsylvania Code of Academic Integrity during this exam. I pledge that all work is my own and has been completed without the use of unauthorized aid or materials.

Signature ______________________________

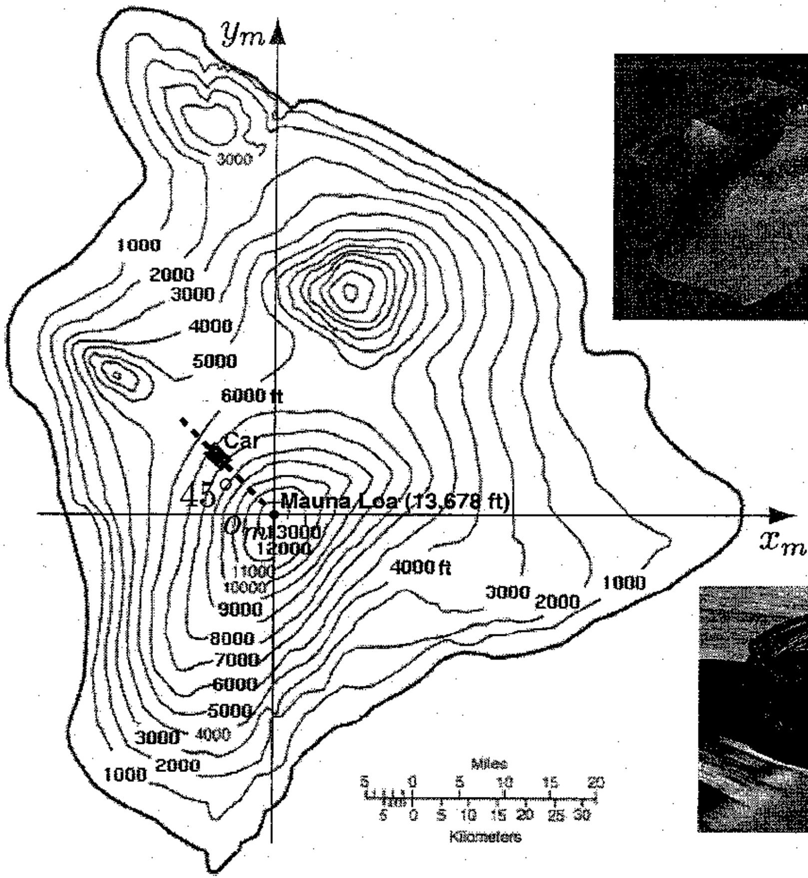
Date ______________________________

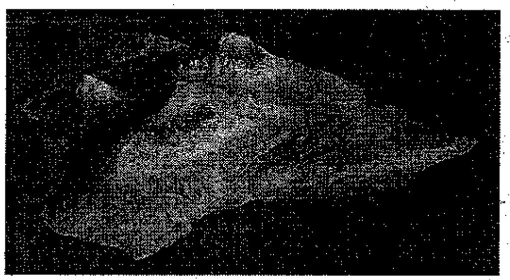
Problem 1: Homogeneous Transformations *(20 points)*

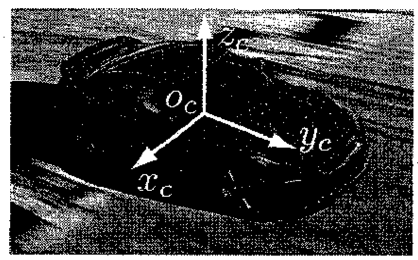
The diagram below depicts a topographic map of the island of Hawaii. As you can see in the upper inset image, this island has a tall mountain named Mauna Loa plus some other peaks.

Each contour line on the topographic map marks a line of constant elevation above sea level, which is the outer border of the island. Each successive line shows an elevation (height) increase of 1000 ft, as marked. We place o_m, the origin of our mountain coordinate frame, at the top of Mauna Loa (elevation of 13,678 ft) with the axes x_m and y_m defined as shown; z_m is positive out of the page. The x and y distance scale is below the diagram; 1000 ft $\approx$ 305 m and 1 mile = 5280 ft.

Imagine you are driving your car straight up Mauna Loa, as shown by the dashed line and the rectangle marked "Car". From overhead, your path makes a 45° angle with the negative x_m axis. We rigidly attach a coordinate frame $o_c x_c y_c z_c$ to your car, as shown in the lower right image.



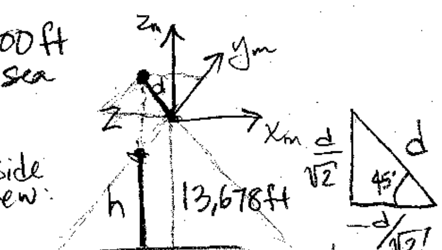




a. What is o_c^m, the **position of the car's center** expressed in the mountain frame? Please use either feet or meters for your calculations. *(6 points)*

h ≅ 7,500 ft above sea level.

Side view:



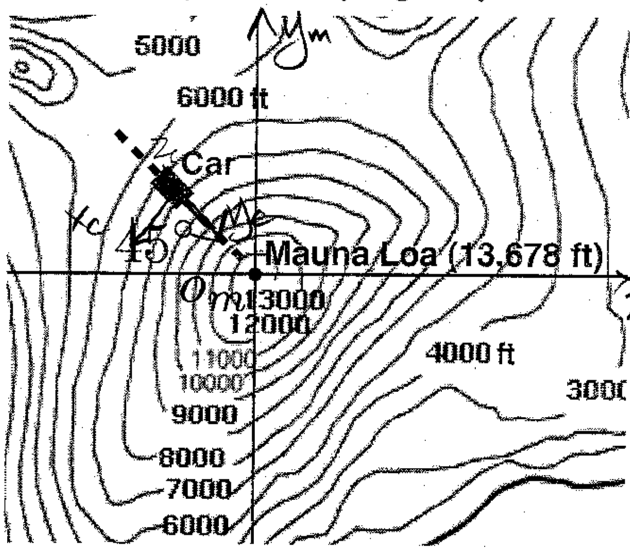
$$\boxed{O_c^m = \begin{bmatrix} -d/\sqrt{2} \\ d/\sqrt{2} \\ -13{,}678\text{ft} + h \end{bmatrix} \cong \begin{bmatrix} -37{,}335\text{ ft} \\ 37{,}335\text{ ft} \\ -6{,}178\text{ft} \end{bmatrix} \cong \begin{bmatrix} -11{,}387\text{m} \\ 11{,}387\text{m} \\ -1884\text{m} \end{bmatrix}}$$

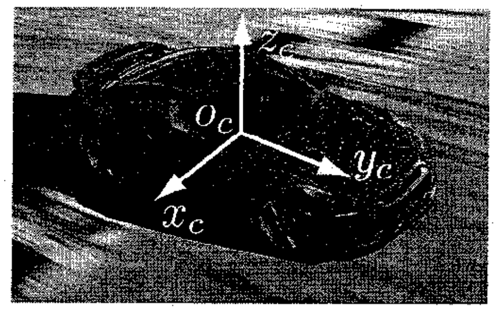
Car is about 10 miles from O_m

$d \cong 10\text{miles} \cdot \frac{5280\text{ft}}{\text{mile}} = 52{,}800\text{ft}$

* other numerical values are also reasonable, as the map does not provide precise coordinates especially in x and y directions.

b. What is R_c^m, the **orientation of the car's coordinate frame** expressed in the mountain frame? Clearly show how you get to your answer. *(12 points)*





Local slope of mountain 1000 ft up for ~3 miles

α 1000ft 3·5280ft

$\frac{1000\text{ ft}}{3\cdot 5280\text{ ft}} \cong \tan\alpha \quad \alpha \cong \tan^{-1}(0.06313)$

$\alpha \cong 3.61°$

Second rotate around fixed z axis -135° for angle of drive

First rotate car up around x_c by 3.61° for slope.

$$R_c^m = R_{z,-135°}\, R_{x,3.61°}$$

*Note that you get the same answer by first rotating -135° around z and then 3.61° about intermediate x axis, so post multiply...

$$R_c^m = \begin{bmatrix} \cos -135° & -\sin -135° & 0 \\ \sin -135° & \cos -135° & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 3.61° & -\sin 3.61° \\ 0 & \sin 3.61° & \cos 3.61° \end{bmatrix} = \begin{bmatrix} c{-135°} & -s{-135°}c3.61° & s{-135°}s3.61° \\ s{-135°} & c{-135°}c3.61° & -c{-135°}s3.61° \\ 0 & s3.61° & c3.61° \end{bmatrix}$$

Check: x_c is in $-x_m$ and $-y_m$ directions ✓
y_c is in $+x_m$, $-y_m$, $+z_m$ (small) directions ✓
z_c is in $+z_m$, $-x_m$ (small), $+y_m$ (small) directions ✓

$$\boxed{R_c^m = \begin{bmatrix} -0.707 & 0.7057 & -0.045 \\ -0.707 & -0.7057 & 0.045 \\ 0 & 0.063 & 0.998 \end{bmatrix}}$$

c. What is the **homogeneous transformation** H_c^m? *(2 points)*

* Other numerical values OK

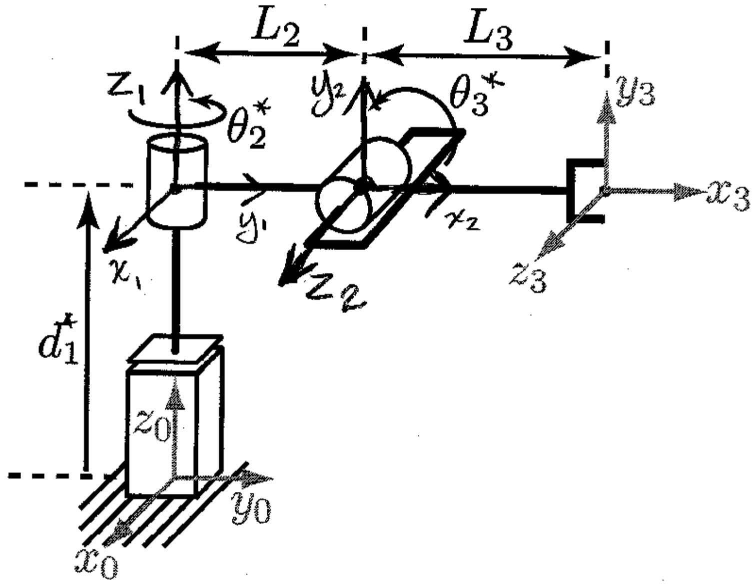
* Just need correct structure.

$$H_c^m = \begin{bmatrix} R_c^m & O_c^m \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

$$\boxed{H_c^m = \begin{bmatrix} -0.707 & 0.706 & -0.045 & -11{,}387\text{m} \\ -0.707 & -0.706 & 0.045 & 11{,}387\text{m} \\ 0 & 0.063 & 0.998 & -1884\text{m} \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Problem 2: DH Parameters *(15 points)*

The diagram below shows a PRR manipulator; this same robot appears in the last problem of this exam. The prismatic joint d_1 is drawn at a positive displacement, and the revolute joints θ_2 and θ_3 are drawn at zero. Positive joint directions are marked with arrows. The base frame is frame 0, and the end-effector frame is frame 3.



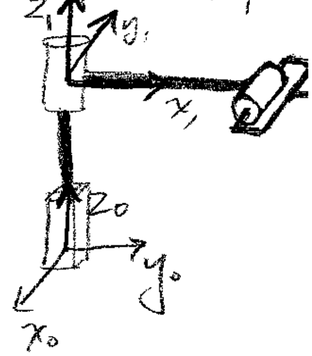
a. Follow the Denavit-Hartenberg conventions to **draw your intermediate frames** on the diagram above. *(3 points)*

b. Then fill in the table of **DH parameters** below. *(12 points)*

i	a	α	d	θ
1	0	0	d_1^*	0
2	L_2	$90°$	0	$\theta_2^* + 90°$
3	L_3	$0°$	0	θ_3^*

$90°$, θ_2^* is another correct solution

puts x_1

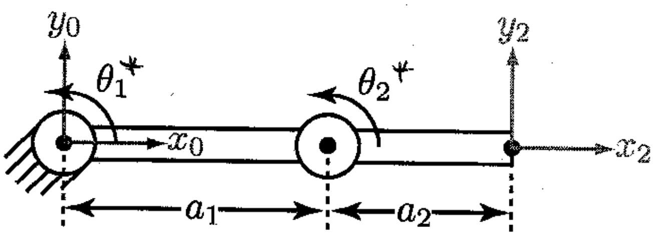


You did not need to use the * notation

Other solutions also possible.

Problem 3: More DH Parameters *(10 points)*

The diagram below shows a planar manipulator with two revolute joints. Both θ_1 and θ_2 are measured relative to the robot's previous link; the angles are zero in the depicted pose and increase when the links are rotated in the depicted directions. The base frame and the end-effector frame are defined in the drawing, with the base frame (frame 0) being stationary and the end-effector frame (frame 2) moving with the manipulator's distal link. Both z_0 and z_2 are positive out of the page. One correct set of DH parameters is listed below.



i	a	α	d	θ
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2

a. Write **another correct set of DH parameters** for this manipulator. They should yield the same T_2^0 matrix as the parameters above but differ in a non-trivial way. *(5 points)*

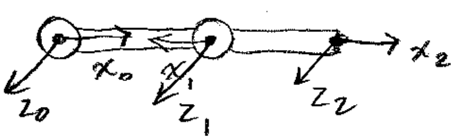
The key insight on this problem is that you can change the intermediate frame (frame 1) but must still follow DH rules.

i	a	α	d	θ
1	a_1	0	L	θ_1^*
2	a_2	0	$-L$	θ_2^*

puts the intermediate frame out of the plane.

b. Write a **third correct set of DH parameters** for this manipulator. They should yield the same T_2^0 matrix but differ in a new non-trivial way from both sets above. *(5 points)*

Don't forget this negative!



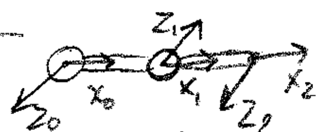
i	a	α	d	θ
1	$-a_1$	0	0	$\theta_1^* + 180°$
2	a_2	0	0	$\theta_2^* + 180°$

a_1 is along x_1. Flips the direction of x_1 – there are always two choices for x.

either or both of these could also be $\theta^* - 180°$ for the same effect.

A final trick is to flip the direction of the z_1 axis, since it's not specified, only the positive direction for θ_2.

i	a	α	d	θ
1	a_1	180°	0	θ_1^*
2	a_2	180°	0	$-\theta_2^*$



don't forget this negative sign.

either or both could be 180

Matlab code attached to prove equivalence.

Contents

- Define the problem.
- First alternative.
- Second alternative.
- Third alternative.

```
% Clear the workspace.
```

Define the problem.

```
% Define symbolic variables
syms a1 a2 theta1 theta2 L pi

% Calculate and simplify the correct transformation for the planar RR robot.
T02 = dh_kuchenbe(a1,0,0,theta1)*dh_kuchenbe(a2,0,0,theta2)
T02 = simple(T02)
```

```
T02 =

[ cos(theta1)*cos(theta2) - sin(theta1)*sin(theta2), - cos(theta1)*sin(theta2) - cos(theta2)*sin(thet
[ cos(theta1)*sin(theta2) + cos(theta2)*sin(theta1),   cos(theta1)*cos(theta2) - sin(theta1)*sin(thet
[                                                 0,
[                                                 0,


T02 =

[ cos(theta1 + theta2), -sin(theta1 + theta2), 0, a2*cos(theta1 + theta2) + a1*cos(theta1)]
[ sin(theta1 + theta2),  cos(theta1 + theta2), 0, a2*sin(theta1 + theta2) + a1*sin(theta1)]
[                    0,                     0, 1,                                        0]
[                    0,                     0, 0,                                        1]
```

First alternative.

```
% Calculate and simplify the first alternative, which puts frame 1 out of
% the plane of the robot by a distance L.  Since all the revolute axes are
% parallel, we can put the intermediate frame whereever we want along z1.
A = dh_kuchenbe(a1,0,L,theta1)*dh_kuchenbe(a2,0,-L,theta2)
A = simple(A)

% Subtract the two matrices to see if they are truly the same.
A - T02
```

```
A =

[ cos(theta1)*cos(theta2) - sin(theta1)*sin(theta2), - cos(theta1)*sin(theta2) - cos(theta2)*sin(thet
[ cos(theta1)*sin(theta2) + cos(theta2)*sin(theta1),   cos(theta1)*cos(theta2) - sin(theta1)*sin(thet
[                                                 0,
[                                                 0,
```

```
A =

[ cos(theta1 + theta2), -sin(theta1 + theta2), 0, a2*cos(theta1 + theta2) + a1*cos(theta1)]
[ sin(theta1 + theta2),  cos(theta1 + theta2), 0, a2*sin(theta1 + theta2) + a1*sin(theta1)]
[                    0,                     0, 1,                                        0]
[                    0,                     0, 0,                                        1]


ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
```

all zeros, so identical.

Second alternative.

```
% Calculate and simplify the second alternative, which flips the direction
% of x1 to point toward o2 instead of away from it.  We can always choose
% the direction of the x-axis to be toward or away from the previous frame.
% This change requires adding pi radians (180 degrees) to both thetas and
% also moving a negative a1 for the first x-step.
B = dh_kuchenbe(-a1,0,0,theta1+pi)*dh_kuchenbe(a2,0,0,theta2+pi)
B = simple(B)

% Subtract the two matrices to see if they are truly the same.
B - T02

% Either or both of the positive pi radians offsets can be negative.
B_alt = dh_kuchenbe(-a1,0,0,theta1-pi)*dh_kuchenbe(a2,0,0,theta2-pi);
B_alt = simple(B_alt)

% Subtract the two matrices to see if they are truly the same.
B_alt - T02
```

```
B =

[ cos(theta1)*cos(theta2) - sin(theta1)*sin(theta2), - cos(theta1)*sin(theta2) - cos(theta2)*sin(thet
[ cos(theta1)*sin(theta2) + cos(theta2)*sin(theta1),   cos(theta1)*cos(theta2) - sin(theta1)*sin(thet
[                                                 0,
[                                                 0,


B =

[ cos(theta1 + theta2), -sin(theta1 + theta2), 0, a2*cos(theta1 + theta2) + a1*cos(theta1)]
[ sin(theta1 + theta2),  cos(theta1 + theta2), 0, a2*sin(theta1 + theta2) + a1*sin(theta1)]
```

```
[                      0,                      0, 1,                                                  0]
[                      0,                      0, 0,                                                  1]


ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]


B_alt =

[ cos(theta1 + theta2), -sin(theta1 + theta2), 0, a2*cos(theta1 + theta2) + a1*cos(theta1)]
[ sin(theta1 + theta2),  cos(theta1 + theta2), 0, a2*sin(theta1 + theta2) + a1*sin(theta1)]
[                      0,                      0, 1,                                                  0]
[                      0,                      0, 0,                                                  1]


ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]     all zeros
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
```

Third alternative.

```
% Calculate and simplify the third alternative, which flips the direction
% of z1 to point into the page instead of toward it.  Doing so requires us
% to enter -theta2 as the joint angle for that joint, and it also requires
% both alphas to be positive or negative pi radians (180 degrees).  This
% alternative might seem to violate the DH rules, but it does yield the
% same final transformation matrix.
C = dh_kuchenbe(a1,pi,0,theta1)*dh_kuchenbe(a2,pi,0,-theta2)
C = simple(C)

% Subtract the two matrices to see if they are truly the same.
C - T02
```

```
C =

[ cos(theta1)*cos(theta2) - sin(theta1)*sin(theta2), - cos(theta1)*sin(theta2) - cos(theta2)*sin(thet
[ cos(theta1)*sin(theta2) + cos(theta2)*sin(theta1),   cos(theta1)*cos(theta2) - sin(theta1)*sin(thet
[                                                 0,
[                                                 0,
```

```
C =

[ cos(theta1 + theta2), -sin(theta1 + theta2), 0, a2*cos(theta1 + theta2) + a1*cos(theta1)]
[ sin(theta1 + theta2),  cos(theta1 + theta2), 0, a2*sin(theta1 + theta2) + a1*sin(theta1)]
[                    0,                     0, 1,                                        0]
[                    0,                     0, 0,                                        1]


ans =

[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
[ 0, 0, 0, 0]
```

all zeros.

Problem 4: Calculating a Trajectory *(15 points)*

We seek a trajectory $d(t)$ that will move a prismatic robot joint from position $d_0 = 4$ cm at time $t_0 = 0$ s to be at position $d_f = 8$ cm at time $t_f = 2$ s. The trajectory should have zero initial and final velocity. To avoid colliding with an obstacle in the environment, this trajectory must move through an intermediate point $d_i = 0$ cm at time $t_i = 1$ s. The trajectory must be continuous in position, velocity, and acceleration at the midpoint $t_i = 1$ s.

We will create this trajectory using two different cubic polynomials. Let the trajectory be described as follows:

$$\text{for } 0\,\text{s} \le t \le 1\,\text{s}, \quad d(t) = \alpha(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$\dot\alpha(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad \ddot\alpha(t) = 2a_2 + 6a_3 t$

$$\text{for } 1\,\text{s} \le t \le 2\,\text{s}, \quad d(t) = \beta(t) = b_0 + b_1(t - 1\,\text{s}) + b_2(t - 1\,\text{s})^2 + b_3(t - 1\,\text{s})^3$$

$\dot\beta(t) = b_1 + 2b_2(t-1s) + 3b_3(t-1s)^2 \quad \ddot\beta(t) = 2b_2 + 6b_3(t-1s)$

a. Write down all of the **independent equations** that the unknown coefficients ($a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$) must satisfy to obey the constraints listed above. *(8 points)*

$\alpha(t=0s) = \boxed{4cm = a_0}$ initial position

$\dot\alpha(t=0s) = \boxed{0 cm/s = a_1}$ init. vel.

$\alpha(t=1s) = \boxed{0cm = a_0 + a_1(1s) + a_2(1s)^2 + a_3(1s)^3}$ intermediat position.

$\beta(t=1s) = \boxed{0cm = b_0}$ intermed. position.

$\beta(t=2s) = \boxed{8cm = b_0 + b_1(1s) + b_2(1s)^2 + b_3(1s)^3}$ final position

$\dot\beta(t=2s) = \boxed{0 cm/s = b_1 + 2b_2(1s) + 3b_3(1s)^2}$ final velocity

$\dot\alpha(t=1s) = \dot\beta(t=1s)$

$\boxed{a_1 + 2a_2(1s) + 3a_3(1s)^2 = b_1}$ intermediate velocity

$\ddot\alpha(t=1s) = \ddot\beta(t=1s)$

$\boxed{2a_2 + 6a_3(1s) = 2b_2}$ intermed. acceleration

8 unknowns, So need 8 equations

b. Given the additional information that $a_3 = 11$ cm/s^3, $b_1 = 3$ cm/s, and $b_3 = -13$ cm/s^3, **solve for the other coefficients**. Include appropriate units. *(7 points)*

$\boxed{a_0 = 4cm}$

$\boxed{a_1 = 0 cm/s}$

$\boxed{b_0 = 0cm}$

$a_2 = ?$

$b_2 = ?$

$0 cm/s = b_1 + 2s\, b_2 + 3s^2 b_3$

$0 cm/s = (3 cm/s) + (2s) b_2 + (3s^2)(-13 cm/s^3)$

$-(2s) b_2 = 3 cm/s - 39 cm/s$

$-(2s) b_2 = -36 cm/s$

$b_2 = (36 cm/s)/(2s)$

$\boxed{b_2 = 18 cm/s^2}$

*Don't forget units

$0cm = a_0 + (1s)a_1 + (1s^2)a_2 + (1s^3)a_3$

$0cm = 4cm + (1s)(0 cm/s) + (1s^2)a_2 + (1s^3)(11 cm/s^3)$

$-4cm = 0cm + (1s^2)a_2 + 11cm$

$(1s^2)a_2 = -15cm$

$\boxed{a_2 = -15 cm/s^2}$

$a_0 = 4cm$	$b_0 = 0cm$
$a_1 = 0 cm/s$	$b_1 = 3 cm/s$
$a_2 = -15 cm/s^2$	$b_2 = 18 cm/s^2$
$a_3 = 11 cm/s^3$	$b_3 = -13 cm/s^3$

Matlab code attached to show the resulting trajectory

Table of Contents

Clear the workspace.

```
clear
```

Set coefficient values.

```
a0 = 4; % cm
a1 = 0; % cm/s
a2 = -15; % cm/s^2
a3 = 11; % cm/s^3

b0 = 0; % cm
b1 = 3; % cm/s
b2 = 18; % cm/s^2
b3 = -13; % cm/s^3
```

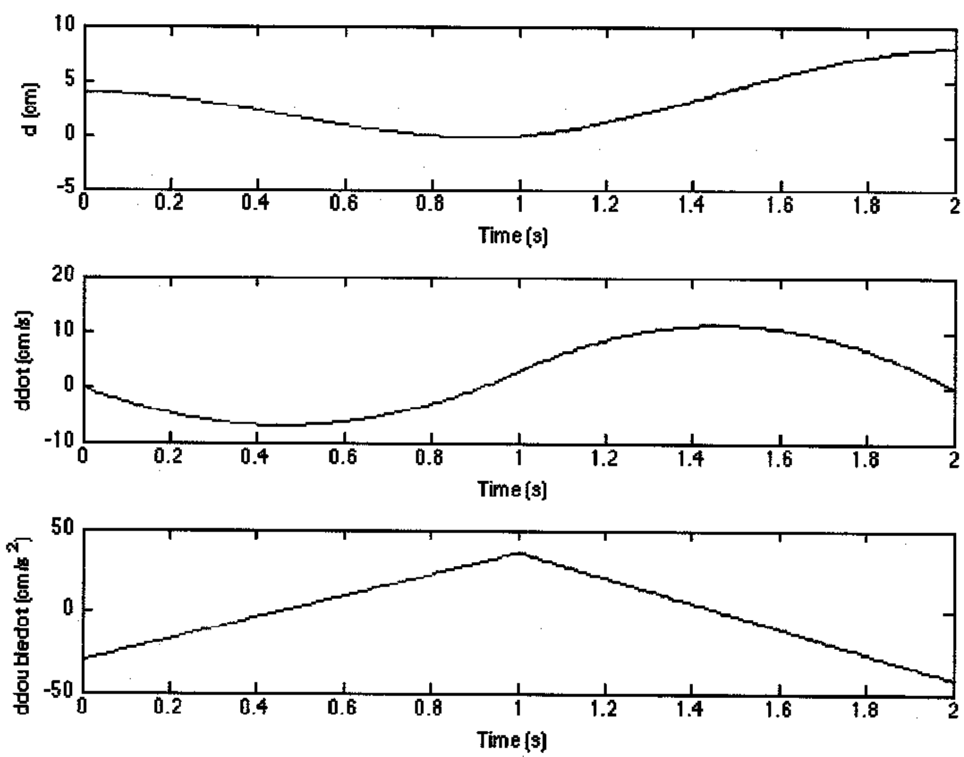
Calculate the trajectories.

```
% Create time vector from 0 s to 1 s.
tstep = 0.01;
t = (0:tstep:1)';

% Calculate alpha.
alpha = a0 + a1*t + a2*t.^2 + a3*t.^3;

% Store alpha's time vector.
talpha = t;

% Create time vector from 1 s to 2 s.
t = (1:tstep:2)';

% Calculate beta.
beta = b0 + b1*(t-1) + b2*(t-1).^2 + b3*(t-1).^3;

% Store beta's time vector.
tbeta = t;
```

Plot the trajectories.

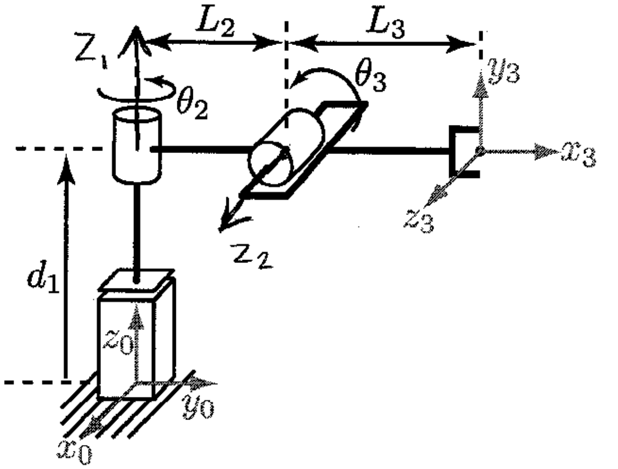
```
% Open figure.
figure(1)
```

```
clf

% Plot alpha and beta, both of which are cubic trajectories.
subplot(3,1,1)
plot(talpha,alpha,'r',tbeta,beta,'b')
xlabel('Time (s)')
ylabel('d (cm)')

% Plot first derivative (velocity) of both trajectories.
subplot(3,1,2)
plot(talpha,a1+2*a2*talpha+3*a3*talpha.^2,'r',tbeta,b1+2*b2*(tbeta-1)+3*b3*(tbe
xlabel('Time (s)')
ylabel('ddot (cm/s)')

% Plot second derivative (acceleration) of both.
subplot(3,1,3)
plot(talpha,2*a2+6*a3*talpha,'r',tbeta,2*b2+6*b3*(tbeta-1),'b')
xlabel('Time (s)')
ylabel('ddoubledot (cm/s^2)')
```



Published with MATLAB® 8.0

Problem 5: Velocity Kinematics and Singularities *(40 points)*



$$o_3^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sin\theta_2(L_2 + L_3\cos\theta_3) \\ \cos\theta_2(L_2 + L_3\cos\theta_3) \\ d_1 + L_3\sin\theta_3 \end{bmatrix}$$

$$= \begin{bmatrix} -s_2(L_2 + L_3c_3) \\ c_2(L_2 + L_3c_3) \\ d_1 + L_3s_3 \end{bmatrix}$$

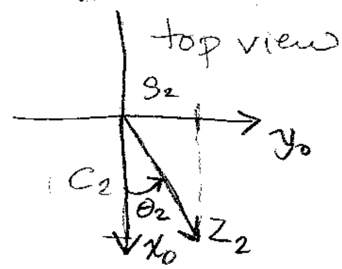
The diagram above shows a PRR manipulator; this same robot appears in the second problem of this exam. The prismatic joint d_1 is drawn at a positive displacement, and the revolute joints θ_2 and θ_3 are drawn at zero. Positive joint directions are marked with arrows. The end-effector frame is frame 3; the position of the end-effector's origin in frame 0 is written above.

a. What is the **angular velocity Jacobian** J_ω for this robot? *(4 points)*

$$\boxed{J_\omega = \begin{bmatrix} 0 & 0 & \cos(\theta_2) \\ 0 & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \end{bmatrix}}$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \text{prismatic} & z_1^0 & z_2^0 \end{array}$$

*Common error is to put $\begin{smallmatrix}1\\0\\0\end{smallmatrix}$ as third column, but z_2 rotates around z_1.

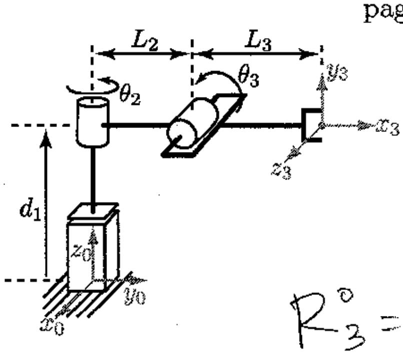


b. The robot is in the pose depicted above with $\dot{d}_1 = 0.1\,\text{m/s}$, $\dot{\theta}_2 = -0.2\,\text{rad/s}$, and $\dot{\theta}_3 = 0.3\,\text{rad/s}$. What is $\omega_{0,3}^0$, the **angular velocity of the end-effector frame** relative to frame 0, expressed in frame 0? *(3 points)*

$$\omega_{0,3}^0 = J_\omega \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & c2 \\ 0 & 0 & s2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.1\,\text{m/s} \\ -0.2\,\text{rad/s} \\ 0.3\,\text{rad/s} \end{bmatrix} = \begin{bmatrix} c2 \cdot 0.3\,\text{rad/s} \\ s2 \cdot 0.3\,\text{rad/s} \\ -0.2\,\text{rad/s} \end{bmatrix}$$

at pose shown, $\theta_2 = 0°$ so $s2 = 0$ $c2 = 1$

$$\boxed{\omega_{0,3}^0 = \begin{bmatrix} 0.3\,\text{rad/s} \\ 0\,\text{rad/s} \\ -0.2\,\text{rad/s} \end{bmatrix}}$$



$$o_3^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -s_2(L_2 + L_3 c_3) \\ c_2(L_2 + L_3 c_3) \\ d_1 + L_3 s_3 \end{bmatrix}$$

$R_3^0 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ from inspection

c. For the same conditions (the pose depicted above with $\dot{d}_1 = 0.1\,\mathrm{m/s}$, $\dot{\theta}_2 = -0.2\,\mathrm{rad/s}$, and $\dot{\theta}_3 = 0.3\,\mathrm{rad/s}$), what is $\frac{dR_3^0}{dt}$, the **time derivative of the rotation matrix** that represents the orientation of the end-effector frame relative to the base frame? *(6 points)*

$$\frac{dR_3^0}{dt} = S(\vec{\omega})\underbrace{R_3^0}_{R_3^0} = \begin{bmatrix} 0 & 0.2\,\mathrm{r/s} & 0 \\ -0.2\,\mathrm{r/s} & 0 & -0.3\,\mathrm{r/s} \\ 0 & 0.3\,\mathrm{r/s} & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\vec{\omega} = \begin{bmatrix} 0.3\,\mathrm{rad/s} \\ 0 \\ -0.2\,\mathrm{rad/s} \end{bmatrix}$$

$$\boxed{\frac{dR_3^0}{dt} = \begin{bmatrix} 0.2\,\mathrm{rad/s} & 0 & 0 \\ 0 & -0.3\,\mathrm{rad/s} & -0.2\,\mathrm{rad/s} \\ 0.3\,\mathrm{rad/s} & 0 & 0 \end{bmatrix}}$$

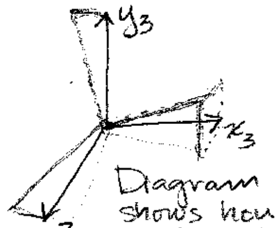


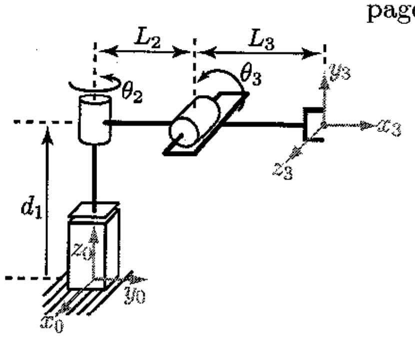
Diagram shows how unit vect are char

d. Calculate the **linear velocity Jacobian** J_v for this robot. *(8 points)*

$$o_3^0 = \begin{bmatrix} -S^2(L_2 + L_3 C_3) \\ C_2(L_2 + L_3 C_3) \\ d_1 + L_3 S_3 \end{bmatrix} \qquad J_v = \begin{bmatrix} \partial x/\partial d_1 & \partial x/\partial \theta_2 & \partial x/\partial \theta_3 \\ \partial y/\partial d_1 & \partial y/\partial \theta_2 & \partial y/\partial \theta_3 \\ \partial z/\partial d_1 & \partial z/\partial \theta_2 & \partial z/\partial \theta_3 \end{bmatrix}$$

$$\boxed{J_v = \begin{bmatrix} 0 & -C_2(L_2 + L_3 C_3) & L_3 S_2 S_3 \\ 0 & -S_2(L_2 + L_3 C_3) & -L_3 C_2 S_3 \\ 1 & 0 & L_3 C_3 \end{bmatrix}}$$

Check: in zero config: $J_v((0,0,0)^T) = \begin{bmatrix} 0 & -(L_2+L_3) & 0 \\ 0 & 0 & 0 \\ 1 & 0 & L_3 \end{bmatrix}$ ✓



$$o_3^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -s_2(L_2 + L_3c_3) \\ c_2(L_2 + L_3c_3) \\ d_1 + L_3s_3 \end{bmatrix}$$

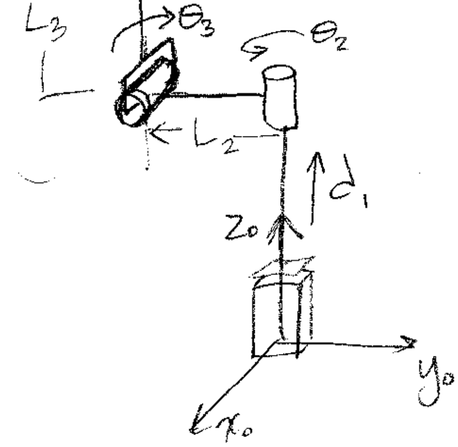
e. Imagine the robot is at $d_1 = 1\,\text{m}$, $\theta_2 = \pi\,\text{rad}$, and $\theta_3 = \pi/2\,\text{rad}$. What is v_3^0, the **linear velocity of the end-effector**, in terms of $\dot{d}_1$, $\dot{\theta}_2$, $\dot{\theta}_3$, L_2, and L_3? **Sketch** the manipulator in this configuration to check your answer graphically. *(4 points)*

$d_1 = 1\text{m}$

$\theta_2 = \pi \text{ rad}$ $\quad S2=0$, $C2=-1$

$\theta_3 = \pi/2 \text{ rad}$ $\quad S3=1$, $C3=0$

$$V_3^0 = J_V \cdot \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \qquad J_V = \begin{bmatrix} 0 & -(-1)(L_2+0) & 0 \\ 0 & 0 & -L_3(-1)(1) \\ 1 & 0 & 0 \end{bmatrix}$$



$$V_3^0 = \begin{bmatrix} 0 & L_2 & 0 \\ 0 & 0 & L_3 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \boxed{\begin{bmatrix} L_2\dot{\theta}_2 \\ L_3\dot{\theta}_3 \\ \dot{d}_1 \end{bmatrix} = V_3^0}$$

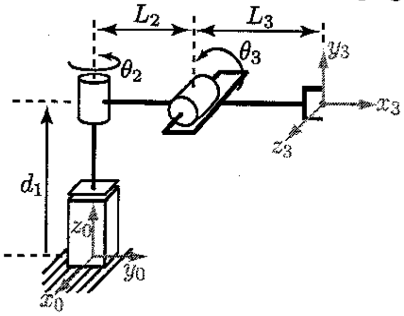
$\dot{\theta}_2 > 0$ moves $+x_0$, L_2 moment arm

$\dot{\theta}_3 > 0$ moves $+y_0$, L_3 moment arm $\quad d_1 \uparrow z_0$ ✓

f. Imagine you pre-multiplied J_v by R_0^3. **What could you calculate with the resulting matrix** $K = R_0^3 J_v$? *(3 points)*

$R_0^3 J_V$ Premultiplying by R_0^3 converts the output to be expressed in frame 3, the end-effector frame. It can be useful to specify the velocity of the end-effector in this way.

Also, $\det(R_0^3 J_V)$
$= \det(R_0^3)\det(J_V)$
$= 1 \cdot \det(J_V)$
Could be used to find singularities... Sometimes easier!



$$o_3^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -s_2(L_2 + L_3c_3) \\ c_2(L_2 + L_3c_3) \\ d_1 + L_3s_3 \end{bmatrix}$$

g. Use your answers from above to derive the **singular configurations** of the arm, if any, assuming $L_3 > L_2 > 0\,\text{m}$. **Sketch** the manipulator in each singular configuration that you find, and explain what effect the singularity has on the robot's motion in that configuration. *(12 points)*

$$J_v = \begin{bmatrix} 0 & -C_2(L_2 + L_3C_3) & L_3S_2S_3 \\ 0 & -S_2(L_2 + L_3C_3) & -L_3C_2S_3 \\ 1 & 0 & L_3C_3 \end{bmatrix}$$

$$\det(J_v) = +C_2(L_2 + L_3C_3)(L_3C_2S_3) + L_3S_2S_3(S_2(L_2 + L_3C_3))$$

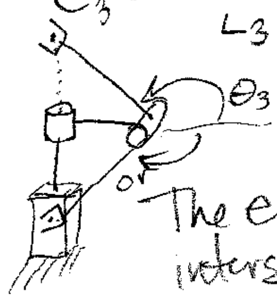
$$= (L_2 + L_3C_3)\left[L_3C_2^2S_3 + L_3S_2^2S_3\right]$$

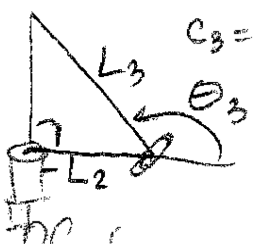
$$C_2^2 + S_2^2 = 1$$

$$= (L_2 + L_3C_3)(L_3)(S_3)$$

$$\boxed{L_2 + L_3C_3 = 0}$$

$$C_3 = -\frac{L_2}{L_3}$$



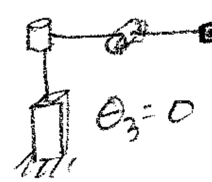


$$C_3 = -\frac{L_2}{L_3}$$

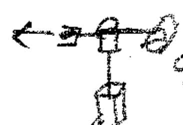
The end-effector intersects Z_1 so that $\dot{\theta}_2$ causes no motion

$$\boxed{\sin(\theta_3) = 0}$$

$\theta_3 = 0 + k\pi$ where $k = \ldots -2, -1, 0, 1, 2$



$\dot{d}_1$ and $\dot{\theta}_3$ move tip up. Cannot move in X_3 direction $\rightarrow$



Same problem $\theta_3 = \pi$