[Robotics]

(Project\_1)

*Sub.To:*

*DR/ Muhammed Rushdy*

*E/ Eslam Mahmoud*

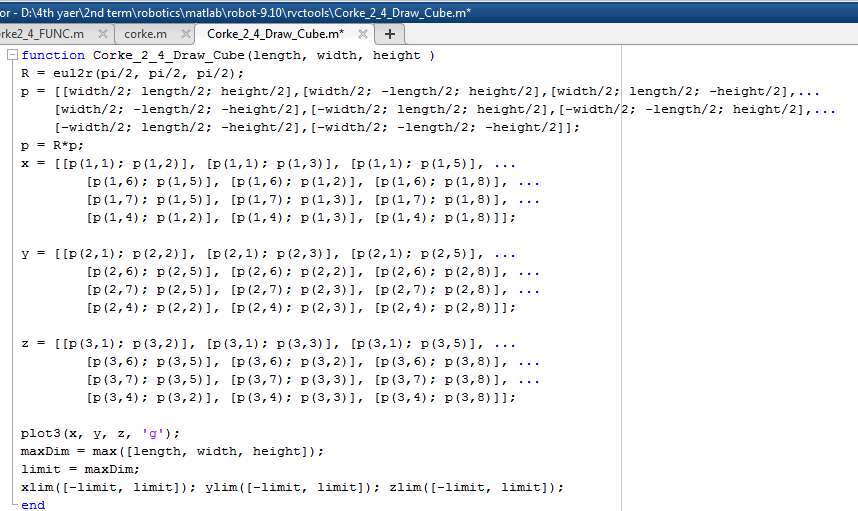
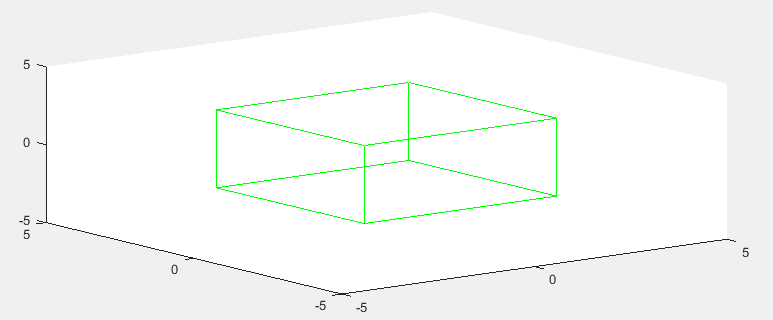
*Mostafa Hisham*

*Sec :2*

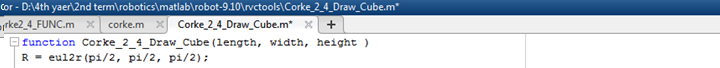
*BN:27*

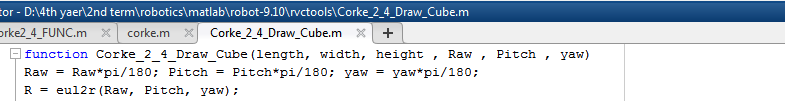
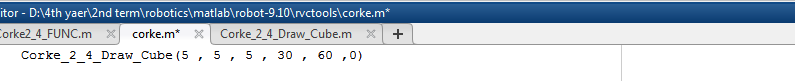
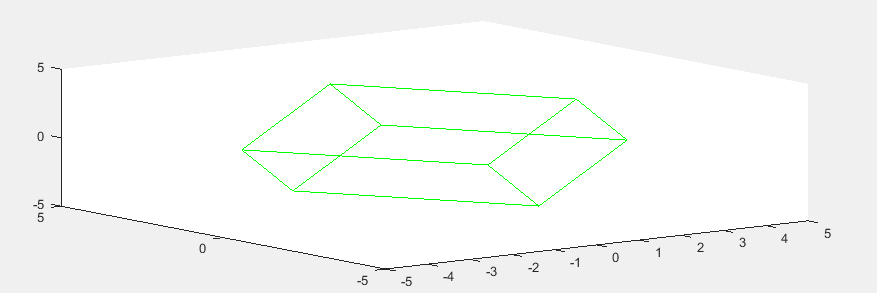
**Project 01 - Representing Position and Orientation**

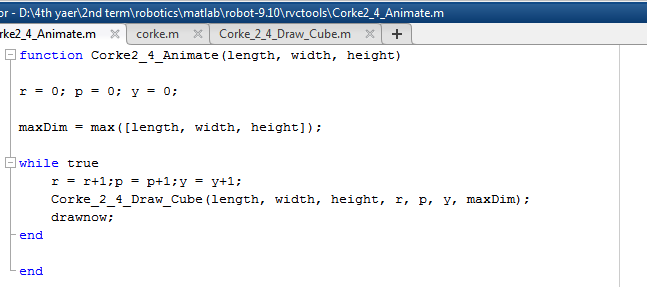
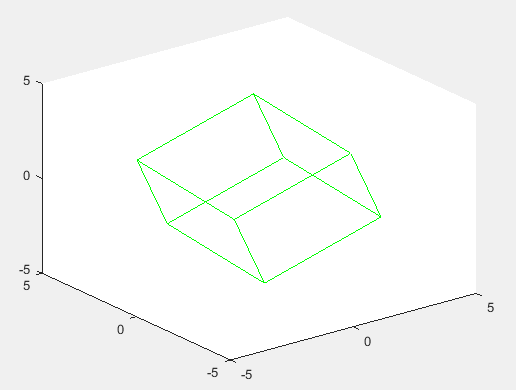
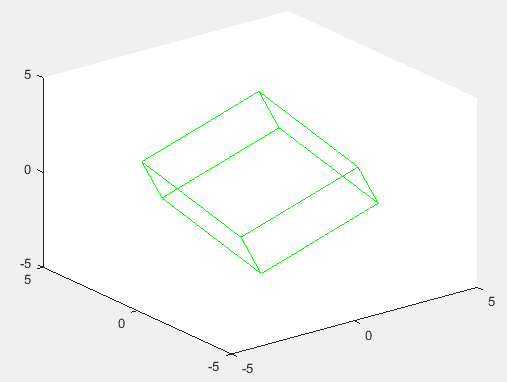
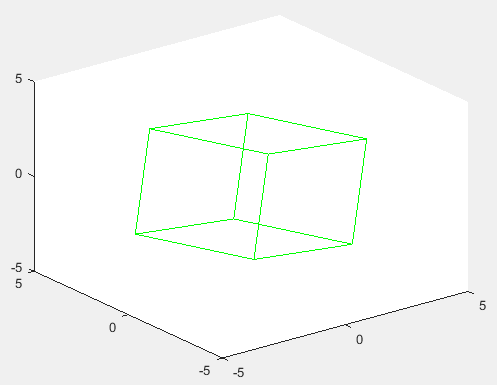
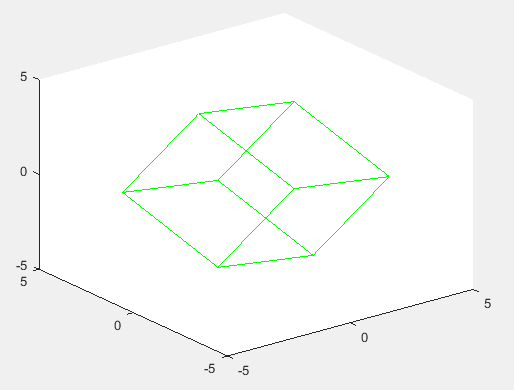
**A)From [Corke 2011], solve the following exercises:**

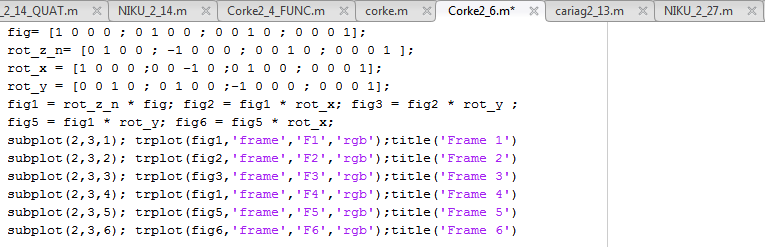
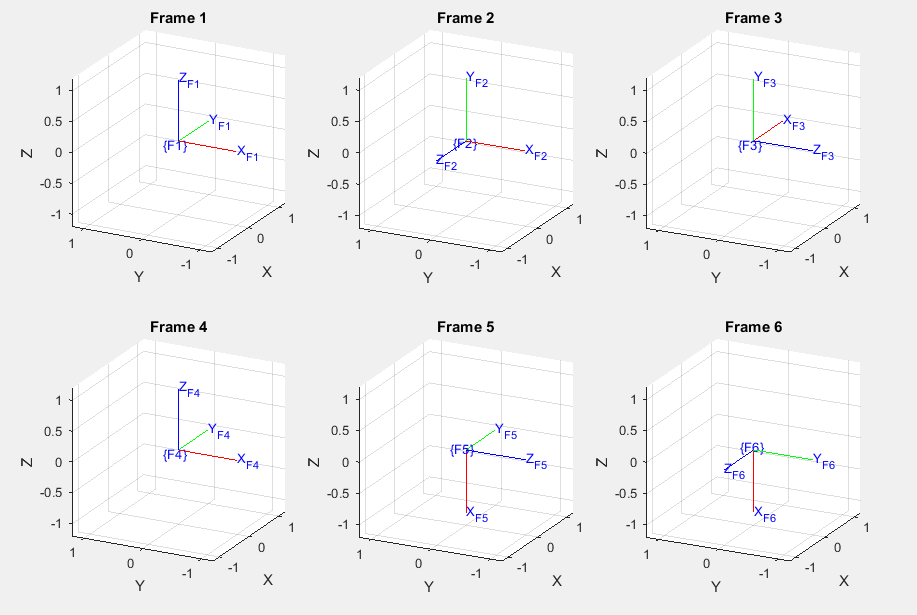
****

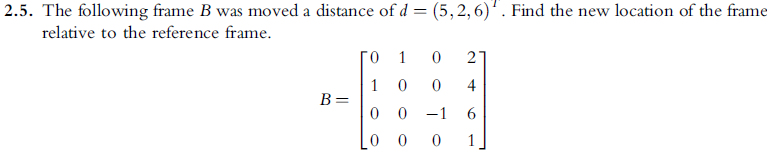
****

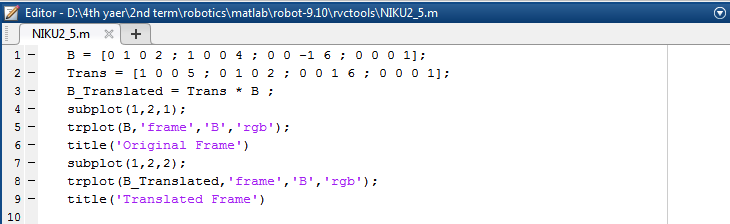
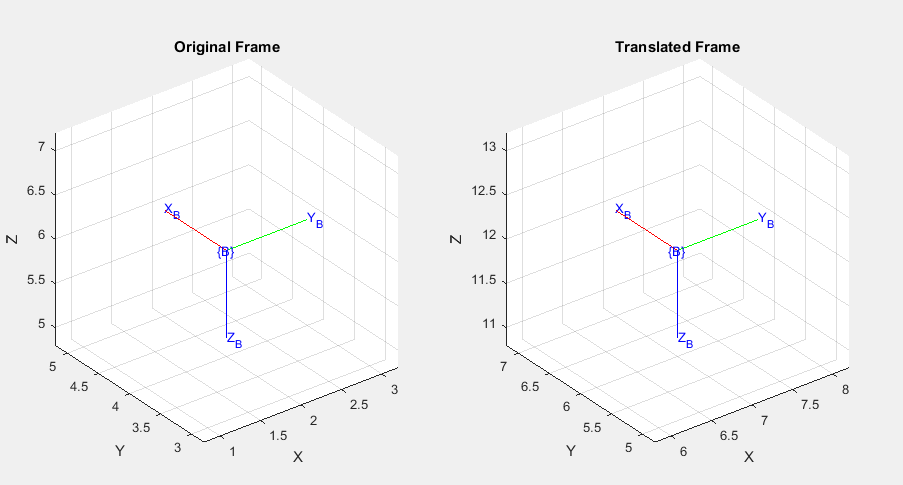
**a)**

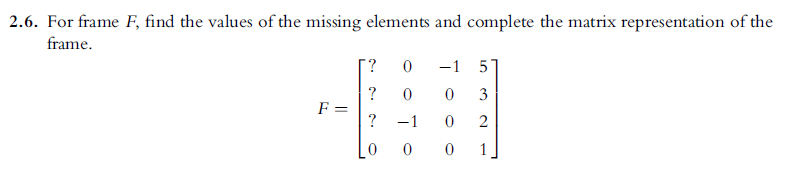
**The Modification**

****

****

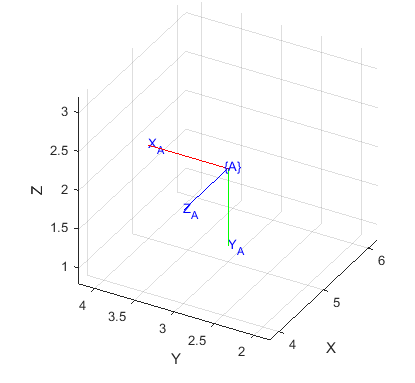
**B) From [Niku 2010], solve the following problems:**

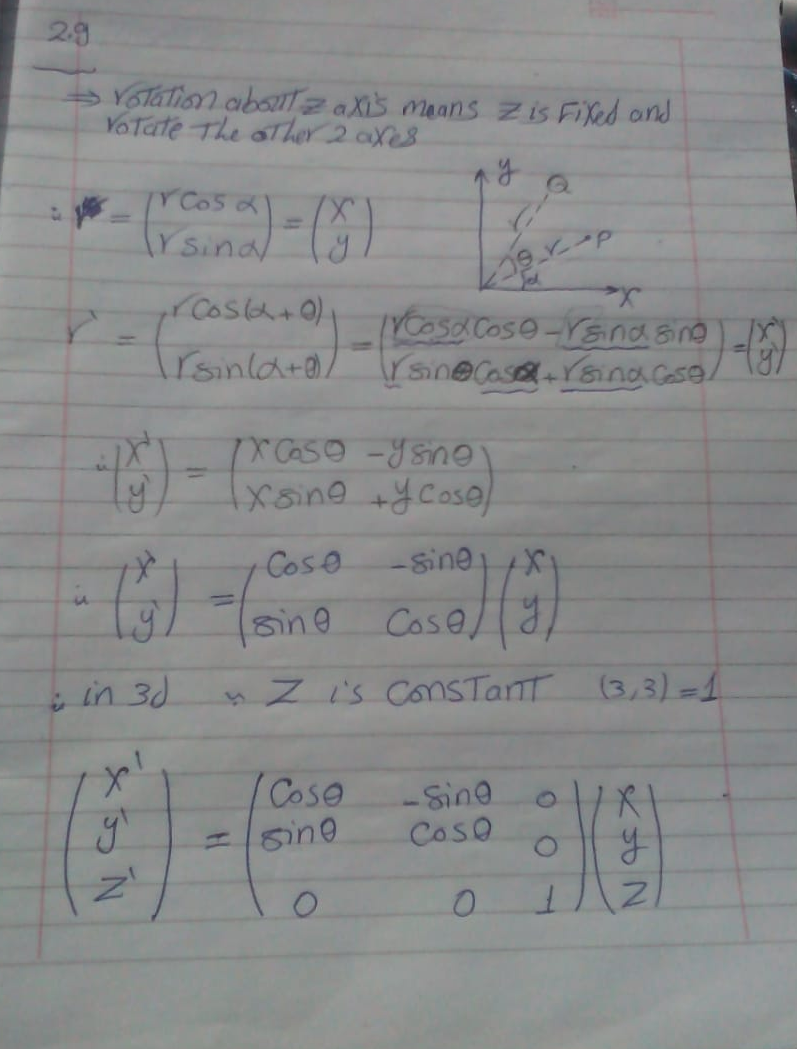
ANS

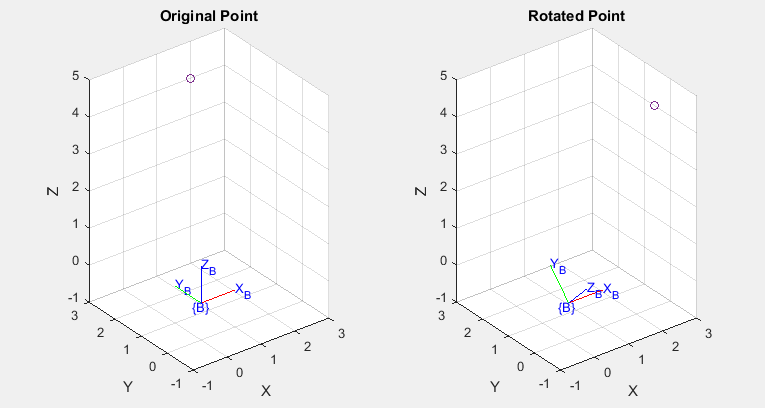
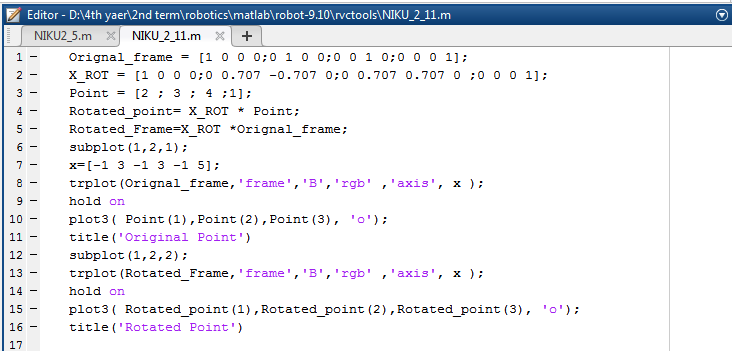
**ANS**

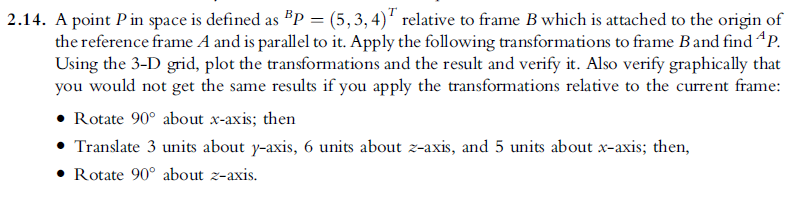
* Using cross product to get n 🡪 O \* A = N

Therefore,





**ANS**

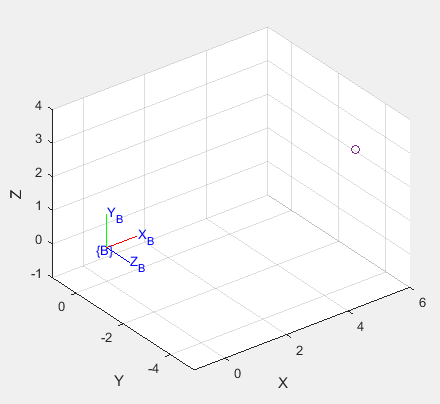
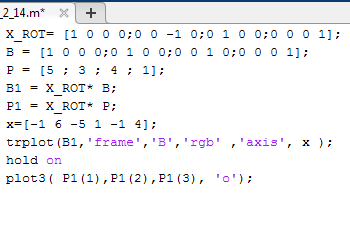
ANS

**A\_B = ROT (Z= 90) Trans (X=5, Y =3, Z=6) \* ROT (X=90) \* B**

**First Step (Rot (x = 90))**

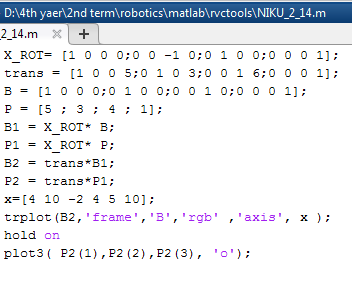
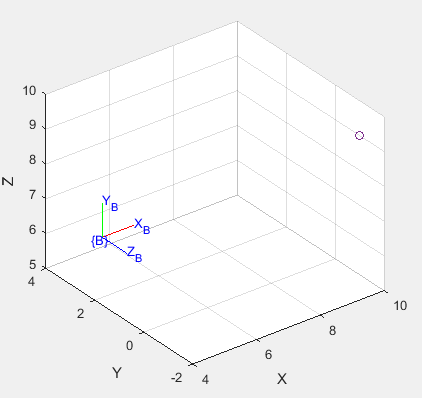
***For the Frame:***

***For the point:***



**Second Step (Trans (X=5, Y =3, Z=6))**

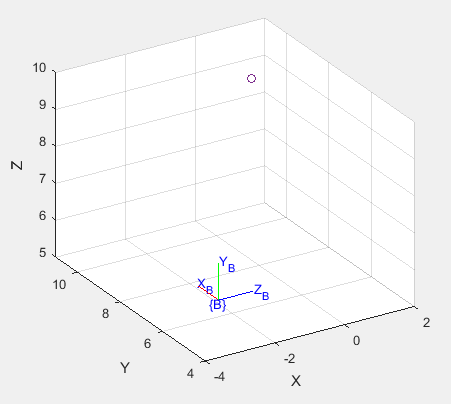
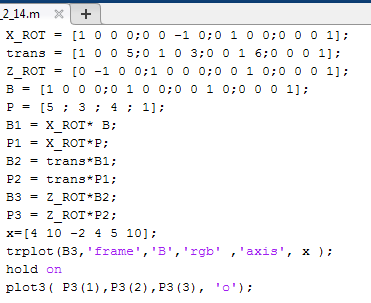
***For the frame:***

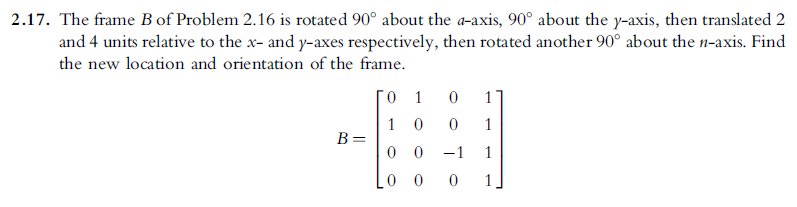
***For the point:***

**Third Step (ROT (Z= 90))**

***For the frame:***

***For the point:***

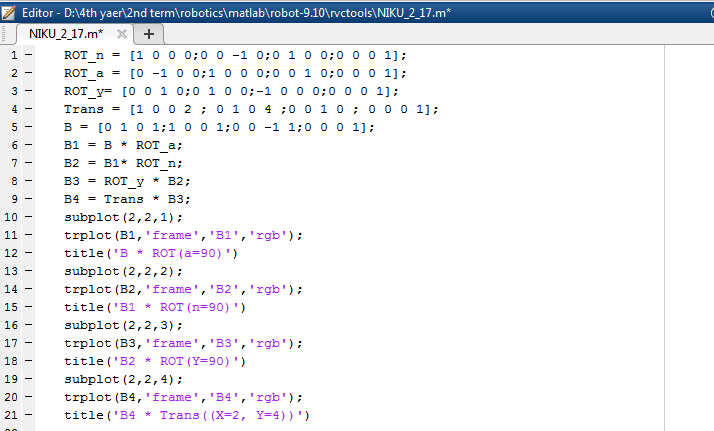
******

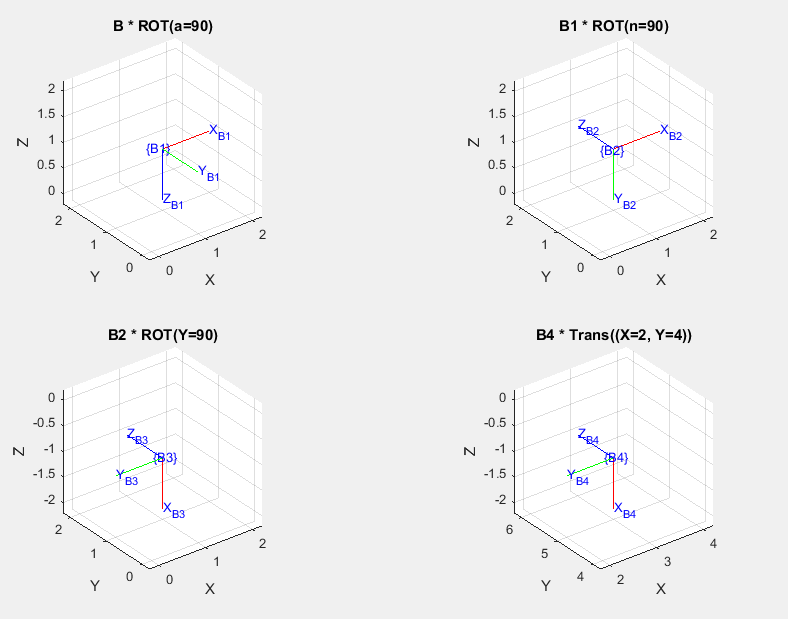
**ANS**

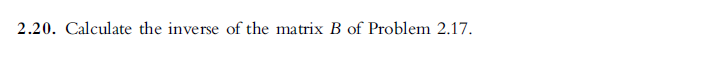
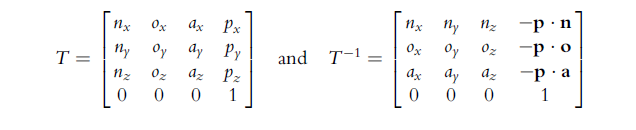
We have 4 transformations

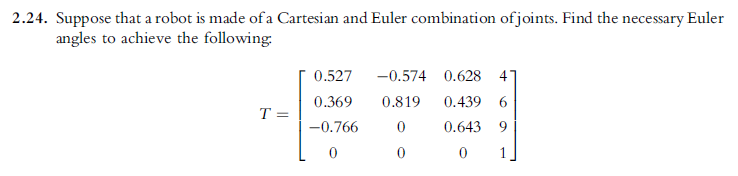
* 2 for the rotated frame which are post calculated.
* 2 for the reference frame.

Therefore, the seq is

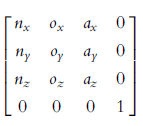
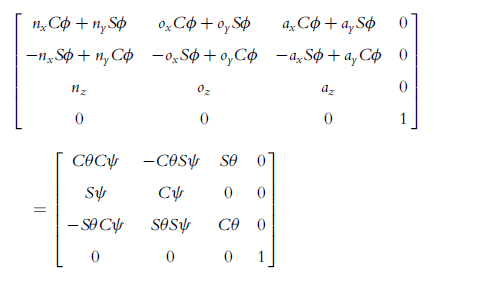
**B\_new = Trans (x=2, y =4) \* ROT (y=90) \* B \* ROT (a= 90) \* ROT (n=90)**

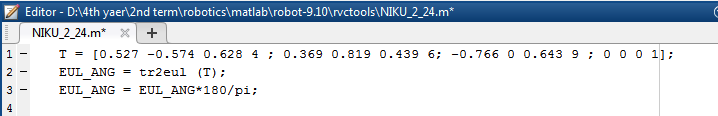
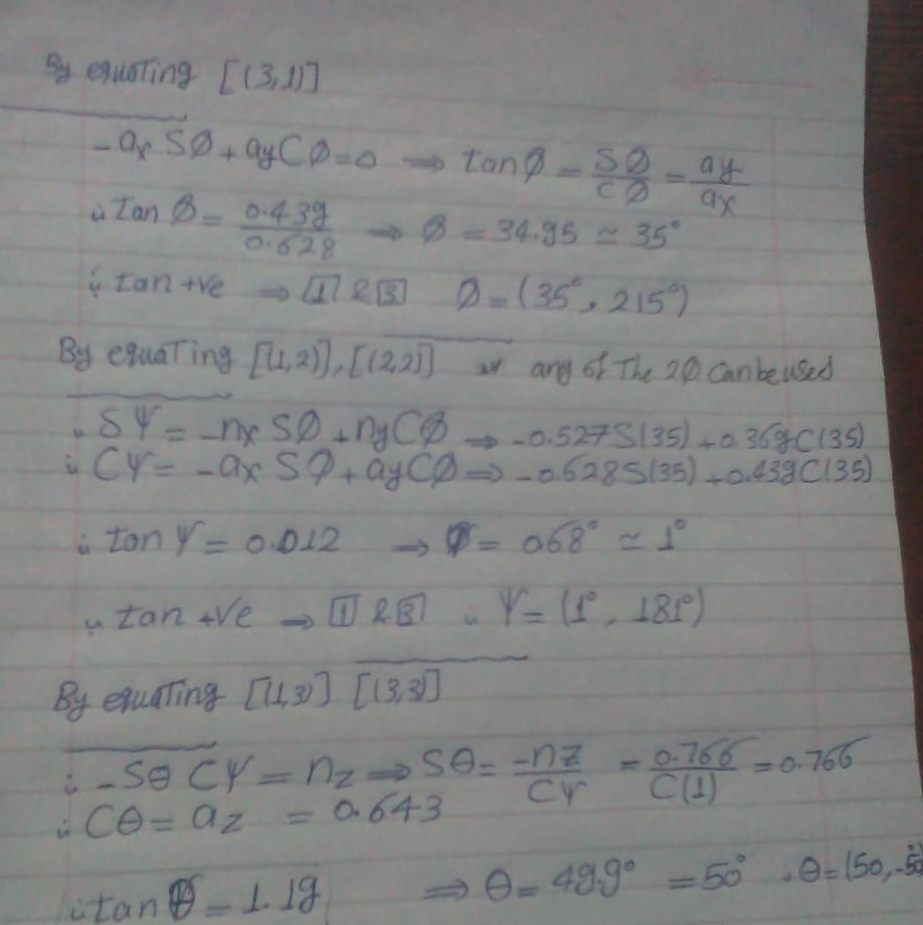


****

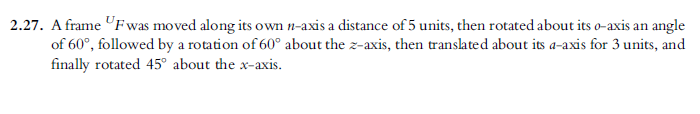
****

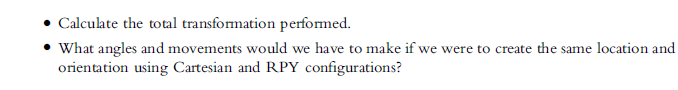
|  |  |
| --- | --- |
| SIN  180- atan(ang) | ALL  = atan(ang) |
| TAN  180+atan(ang) | COS  360+atan(ang) |

****



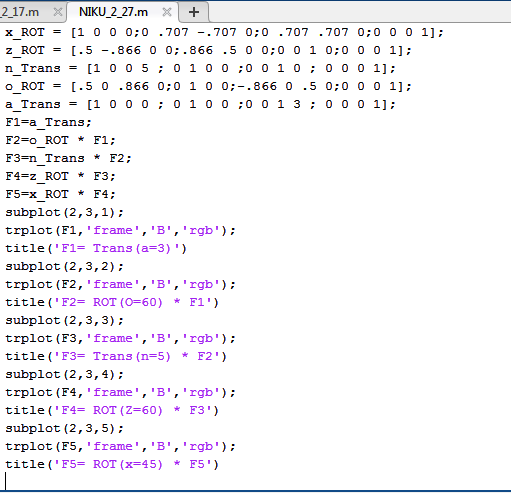
**Φ = [35 , 215] Θ = [50 , -50] ψ = [1 , 181 ]**

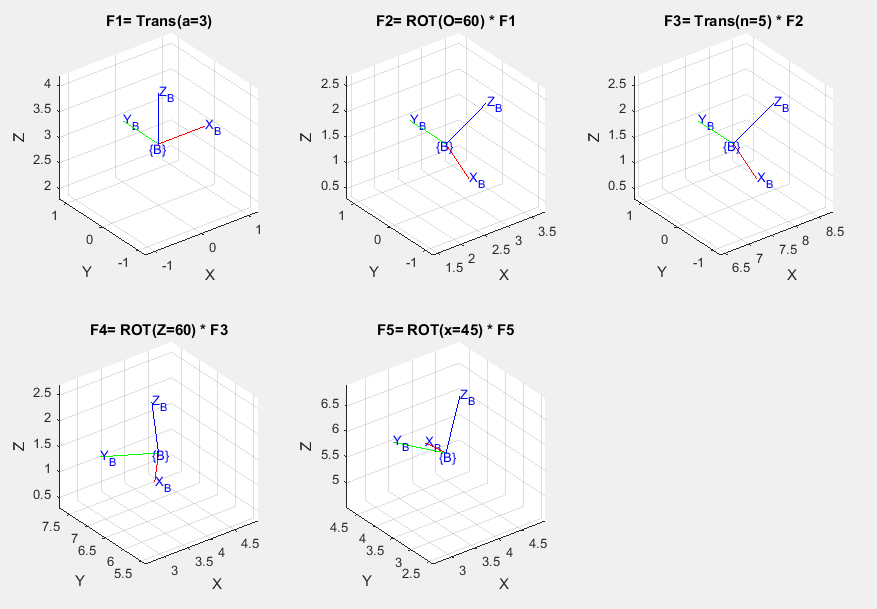




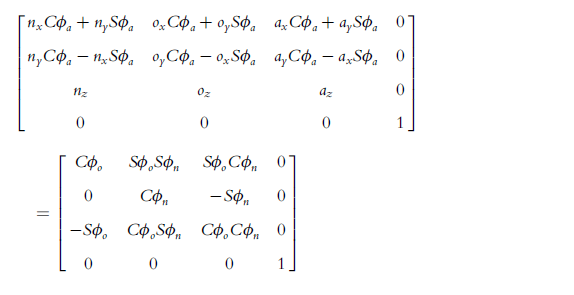
**The sequence is**

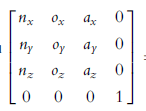
**F (new)=ROT(X=45)\*ROT(Z=60)\*Trans(n=5, O=0, a=0)\*ROT(O= 60)\*Trans(n=0, O=0, a=3)**

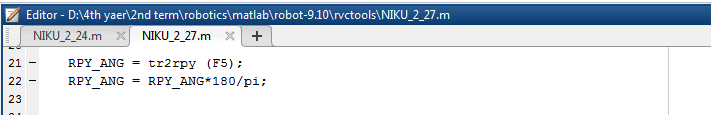
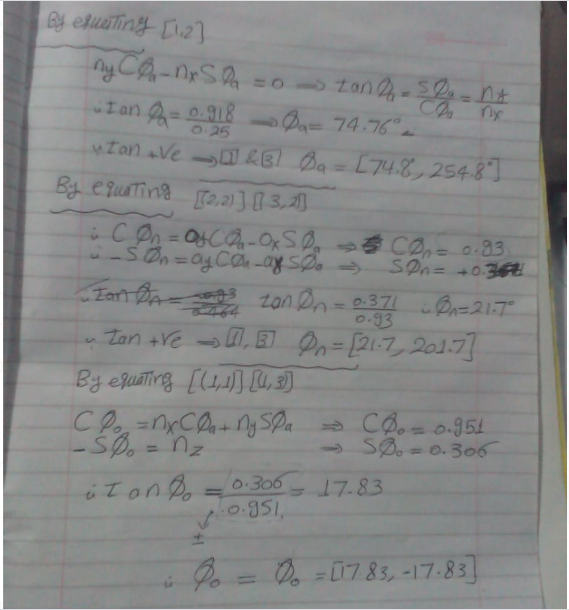




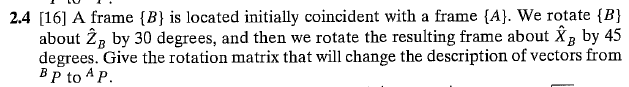
B) For **Cartesian** the angles and movements as the F5, point (3.799, 3.592, 5.713)

C) For RPY

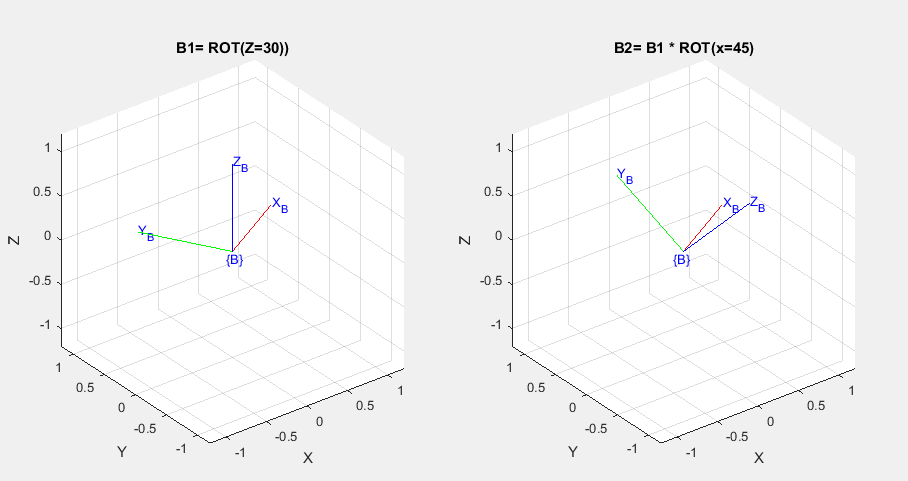
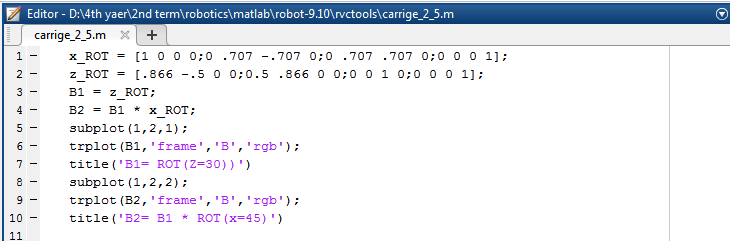




**Φa= [74.8 , 254.8] Φn= [21.7 , 201.7] Φo= [17.83 , -17.83]**

**3) From [Craig 2005], solve the following exercises:**

**ANS**

**Note : We rotate around the object B frame**

****

**ANS**

As in Angle axis representation we have only one axis of rotation and one angle.

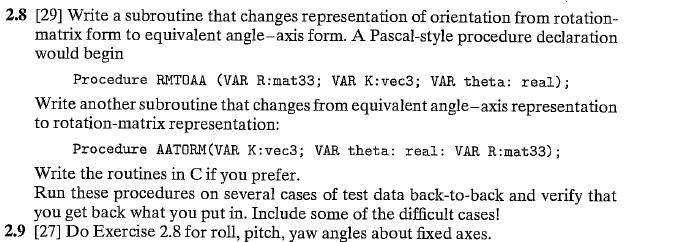
If we have Rotation Matrix **R** and we want to get the vector of the Angle axis representation **V,**

From the definition of eigenvalues and eigenvectors **R\*V = λ\*V**

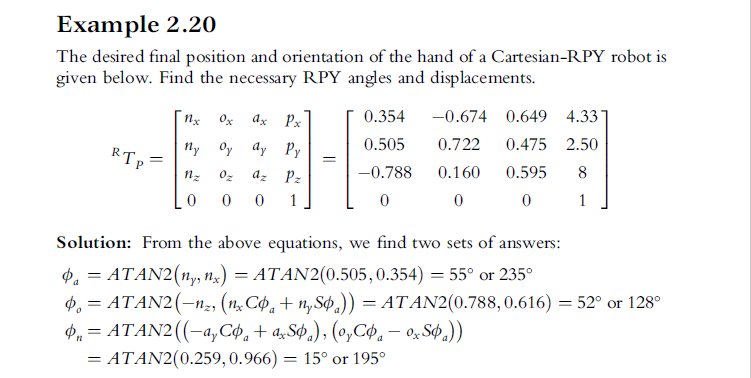
where v is the eigenvector corresponding to λ. For the case λ =1 then

**R\*V = V**

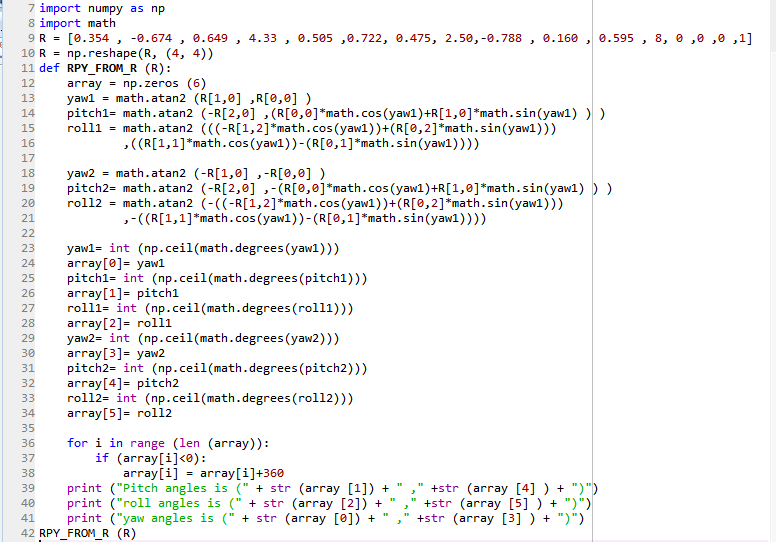
**So, when the eigenvalue is =1 the corresponding eigenvector is the Angle axis representation vector, the rotation vector.**

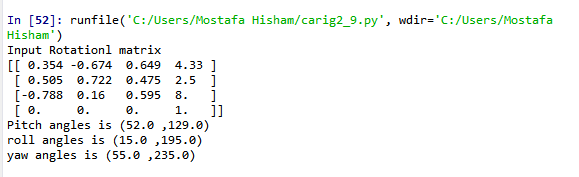
****

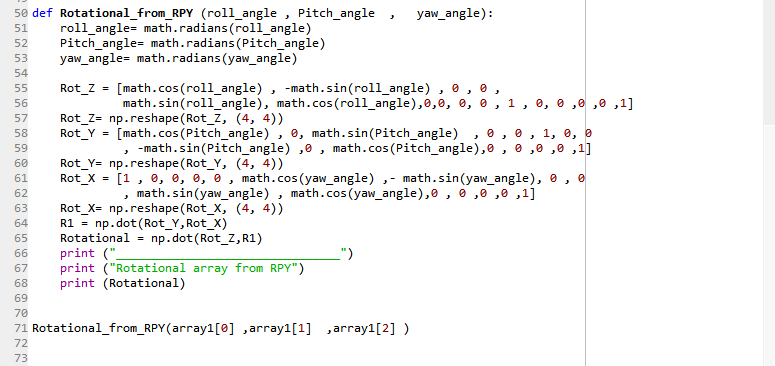
**ANS**

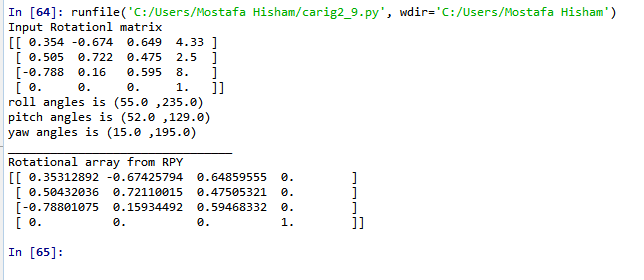
****The script tested by the example (2.20) in **[Niku 2010]**

**Python Script to convert Rotational to RPY**

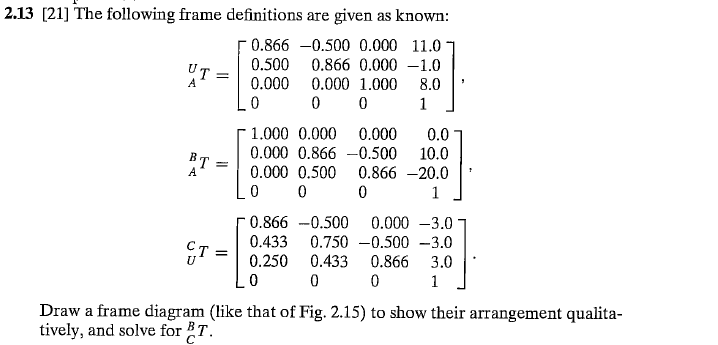
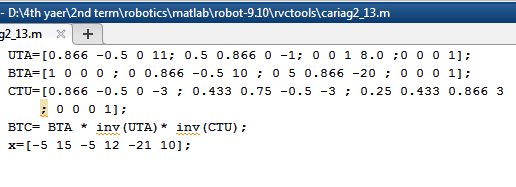
****

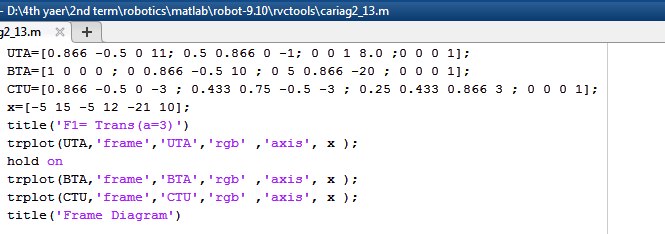
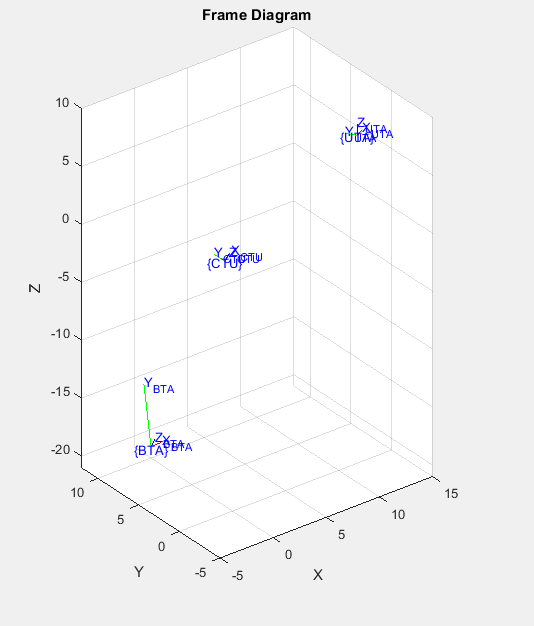
**OUTPUT**

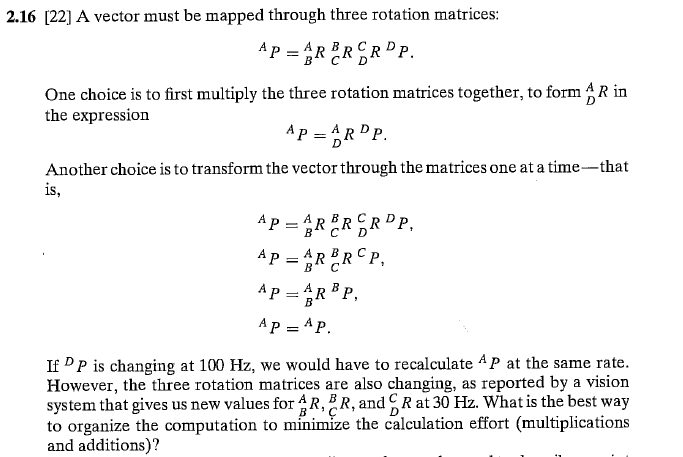
**Script to convert from RPY to Rotational**

**OUTPUT**

**I used the output of the first function as input to the second function and the result was the input for the first function except the point because it wasn’t required to put it in the array.**

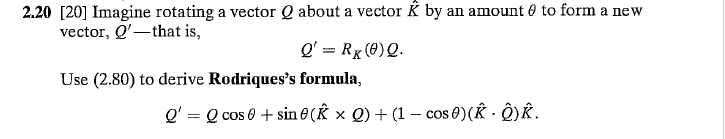
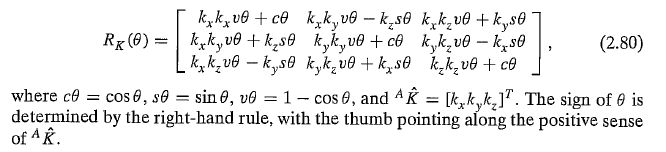
****

**Frame diagram Script**

****

**ANS**

|  |  |
| --- | --- |
| **First Method** | **Second Method** |
| The three rotations will mutiply with each other and then multiply with the point | the result of each muliplication is a point and will multipy with the rotation |
| * Two rotaion matrices multiplication will result in 27 product and 18 addtion, and will happen 2 times.   27 \* 2 \* 30 = 1620 product  18 \* 2 \* 30 = 1080 addtion   * The result rotation matrix will multpy by the point , 9 product and 6 addition   9 \* 100 = 900  6 \* 100 = 600   * The total   1620 + 900 = 2520 product  1080 \* 600 = 1680 addition | * The rotation matrix will multpy by the point , 9 product and 6 addition ,and will happed 3 times * The total is 3\*7= 27 product   3\*6= 18 addition   * The total   27 \* 100 = 2700 product  18 \* 100 = 1800 addition |
| **Therefore, the first method is more effictive** | |

**ANS**

Rodrigues’s formula =

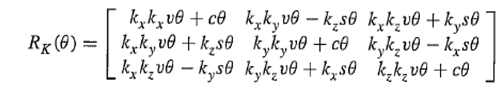


Assume:



🡪 (1)

**Multiply Rk () \* Q**

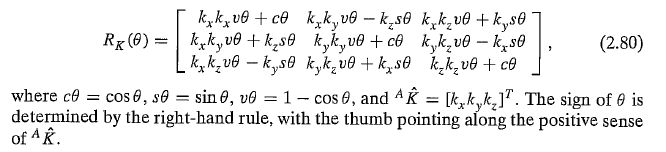
****

**If we rearrange this matrix we will get**

-> (2)

**From (1) And (2)**

**The Formula is Proved**

**ANS**

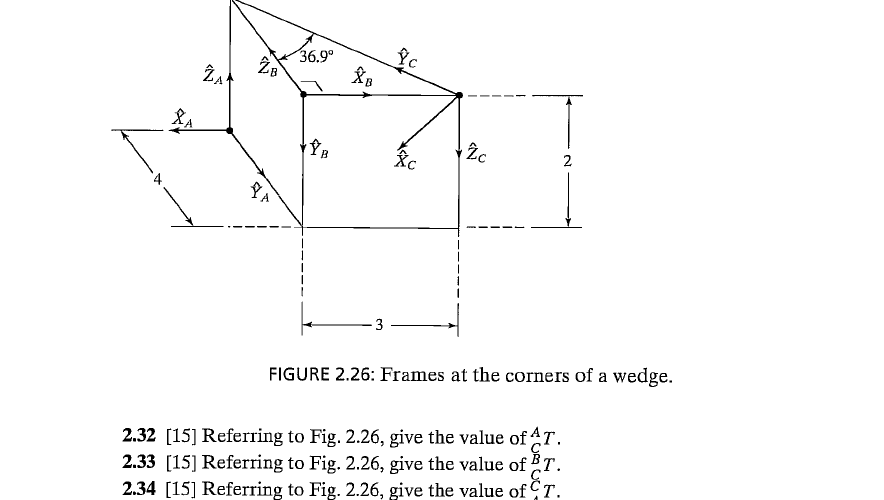
* ****We have C() =1, therefore *v*
* Assume that R1 (Φ) where Φ >>>>1, and R2 (Θ), Θ >>>>1

where Φ >>>>1, and R2 (Θ), Θ >>>>1

therefore, Φ \* Θ = 0

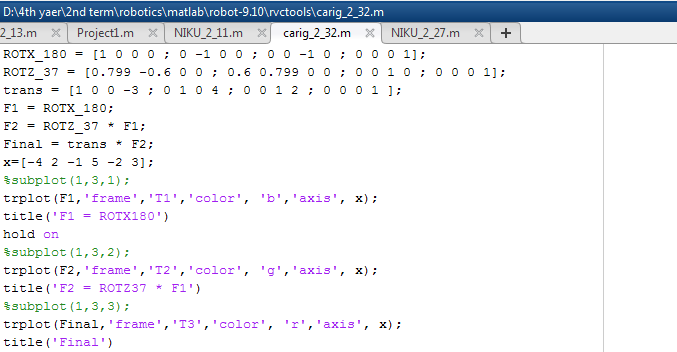
**We have symmetric matrix**

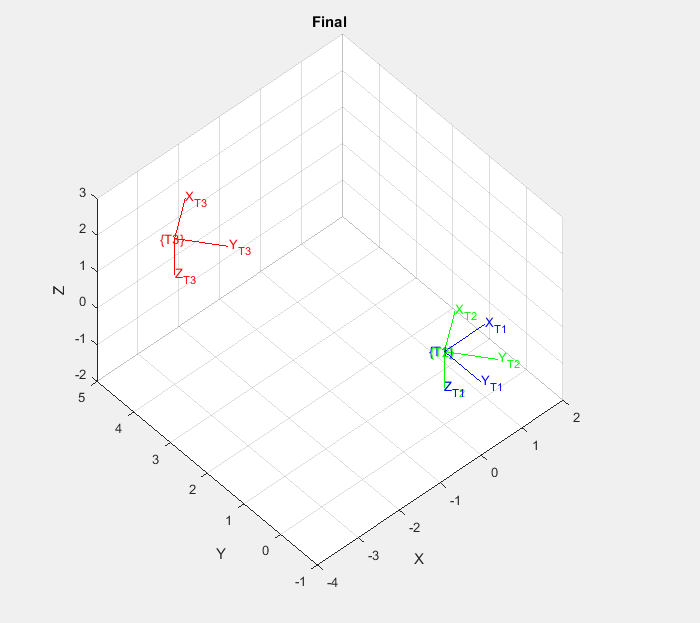
**So**

****

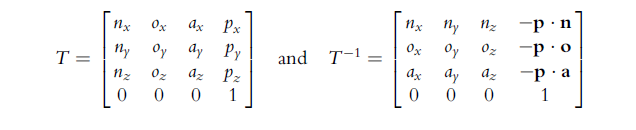
**ANS**

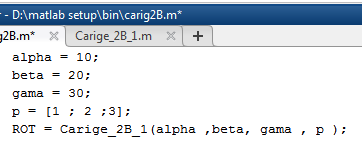
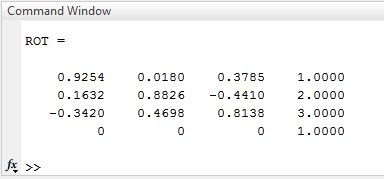
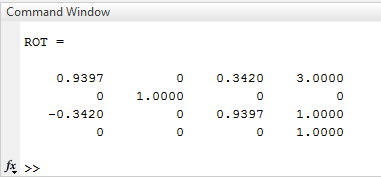
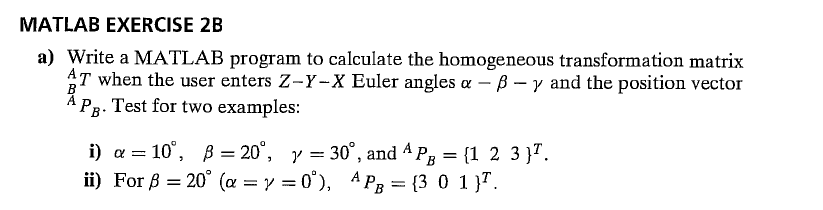
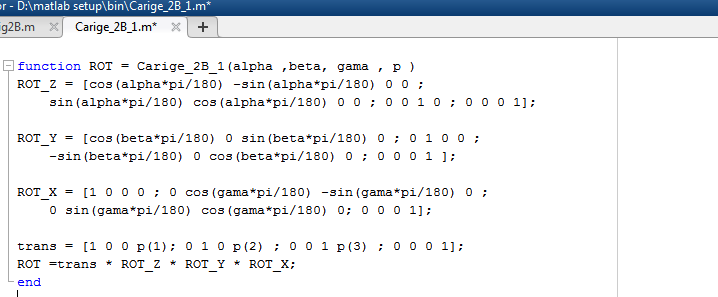
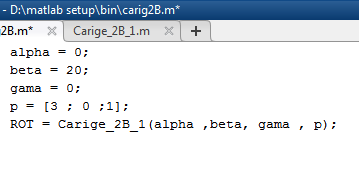
**((((((Assume A is the Reference Frame))))))**

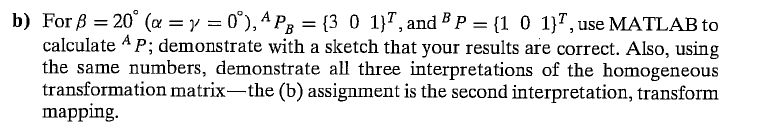
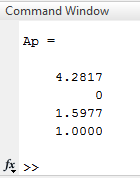
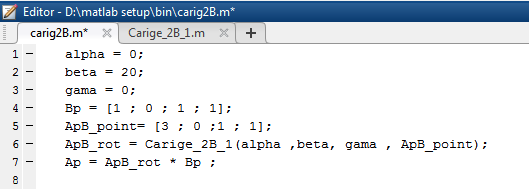
 **The sequence is: Trans (x=-3, y =4, z=2) \* ROT (z= 36.9) \* ROT (x=180) = T(new)**



**2.43**

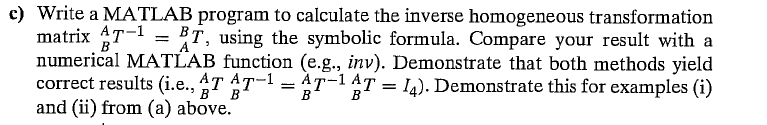
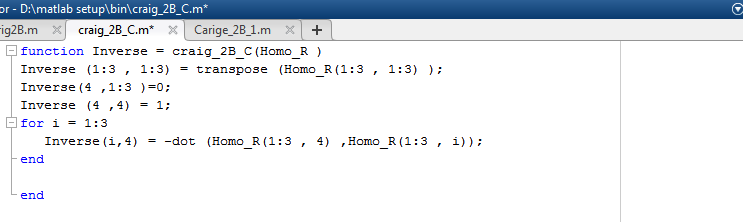
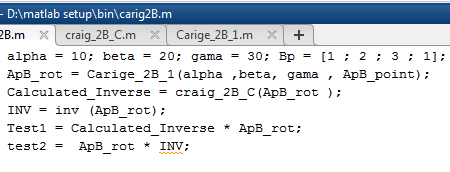
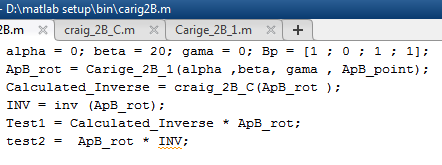
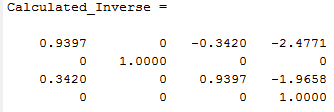
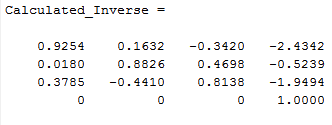
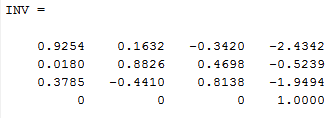
**CTA = INV (ATC) , and we calculate ATC in 2.32.**

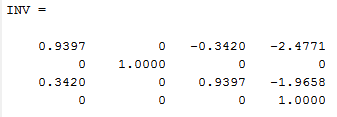
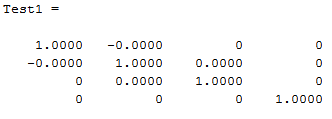
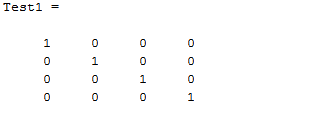
First Input (i) Second input(ii)

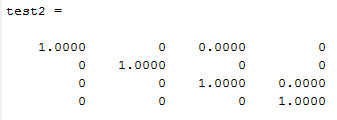
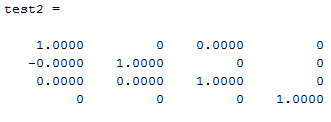


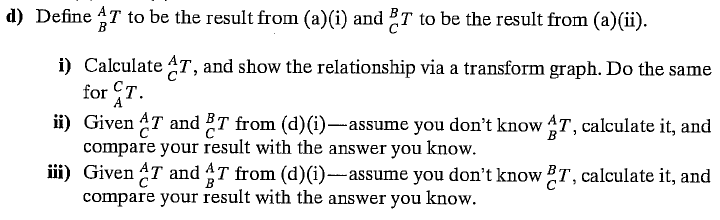
BP

APB

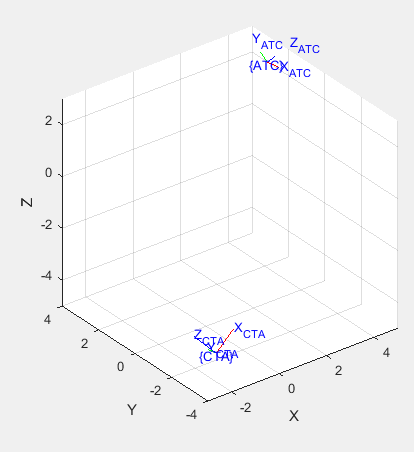
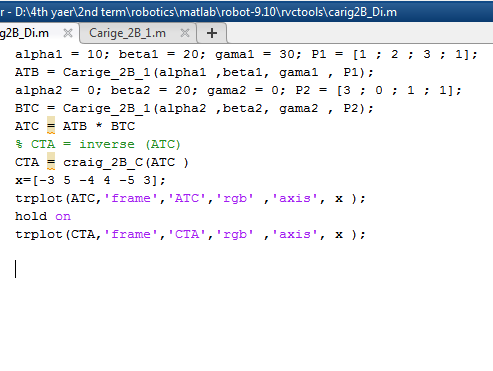
 Second Input (ii) First Input (i)

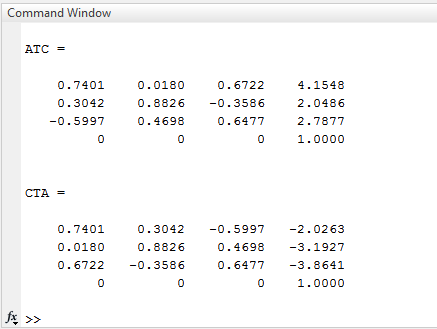




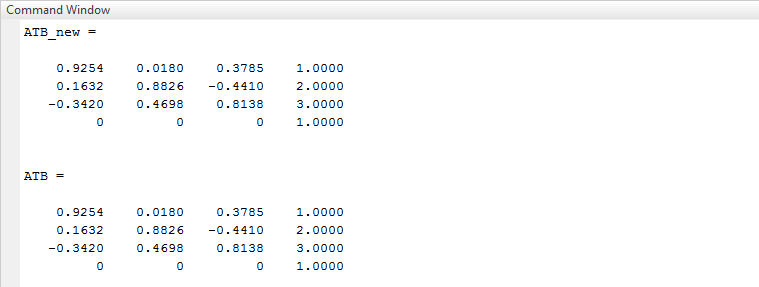
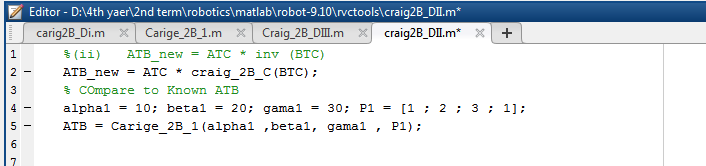


1. **ATC = ATB \* BTC and CTA = inv (ATC)**

****

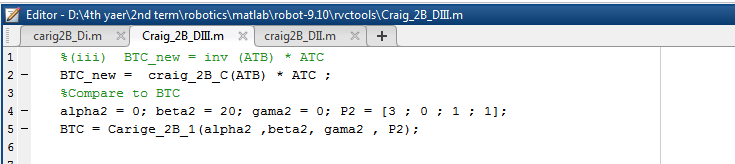
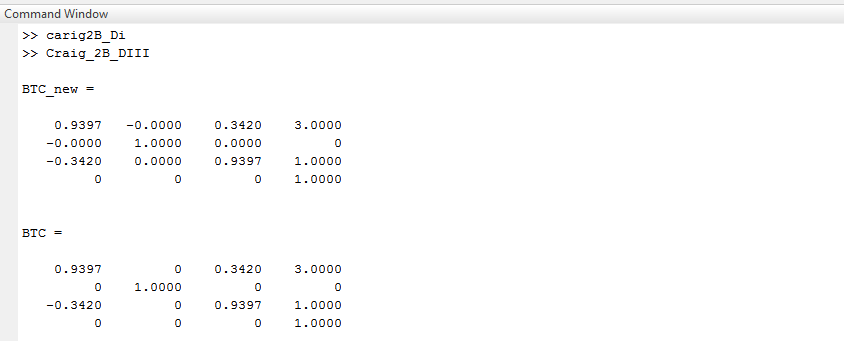


1. **ATB = ATC \* inv (BTC)**

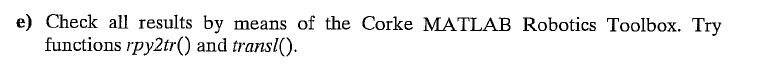
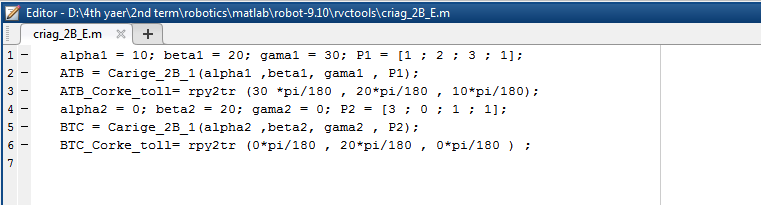
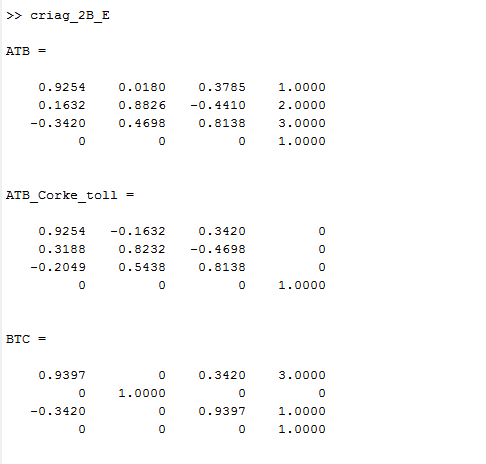
****Note : ATC and BTC are stored variables from D(i)

**The results are equal.**

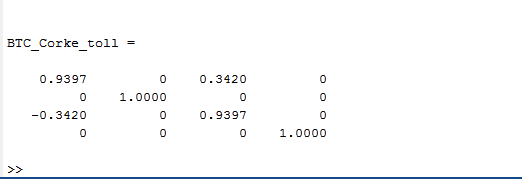
1. **BTC = inv (ATB) \* ATC**

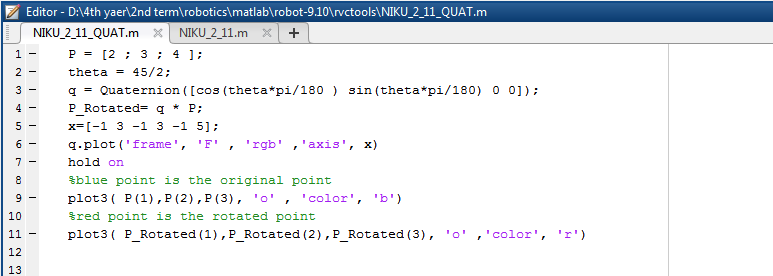
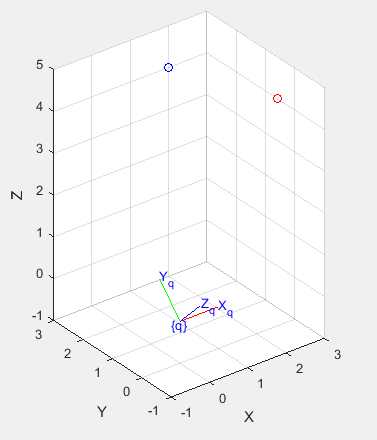
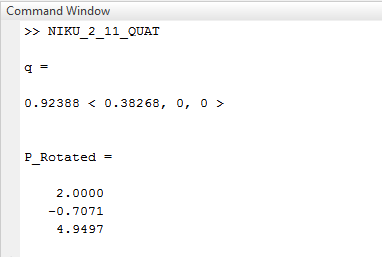
Note : ATC and BTC are stored variables from D(i)

**The results are equal.**



**The results are equal.**



**From [Niku 2010], resolve the following problems using quaternions and visualize your steps using the Robotics Toolbox of [Corke 2011]:**

