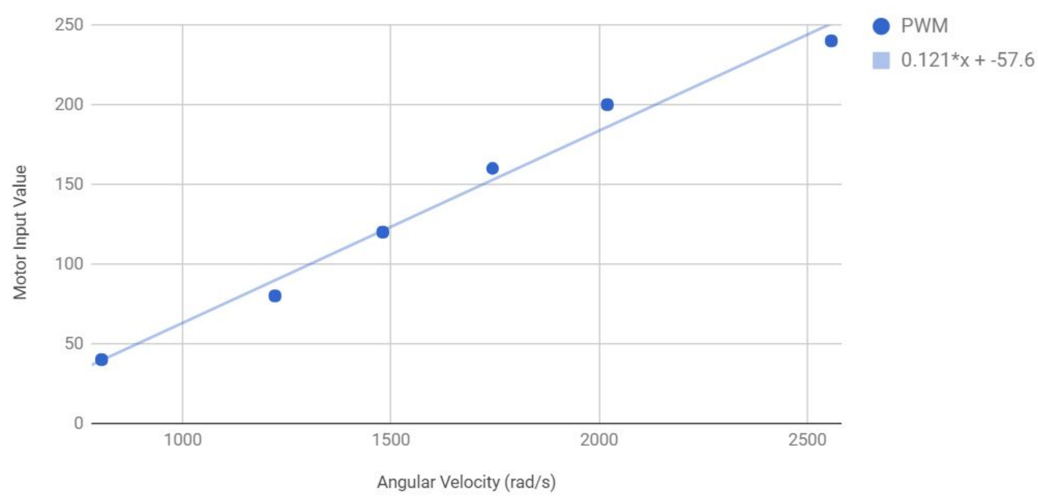
ME136 Lab 2

Group 6

Trey Fortmuller
Joey Kroeger
David Dominguez Hooper

1. PWM-speed map
    a) For generating the PWM speed map we removed the propellers from motors 2,3, and 4. We then attached a small piece of reflective tape to the propeller to allow us to gather RPM data. We taped the Crazeflie to the table to establish a firm base for testing.
    b) We used the map PWM values of 40, 80, 120, 160, 200, and 240. We conducted 3 tests for each value and captured the RPM data with a tachometer. We took the max value after collecting RPM data for 10 seconds at 30cm from the propeller. We used the average of the 3 values when generating our graph.
    c)



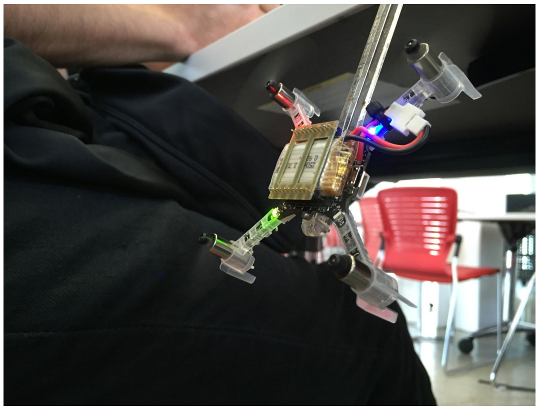
    d) We used a linear fit using an average of the data collected from our tachometer readings to relate the motor input value to angular velocity.

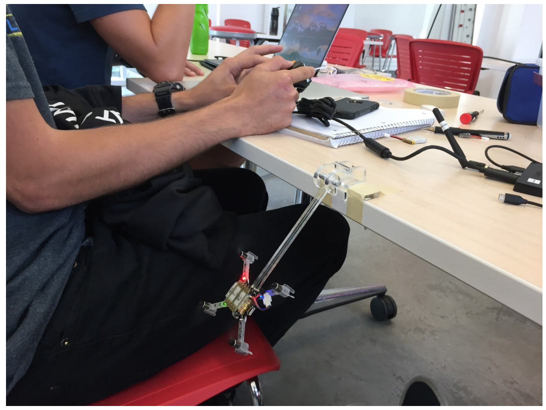
```
int pwmCommandFromSpeed(float desiredSpeed_rad_per_sec){
    // Replace these two coefficients with what you get
    // in the experiment. Note the trailing "f" after the
    // number -- this ensures that we use single precision
    // floating point (rather than double precision, which
    // would be substantially slower on the microcontroller).
    float a = -57.57631f; // the zeroth order term
    float b = 0.120601f; // the first order term

    return int(a + b * desiredSpeed_rad_per_sec);
}
```

    e) Our data shows that there is a good linear fit from desired speed in rad/sec to a motor input value. When we tested our function after we recorded our results, we found that we had an accurate map from angular velocity for mapped PWM values above 40.

2. Speed-Force Map

a) We attached the Crazeflie to a hinge and lever made from acrylic with a small aluminum dowel as a pivot. As the motors provide thrust, the quadcopter is lifted through some angle from the vertical and the torques about the hinge of the lever balance, allowing us to determine the force produced by each motor.





b) We commanded the quadcopter at four different motor speed values (1000, 1200, 1400, 1600 in radians per second), gently stabilized the quadcopter at a resting angle from the vertical and waited 5 seconds to obtain a log file including accelerometer data. Below is the dynamics model of the system we used to approximate the force per motor. Beta is the average angle from vertical value we approximated from the accelerometer data in the log file.

Tilt Angle from Accelerometer

$$\alpha_z = ||\vec{g}|| \sin(\beta)$$

$$\beta = \sin^{-1}\left(\frac{\alpha_z}{||\vec{g}||}\right)$$

Force from Tilt Angle

About the pivot point of the mount $\Sigma\tau = 0$. Let $||\vec{g}|| = g$

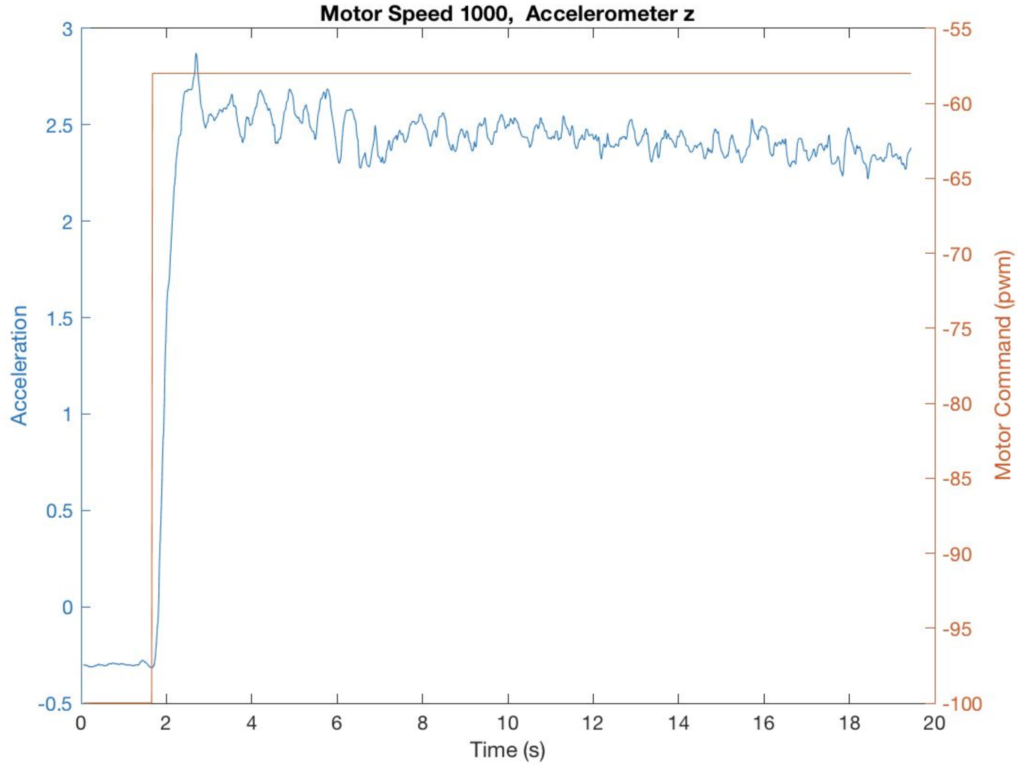
$$-m^R g l^R \sin(\beta) - m^B g l \sin(\beta) + 4 f_p l = 0$$

Solving for f_p we obtain

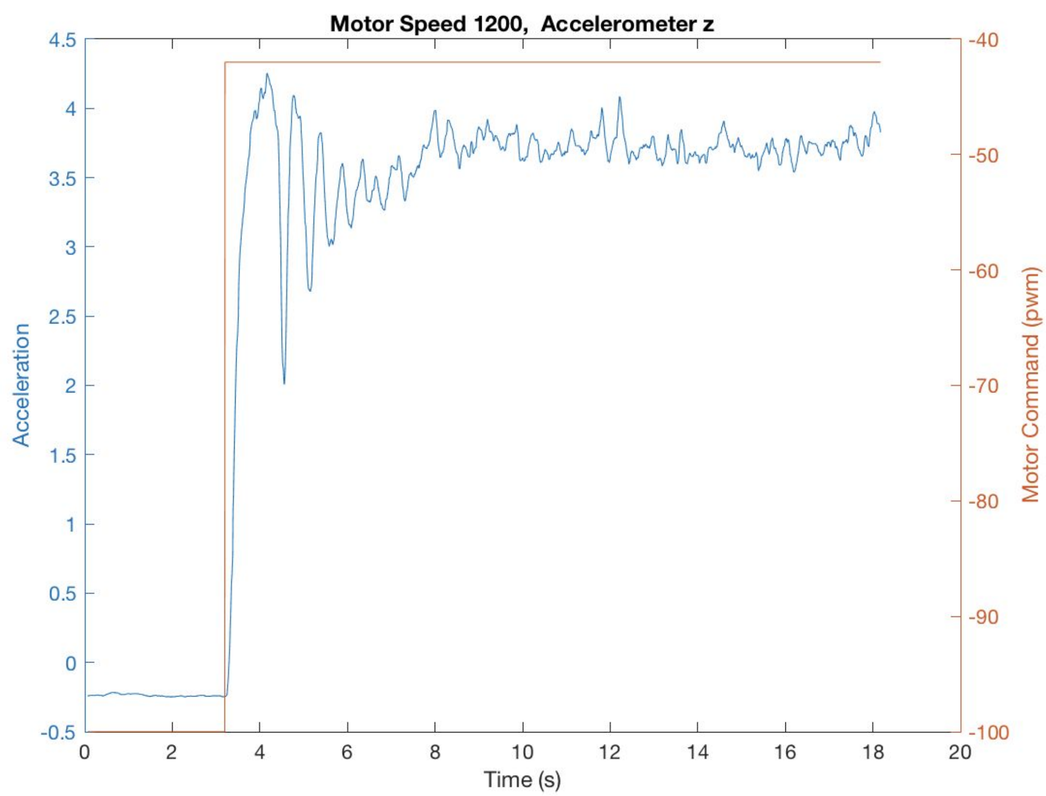
$$f_p = \frac{1}{4l}\left[m^R g l^R \sin(\beta) + m^B g l \sin(\beta)\right]$$

$$f_p = \frac{g\sin(\beta)}{4l}\left[m^R l^R + m^B l\right]$$

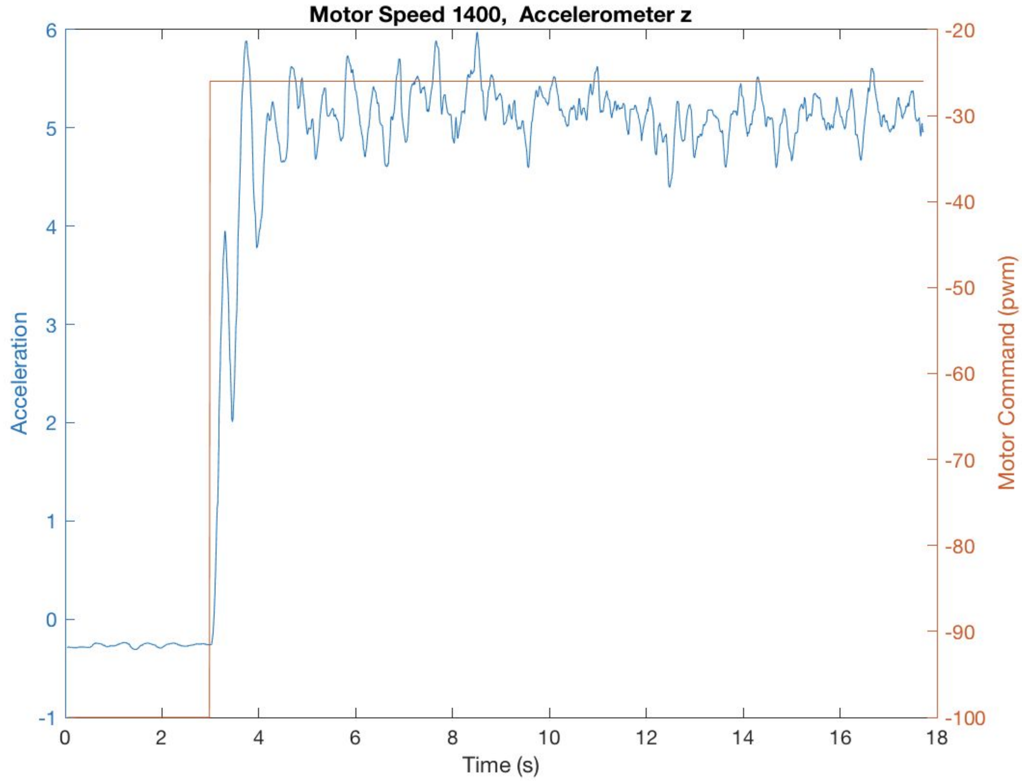
Below are the plots of accelerometer data overlaid with the motor command. We see the motor command change instantaneously and the accelerometer values spike quickly up to a new range of values which we averaged to obtain Beta.

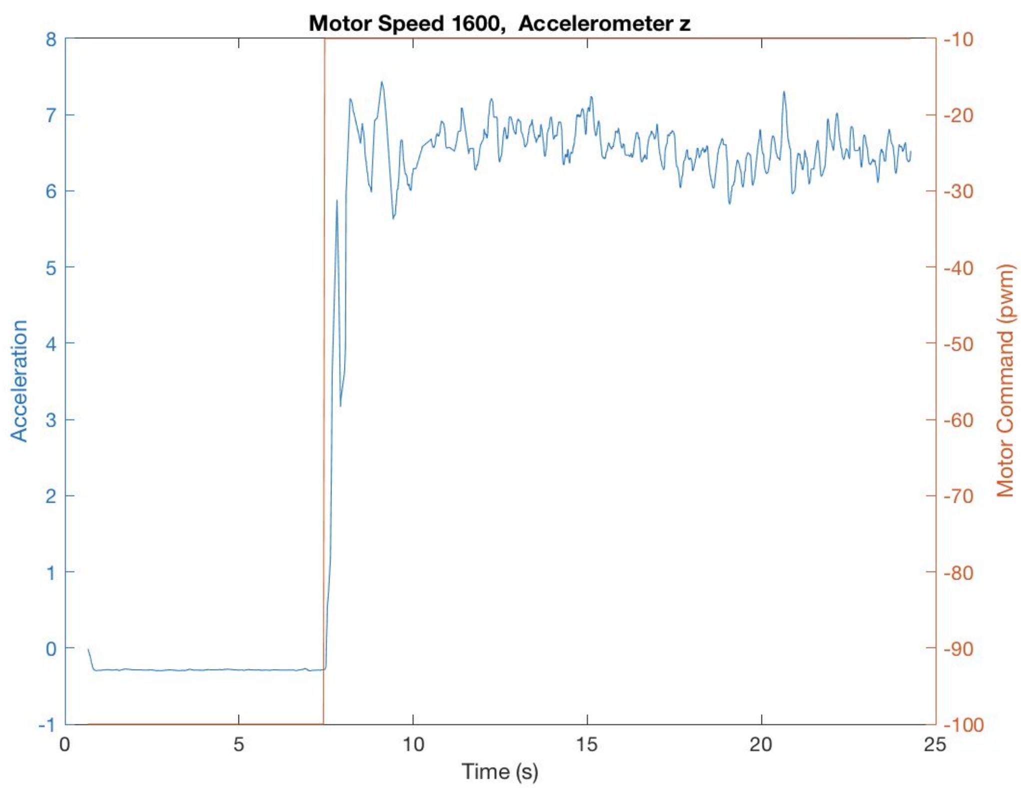


Motor Speed 1000, Accelerometer z
Acceleration
Motor Command (pwm)
Time (s)

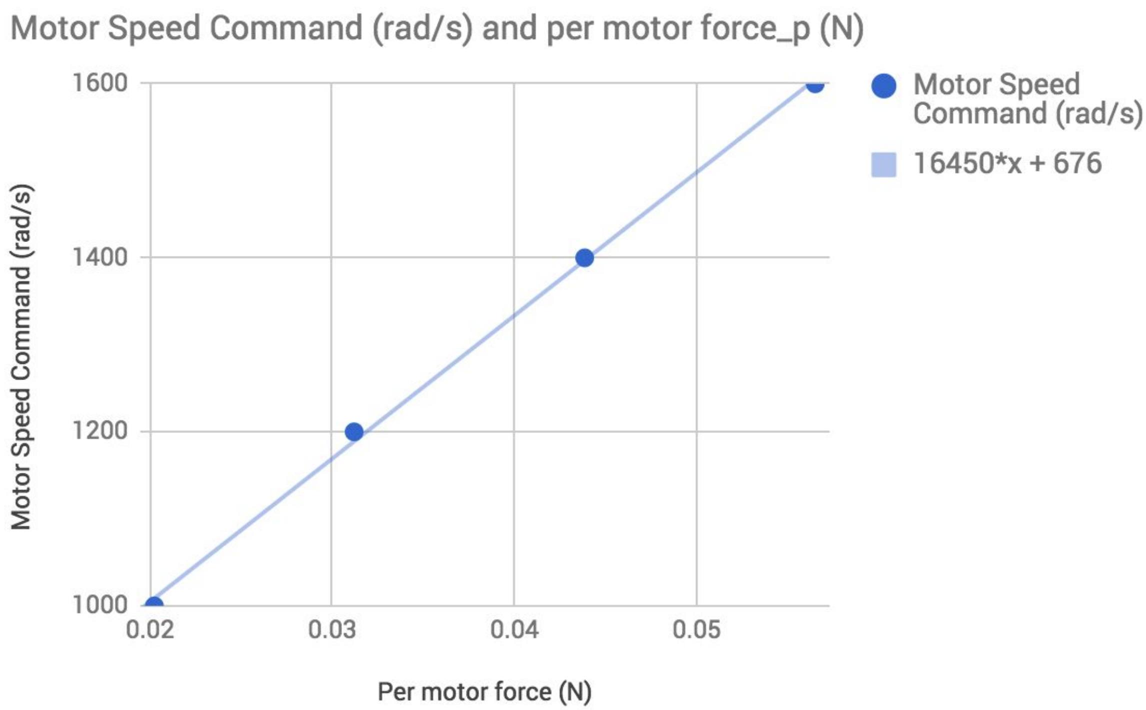


Motor Speed 1200, Accelerometer z
Acceleration
Motor Command (pwm)
Time (s)


Motor Speed 1400, Accelerometer z
Acceleration
Motor Command (pwm)
Time (s)
6
5
4
3
2
1
0
-1
-20
-30
-40
-50
-60
-70
-80
-90
-100
0
2
4
6
8
10
12
14
16
18


Motor Speed 1600, Accelerometer z
Acceleration
Motor Command (pwm)
Time (s)
8
7
6
5
4
3
2
1
0
-1
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
0
5
10
15
20
25

c) After obtaining an approximation of the force per motor, we plot motor speed commanded and the force per motor.



d) We used Google Sheets to obtain a least squares linear fit for our data which correlated closely with collected data.

e) There is a positive, linear relationship between motor speed command and per motor force. When we tested our mapping, we found that we were able to reproduce accurate results using our linear fit.

3.

```
  MainLoopOutput outVals;
//  motorCommand1 -> located at body +x +y
//  motorCommand2 -> located at body +x -y
//  motorCommand3 -> located at body -x -y
//  motorCommand4 -> located at body -x +y
  if (in.joystickInput.buttonGreen == 1 &&
in.joystickInput.buttonBlue == 1){
        outVals.motorCommand1 = 0;
        outVals.motorCommand2 = 240;
```

```
        outVals.motorCommand3 = 0;
        outVals.motorCommand4 = 0;
}
else if (in.joystickInput.buttonGreen == 1){
    outVals.motorCommand1 = 0;
    outVals.motorCommand2 = 40;
    outVals.motorCommand3 = 0;
    outVals.motorCommand4 = 0;
}
else if (in.joystickInput.buttonBlue == 1){
      outVals.motorCommand1 = 0;
      outVals.motorCommand2 = 80;
      outVals.motorCommand3 = 0;
      outVals.motorCommand4 = 0;
}
else if (in.joystickInput.buttonYellow == 1){
      outVals.motorCommand1 = 0;
      outVals.motorCommand2 = 120;
      outVals.motorCommand3 = 0;
      outVals.motorCommand4 = 0;
}
else if (in.joystickInput.buttonStart == 1){
      outVals.motorCommand1 = 0;
      outVals.motorCommand2 = 160;
      outVals.motorCommand3 = 0;
      outVals.motorCommand4 = 0;
}
else if (in.joystickInput.buttonSelect == 1){
      outVals.motorCommand1 = 0;
      outVals.motorCommand2 = 200;
      outVals.motorCommand3 = 0;
      outVals.motorCommand4 = 0;
}
else {
    outVals.motorCommand1 = 0;
    outVals.motorCommand2 = 0;
```

```
    outVals.motorCommand3 = 0;
    outVals.motorCommand4 = 0;
  }
  //copy the inputs and outputs:
  lastMainLoopInputs = in;
  lastMainLoopOutputs = outVals;
  return outVals;
}
```

```
% example plotting data using csv2struct() function

% assuming your csv file name is log_file_speed1000.csv with 28 columns:
name = 'accelerometer_tilt_logs-';
ext_pic = '.png';
speeds_n = [1000, 1200, 1400, 1600];

s_speed1000 = csv2struct('accelerometer_tilt_logs-1000.csv', 29);
s_speed1200 = csv2struct('accelerometer_tilt_logs-1200.csv', 29);
s_speed1400 = csv2struct('accelerometer_tilt_logs-1400.csv', 29);
s_speed1600 = csv2struct('accelerometer_tilt_logs-1600.csv', 29);

speeds = [s_speed1000, s_speed1200, s_speed1400, s_speed1600];

% This saves all data in the above struct 's_speed1000'.
% Now you can plot z-accelerometer output:
for i = 1:4
    figure;
    yyaxis left
    plot(speeds(i).time, speeds(i).accelerometer.z);
    yyaxis right
    plot(speeds(i).time, speeds(i).motorcmd1);

    yyaxis left
    title('Motor Speed ' + string(speeds_n(i)) + ',  Accelerometer z');
    xlabel('Time (s)');
    ylabel('Acceleration');

    yyaxis right;
    ylabel('Motor Command (pwm)');
    str = char(name + string(speeds_n(i)) + ext_pic);
    saveas(gcf, str);
end
```

4. Homework problems

ME136 PS2 Trey Fortmuller, Joey Kroeger, David D. Hooper

1) known

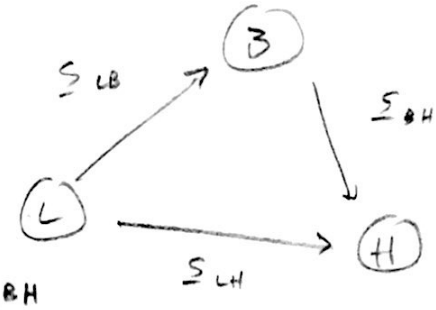
$[\underline{S}_{LB}]^B$, $[\underline{S}_{LH}]^E$, $[T]^{BE}$

find

$[\underline{S}_{LH}]^E$

we know $[\underline{v}]^B = [T]^{BA}[\underline{v}]^A$

and $\underline{S}_{AB} + \underline{S}_{BC} = \underline{S}_{AC}$



$$\underline{S}_{BL} + \underline{S}_{LH} = \underline{S}_{BH} \Rightarrow -\underline{S}_{LB} + \underline{S}_{LH} = \underline{S}_{BH}$$

coordinate in E

$$[-\underline{S}_{LB}]^E + [\underline{S}_{LH}]^E = [\underline{S}_{BH}]^E$$

$$[-\underline{S}_{LB}]^E = \underbrace{[T]^{EB}}_{\left([T]^{BE}\right)^{-1}}[-\underline{S}_{LB}]^B$$

so
$$\boxed{[\underline{S}_{BH}]^E = -\left([T]^{BE}\right)^{-1}[\underline{S}_{LB}]^B + [\underline{S}_{LH}]^E}$$

2) $\| [\underline{x}]^A \| = \sqrt{\overline{[\underline{x}]}^A [\underline{x}]^A} = \sqrt{(x_1^A)^2 + (x_2^A)^2 + (x_3)^2}$

$[\underline{x}]^B = [T]^{BA} [\underline{x}]^A \Rightarrow \overline{(AB)} = \bar{B}\bar{A}$

$\| [\underline{x}]^B \| = \sqrt{\overline{[\underline{x}]}^B [\underline{x}]^B} = \sqrt{\overline{[\underline{x}]}^A \overline{[T]}^{BA} [T]^{BA} [\underline{x}]^A}$

using the orthogonality property of $[T]^{xy}$ $\longrightarrow$ $= \sqrt{\overline{[\underline{x}]}^A \left([T]^{BA}\right)^{-1} \cancel{[T]^{BA}} [\underline{x}]^A}$

$\| [\underline{x}]^B \| = \sqrt{\overline{[\underline{x}]}^A [\underline{x}]^A} = \| [\underline{x}]^A \|$

so $\| [\underline{x}]^B \| = \| [\underline{x}]^A \|$

the magnitude of the 1OT does not change as a function of coordinates used to express the tensor.

3) $[\alpha]^B = [T]^{BI} [\alpha]^I$

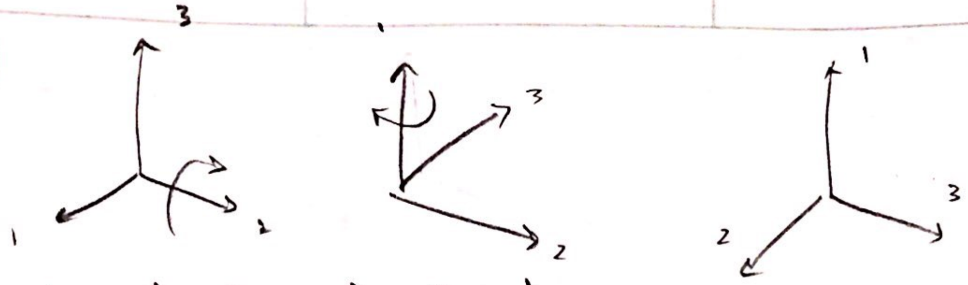
there is a $-45°$ rotation about the z-axis from B to I.

$$= \begin{bmatrix} \cos(-45°) & -\sin(45°) & 0 \\ \sin(-45°) & \cos(-45°) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix} = \boxed{\begin{bmatrix} 5\sqrt{2}/2 \\ -5\sqrt{2}/2 \\ 10 \end{bmatrix}}$$

Scanned by CamScanner

4)



$$R^y(-90)R^x(-90)R^z(\theta)$$

$$= \begin{bmatrix} \cos(-90) & 0 & \sin(-90) \\ 0 & 1 & 0 \\ -\sin(-90) & 0 & \cos(-90) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-90) & -\sin(-90) \\ 0 & \sin(-90) & \cos(-90) \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

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

$$[T]^{xy} = \begin{bmatrix} \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \cos\theta & -\sin\theta & 0 \end{bmatrix}$$

Scanned by CamScanner