Funciones básicas toolbox Peter Corke 9.1

(Posición y Orientación)

Profesor: Ing. Pedro-F. Cárdenas, Ing. Ricardo E. Ramirez

Monitor: Julián A. Hernández R.

rotx roty rotz

Es una matriz de rotación SO(3) (3x3) que representa una rotación de theta radianes alrededor del eje indicado.

Nota: En la mayoria de funciones del toolbox de Peter Corke, si desea usar grados sexagesimales se debe usar "deg" como opción en la función.

```
R1 = rotx(pi/3)
```

```
R1 = 3×3
    1.0000         0         0
         0    0.5000   -0.8660
         0    0.8660    0.5000
```

```
R2 = rotx(60, "deg")
```

```
R2 = 3×3
    1.0000         0         0
         0    0.5000   -0.8660
         0    0.8660    0.5000
```

Entonces podemos usar rotaciones sucesivas al rededor de los 3 ejes:

```
R3 = rotx(pi/3)*roty(pi/5)*rotz(pi/6)
```

```
R3 = 3×3
    0.7006   -0.4045    0.5878
    0.6908    0.1785   -0.7006
    0.1785    0.8969    0.4045
```

rot2

Es una matriz de rotación SO(2)

```
R4= rot2(pi/3)
```

```
R4 = 2×2
    0.5000   -0.8660
    0.8660    0.5000
```

trotx troty trotz

Es una MTH matriz de transformación homogénea (4x4) que representa una rotación de theta radianes alrededor del eje x.

```
T1 = trotx(pi/3)
```

```
T1 = 4×4
    1.0000         0         0         0
         0    0.5000   -0.8660         0
```

```
         0    0.8660    0.5000         0
         0         0         0    1.0000
```

Nota: El componente de traslación es cero.

```
t1 = T1(1:3,4)
```

```
t1 = 3×1
     0
     0
     0
```

Podemos nuevamente operar las matrices post- o premultiplicando.

```
T3 = trotx(pi/3)*troty(pi/5)*trotz(pi/5)
```

```
T3 = 4×4
    0.6545   -0.4755    0.5878         0
    0.7057    0.1053   -0.7006         0
    0.2713    0.8734    0.4045         0
         0         0         0    1.0000
```

trot2

Es una MTH (3x3). Nuevamente con componente de translación igual a cero.

```
T4= trot2(pi/3)
```

```
T4 = 3×3
    0.5000   -0.8660         0
    0.8660    0.5000         0
         0         0    1.0000
```

r2t

Genera una MTH (4x4) a partir de una matriz de rotación R (3x3). Nótese que el componente de traslación será cero.

```
T5 = r2t(R3)
```

```
T5 = 4×4
    0.7006   -0.4045    0.5878         0
    0.6908    0.1785   -0.7006         0
    0.1785    0.8969    0.4045         0
         0         0         0    1.0000
```

Aunque esto es trivial, nos ahorra escribir lo siguiente:

```
T5 = [R3 [ 0 0 0]';
      0 0 0  1]
```

```
T5 = 4×4
    0.7006   -0.4045    0.5878         0
    0.6908    0.1785   -0.7006         0
    0.1785    0.8969    0.4045         0
         0         0         0    1.0000
```

rt2tr

Con esta función podemos obtener fácilmente una MTH a partir del vector de traslación t y la matriz de rotación R. (Aunque la formación de la MTH es trivial, esta función nos ahorra tiempo de escribir código)

```
t  = [1.5 2 3]'
```

```
t = 3×1
    1.5000
    2.0000
    3.0000
```

```
T6 = rt2tr(R3,t)
```

```
T6 = 4×4
    0.7006   -0.4045    0.5878    1.5000
    0.6908    0.1785   -0.7006    2.0000
    0.1785    0.8969    0.4045    3.0000
         0         0         0    1.0000
```

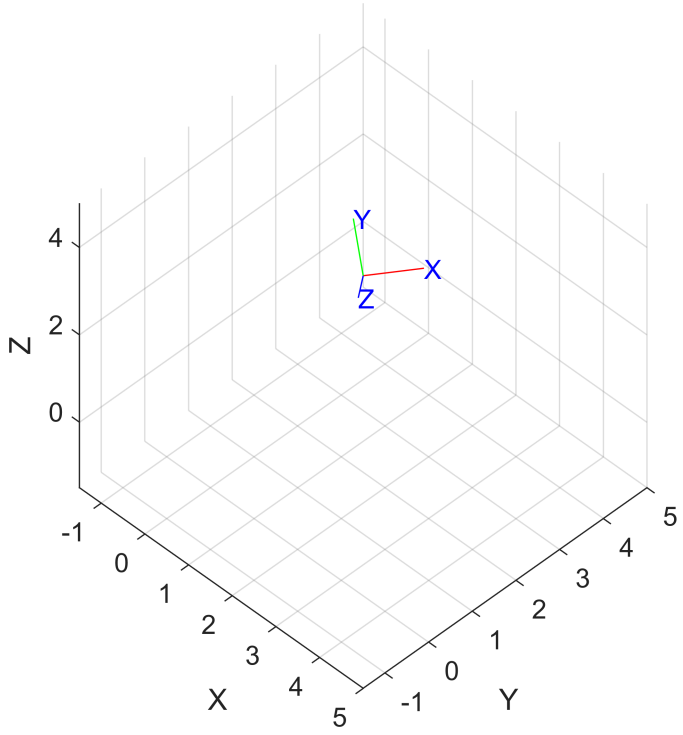
```
T6 = [R3 t;
      0 0 0  1]
```

```
T6 = 4×4
    0.7006   -0.4045    0.5878    1.5000
    0.6908    0.1785   -0.7006    2.0000
    0.1785    0.8969    0.4045    3.0000
         0         0         0    1.0000
```

trplot

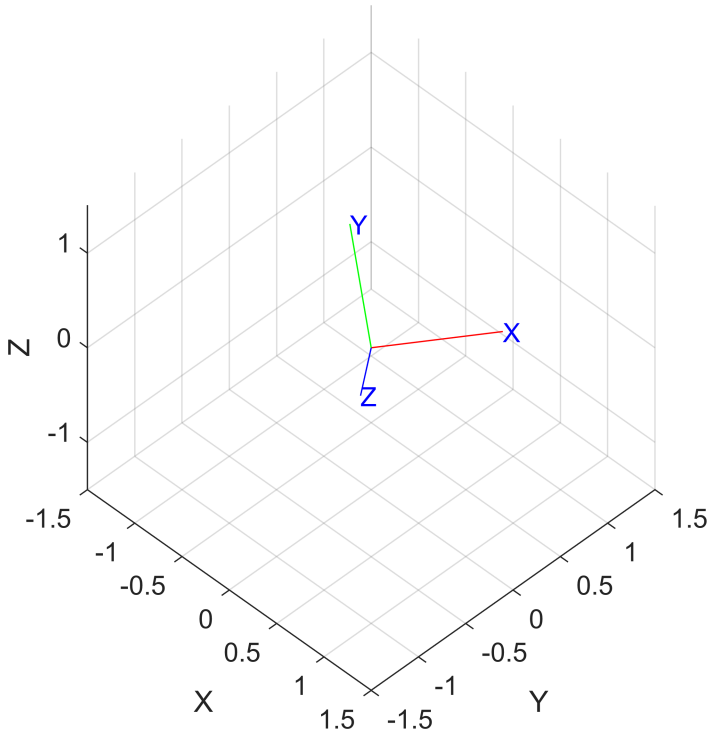
Permite dibujar un sistema coordenado a partir una MTH (4x4)

```
trplot(T6, 'axis', [-1.5 5 -1.5 5 -1.5 5],'view', [45,45], 'rgb')
```



o una matriz de rotación R(3x3), en este caso siempre en el origen.

```
trplot(R3, 'axis', [-1.5 1.5 -1.5 1.5 -1.5 1.5],'view', [45,45], 'rgb')
```



La función cuenta con muchas opciones para poder graficar el sistema coordenado con la estética deseada.

Options

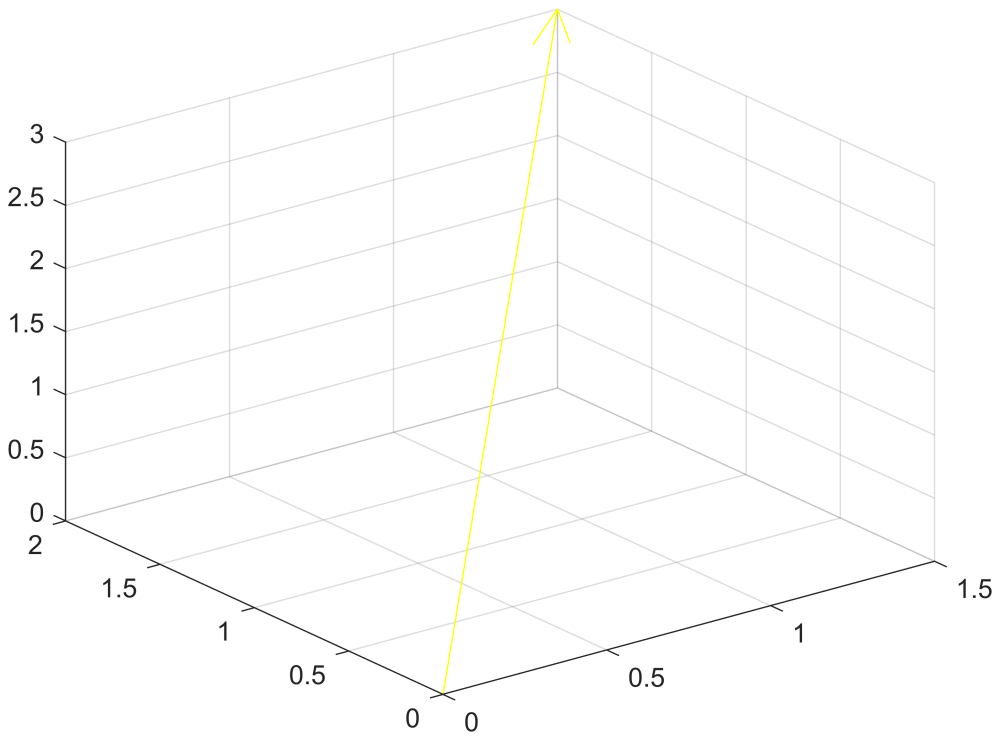
'color', C	The color to draw the axes, MATLAB colorspec C
'noaxes'	Don't display axes on the plot
'axis', A	Set dimensions of the MATLAB axes to A=[xmin xmax ymin ymax zmin zmax]
'frame', F	The coordinate frame is named {F} and the subscript on the axis labels is F.
'text_opts', opt	A cell array of MATLAB text properties
'handle', **H**	Draw in the MATLAB axes specified by the axis handle **H**
'view', V	Set plot view parameters V=[az el] angles, or 'auto' for view toward origin of coordinate frame
'length', s	Length of the coordinate frame arms (default 1)
'arrow'	Use arrows rather than line segments for the axes
'width', w	Width of arrow tips (default 1)
'thick', t	Thickness of lines (default 0.5)
'3d'	Plot in 3D using anaglyph graphics
'anaglyph', A	Specify anaglyph colors for '3d' as 2 characters for left and right (default colors 'rc'): chosen from r)ed, g)reen, b)lue, c)yan, m)agenta.
'dispar', D	Disparity for 3d display (default 0.1)
'text'	Enable display of X,Y,Z labels on the frame
'labels', L	Label the X,Y,Z axes with the 1st, 2nd, 3rd character of the string L
'rgb'	Display X,Y,Z axes in colors red, green, blue respectively
'rviz'	Display chunky rviz style axes

Figura 1: Opciones trplot.

quiver3 (no hace parte del toolbox pero es importante)

En este caso lo usaremos para plotear un vector de traslación pero esta función sirve para muchas más cosas y tiene muchas opciones para editar la estética del plot. https://la.mathworks.com/help/matlab/ref/quiver3.html

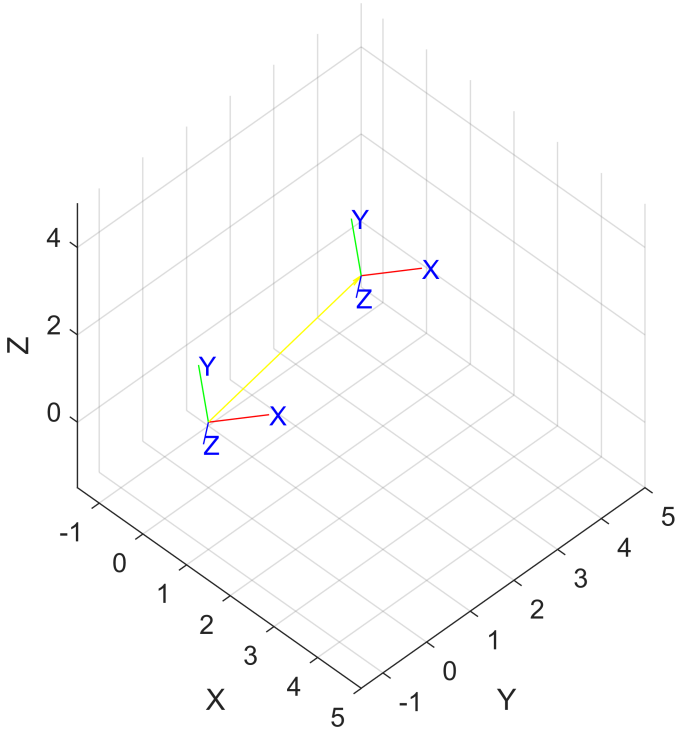
```
quiver3(0,0,0,t(1),t(2),t(3),1,'y')
```



hold (no hace parte del toolbox pero es útil para graficar).

Nos ayuda a mantener los plot pasados en el siguiente plot. Se activa con on y se desactiva con off.

```
trplot(R3, 'axis', [-1.5 5 -1.5 5 -1.5 5],'view', [45,45], 'rgb')
hold on
quiver3(0,0,0,t(1),t(2),t(3),1,'y')
trplot(T6, 'axis', [-1.5 5 -1.5 5 -1.5 5],'view', [45,45], 'rgb')
hold off
```



La siguiente imagen muestra las funciones disponibles en el toolbox para convertir entre representaciones.

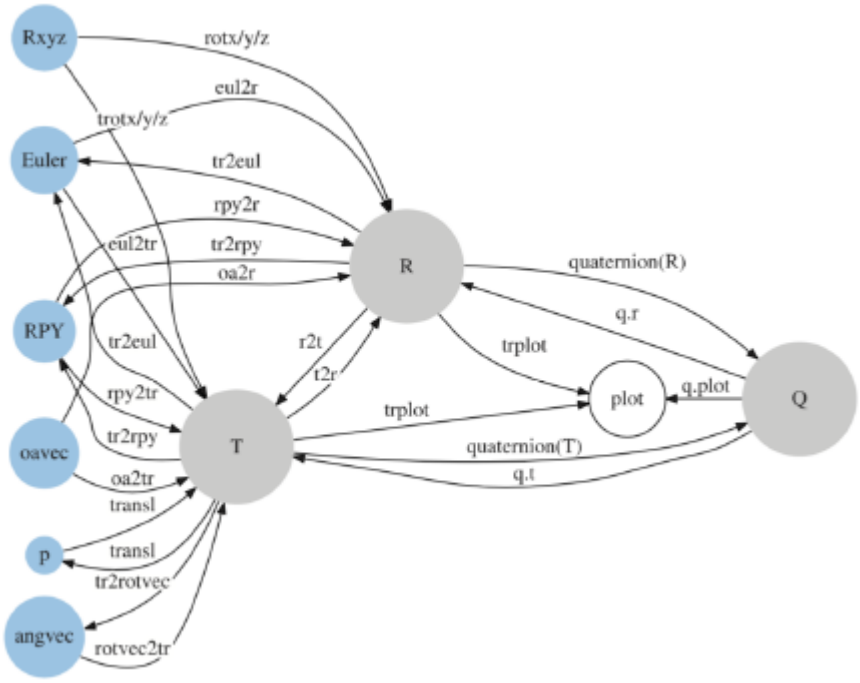


Figura 2: Conjunto de funciones toolbox Peter Corke 9.1

tr2rpy

Obtiene de una MTH la representación de orientación del sistema coordenado en ángulos roll pitch y yaw.

```
tr2rpy(T3)/pi
```

```
ans = 1×3
    0.3333    0.2000    0.2000
```

```
tr2rpy(T3,'deg')
```

```
ans = 1×3
   60.0000   36.0000   36.0000
```

Observamos que se obtienen los mismos valores de ángulos que se usaron para obtener T3 postmultiplicando rotaciones en ***x y z***.

Tambien podemos dar como parámetro de entrada la matriz de rotación.

```
tr2rpy(R3)/pi
```

```
ans = 1×3
    0.3333    0.2000    0.1667
```

transl

Aunque la obtención del vector de traslación a partir de una MTH es trivial también existe esta función.

```
t6 = transl(T6)
```

```
t6 = 3×1
    1.5000
    2.0000
    3.0000
```

```
t6 = T6(1:3,4)
```

```
t6 = 3×1
    1.5000
    2.0000
    3.0000
```

También nos permite a partir de una traslación obtener la MTH con como componente de orientación la matriz identidad. (Proceso también trivial)

```
T7 = transl(t)
```

```
T7 = 4×4
    1.0000         0         0    1.5000
         0    1.0000         0    2.0000
         0         0    1.0000    3.0000
         0         0         0    1.0000
```

```
T7 = [eye(3) t;
     0 0 0 1]
```

```
T7 = 4×4
    1.0000         0         0    1.5000
         0    1.0000         0    2.0000
         0         0    1.0000    3.0000
         0         0         0    1.0000
```

tr2eul

Permiten obtener la representación de la orientación en ángulos de Euler (ZYZ) a partir de una MTH o matriz de rotación.

```
eul = tr2eul(T3)
```

```
eul = 1×3
   -0.8728    1.1544    1.8720
```

eul2tr y eul2r

Son entonces la función inversas, es decir, obtenemos MTH (con componente de traslación cero) o matriz de rotación respectivamente a partir de ángulos de Euler.

```
eul2r(eul)
```

```
ans = 3×3
    0.6545   -0.4755    0.5878
    0.7057    0.1053   -0.7006
    0.2713    0.8734    0.4045
```

Se comprueba

```
rotz(eul(1))*roty(eul(2))*rotz(eul(3))
```

```
ans = 3×3
    0.6545   -0.4755    0.5878
    0.7057    0.1053   -0.7006
    0.2713    0.8734    0.4045
```

Quaternion

Es un clase para instaciar el objeto que representa la componente de rotación de la MTH en cuaternio. (los cuaternios en este toolbox se manejan como objetos y tienen métodos)

```
q = Quaternion(T6)
```

```
q =

0.75558 < 0.52859, 0.13542, 0.36242 >
```

q.T o q.R

Son métodos de la clase Quaternion y sirven para obtener la repesentación en MTH (con traslación igual a cero) o matriz de rotación del objeto Quaternion instanciado.

```
q.T
```

```
ans = 4×4
    0.7006   -0.4045    0.5878         0
    0.6908    0.1785   -0.7006         0
    0.1785    0.8969    0.4045         0
         0         0         0    1.0000
```

tr2angvec

Se obtiene la representación ángulo vector a partir de una MTH o matriz de rotación.

```
[ang  vec] = tr2angvec(T3)
```

```
ang = 1.4885
vec = 1×3
   -0.7897   -0.1588   -0.5926
```

oa2tr o oa2r

A partir del vector ***o*** (orientation) y ***a*** (aproach) de la convención NOA obtenemos MTH (con componente de traslación cero) o matriz de rotación respectivamente.

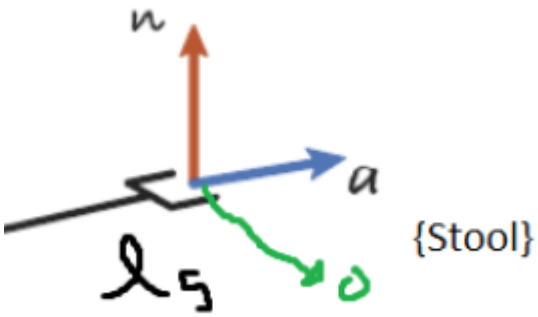


Figura 3: noa

Nota: N = o x a.

```
o = [0 0 1]';
a = [1 0 0]';
oa2tr(o,a)
```

```
ans = 4×4
     0     0     1     0
     1     0     0     0
     0     1     0     0
     0     0     0     1
```

Bibliografia:

[1]Peter Corke. Robotics Toolbox for Matlab, Release 9. 2015.