Assignment 2

Question 1: Zero-Order-Hold Discretization

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \tag{1}$$

First, we can derive the properties, for given **time variable** t_1 and t_2

$$e^{A(t1+t2)} = \sum_{k=0}^{\inf} \frac{1}{k!}A^k(t_1+t_2)^k \tag{2}$$

$$e^{At1}\cdot e^{At2} = \sum_{k1=0}^{\infty} \frac{1}{k!}A_1^k(t_1)_1^k \cdot \sum_{k2=0}^{\infty} \frac{1}{k_2!}A^{k2}(t_2)^{k2} \tag{3}$$

$$= \sum_{k1=0}^{\infty}\sum_{k2=0}^{\infty} A^{k1+k2}\frac{t_1^{k1}t_2^{k2}}{k_1!k_2!} \tag{4}$$

To calculate the summation of all points from $[0,+\infty)$, we can sum them one point a a time, or we can sum them by using a moving slash line(show as figure below, in fact we exchanged the variables)

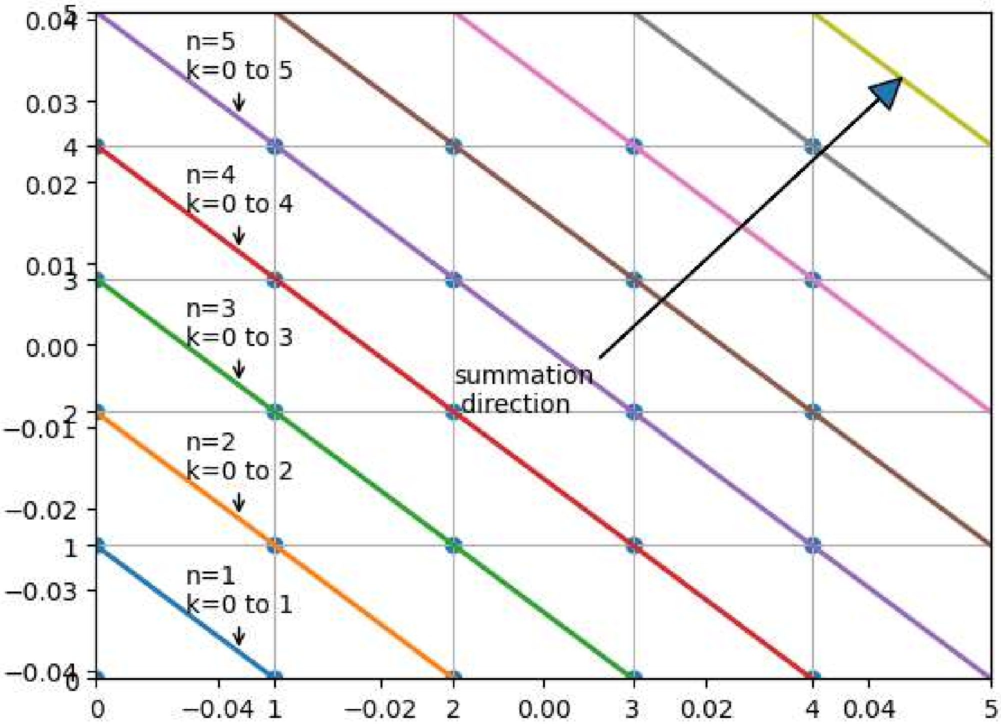
So the summation can be calculated in the direction of $k_1 + k_2 = n$

$$e^{At1}\cdot e^{At2} = \sum_{n=0}^{\infty}\sum_{k=0}^{n} A^n\frac{t_1^k \cdot t_2^{(n-k)}}{k!(n-k)!} \tag{5}$$

$$= \sum_{n=0}^{\infty} A^n\frac{(t_1+t_2)^n}{n!} \tag{6}$$

$$= e^{A(t1+t2)} \tag{7}$$

In [ ]:
```python
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
x = np.arange(0,5)
y = np.arange(0,5)
x_m,y_m = np.meshgrid(x,y)
plt.figure()
plt.axis('equal')
ax = plt.axes()
ax.scatter(x_m,y_m)
for i in range(1,10):
    xx = np.linspace(0,i,100)
    yy = i-xx
    ax.plot(xx,yy,linewidth=2)
    plt.annotate('n={}\nk=0 to {}'.format(i,i),xy=(0.8,i-1+0.2),xytext=(0.5,i-1+0.5),
                 arrowprops=dict(arrowstyle="->",connectionstyle="arc3"))
plt.annotate('summation\n direction',xytext=(2,2),xy=(4.5,4.5),arrowprops=dict(width=0.2))
plt.grid()
plt.xlim([0,5])
plt.ylim([0,5])
plt.show()
```



Next part is about Discretization. $x(t)$ is a continuous time signal only have value in $t > 0$

$$
\begin{aligned}
x[n] =& x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - n\delta t) && (8)\\
=& \sum_{n=0}^{\infty} x(n\delta t)\delta(t - n\delta t) && (9)
\end{aligned}
$$

when the signal pass through a linear system $h(t)$, $h(t)$ only have value when $t > 0$

$$
\begin{aligned}
y(t) &= \int_{-\infty}^{\infty} \sum_{n=0}^{\infty} x(n\delta t)\delta(t - n\delta t)h(\tau)d\tau && (10)\\
&= \int_{0}^{\infty} \sum_{n=0}^{\infty} x(n\delta t)\delta(t - n\delta t)h(\tau)d\tau && (11)\\
&= \sum_{n=0}^{\infty} x(n\delta t) \int_{0}^{\infty} \delta(t - n\delta t)h(\tau)d\tau && (12)\\
&= \sum_{n=0}^{\infty} x(n\delta t)h(t - n\delta t) && (13)
\end{aligned}
$$

For zero-order hold, $x[n]$ pass through $h(t) = u(t) - u(t - \delta t)$ and $y(t) = x(n\delta t)$ for $n\delta t < t < (n+1)\delta t$

x_k is the kth term $x(n\delta t)$ corresponds to $\delta(t - k\delta t)$

$$
x_k = x(k\delta t) = e^{Ak\delta t}x_0 + \int_0^{k\delta t} e^{A(k\delta t - \tau)}Bu(\tau)d\tau
$$

$$
\begin{aligned}
x_{k+1} =& x(k\delta t + \delta t) = e^{Ak\delta t + \delta t}x_0 + \int_0^{k\delta t + \delta t} e^{A(k\delta t + \delta t - \tau)}Bu(\tau)d\tau && (14)\\
=& e^{\delta t}e^{Ak\delta t}x_0 + e^{\delta t}\int_0^{k\delta t} e^{A(k\delta t - \tau)}Bu(\tau)d\tau + \int_{k\delta t}^{k\delta t + \delta t} e^{A(k\delta t + \delta t - \tau)}Bu(\tau)d\tau && (15)\\
=& e^{\delta t}x_k + \int_{k\delta t}^{k\delta t + \delta t} e^{A(k\delta t + \delta t - \tau)}Bu(\tau)d\tau && (16)
\end{aligned}
$$

According to the question $u(t)$ is a zero-hold output $y_u(t)$ of original $u_0(t)$, so it is a constant u_k on $k\delta t < t < k\delta t + \delta t$ and we have

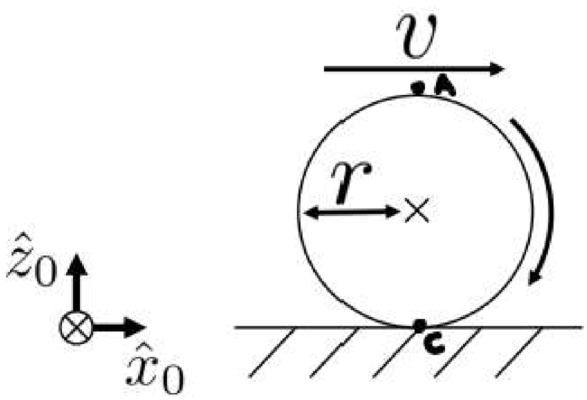
$$\int_{k\delta t}^{k\delta t+\delta t} e^{A(k\delta t+\delta t-\tau)}Bu(\tau)d\tau = \int_{k\delta t}^{k\delta t+\delta t} e^{A(k\delta t+\delta t-\tau)}Bu_k d\tau \tag{17}$$
$$=e^{A(k\delta t+\delta t)}\int_{k\delta t}^{k\delta t+\delta t} e^{-A\tau}Bu_k d\tau \tag{18}$$
$$=e^{A(k\delta t+\delta t)}(e^{-Ak\delta t}-e^{-A(k\delta t+\delta t)})A^{-1}Bu_k \tag{19}$$
$$=(e^{A\delta t}-I)A^{-1}B\cdot u_k \tag{20}$$

$$where \quad \int e^{At}Bdt = A^{-1}e^{At}B+C = e^{At}A^{-1}B+C$$

So $A_k = e^{\delta t}, B_k = (e^{A\delta t}-I)A^{-1}B$

$$x_{k+1} = A_k x_k + B_k u_k$$

Question 2: Spatial Velocity



(1)

The linear velocity of contact point C is $\frac{dC_x}{dt} = v$.

(2)

The linear velocity of top point A is $\frac{dA_x}{dt} = 2v$

(3)

Choose the center of the cylinber P as the body fixed point on axis. The velocity of body fixed point currently coincides with C is

$$v_c = v_p + \omega \times \overrightarrow{PC}$$

$$^o v_c =^o v_p +^o \omega \times^o \overrightarrow{PC} = [0,0,0]^T$$

(4)

The velocity of body fixed point currently coincides with A is

$$v_a = v_p + \omega \times \overrightarrow{PA}$$

$$^o v_a =^o v_p +^o \omega \times^o \overrightarrow{PA} = [2v,0,0]^T$$

(5)

The velocity of body fixed point currently coincides with O is

$$v_o = v_p + \omega \times \overrightarrow{PO}$$

$$^o v_o =^o v_p +^o \omega \times^o \overrightarrow{PO} = [0, 0, \omega C_x]^T$$

$$^o \nu = [^o \omega,^o v_p]^T = [0, \omega, 0, 0, 0, \omega C_x]^T$$

(6)

The velocity of body fixed point currently coincides with C is

$$v_c = v_p + \omega \times \overrightarrow{PC}$$

$$^c v_c =^c v_p +^c \omega \times^o \overrightarrow{PC} = [0, 0, 0]^T$$

$$^o \nu = [^c \omega,^c v_c]^T = [0, \omega, 0, 0, 0, 0]^T$$

Question 3: Twist

(a)

Position of point P

$$^s p =^s \overrightarrow{OP} = [L + dsin\theta, L - dcos\theta]^T$$

(b)

Velocity of point P

$$\frac{d^s p}{dt} = [dcos\theta, dsin\theta, 0]^T$$

(c)

The configuration of {b} seen from {s} is

$$^s T_b = \begin{bmatrix} ^s R_b & ^s p_b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} c_\theta & -s_\theta & 0 & L + ds_\theta \\ s_\theta & c_\theta & 0 & L - dc_\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(d)

Let the point r be the body fixed point in rotation axis.

The velocity of body fixed point corresponds to O_b is

$$^b v_p =^b v_r +^b \omega \times^b \overrightarrow{ro_b} = [d, 0, 0]^T$$

The twist of $^s T_b$ in {b} is

$$^b \nu = [0, 0, 1, d, 0, 0]$$

(e)

The velocity of body fixed point corresponds to O_s is

$$^s v_{Os} =^s v_r +^s \omega \times^s \overrightarrow{ro_s} = [L, -L, 0]^T$$

The twist of $^s T_b$ in {s} is

$$^s\nu = [0,0,1,L,-L,0]$$

(f)

$$^s\omega = ^sR_b{}^b\omega \tag{21}$$
$$^sv_{ob} = ^sR_b{}^bv_{ob} + [^sp_b]^sR_b{}^b\omega \tag{22}$$

(g)

The linear velocity part of twist $^b\nu$ is $\dot{p}$ expressed in {b} frame.

(h)

The linear velocity part of twist $^s\nu$ is $\dot{p} -^s \omega \times p$.

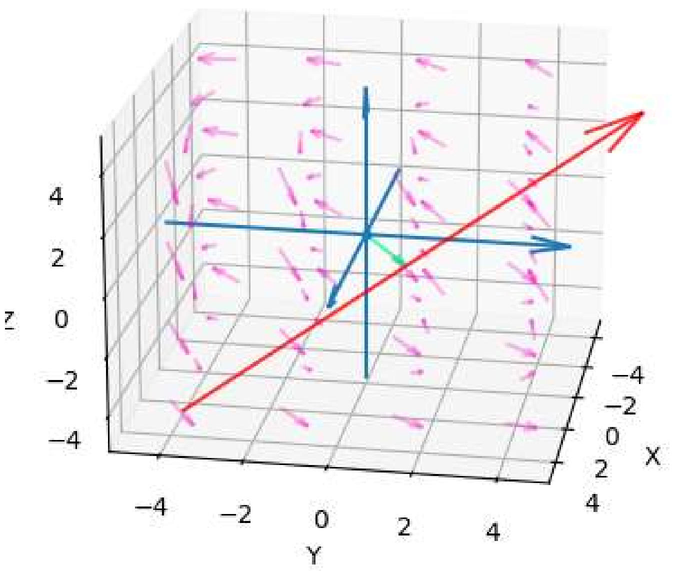
Question 4: Screw axis and its transformation

(a)

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
w = np.array([0,2,2])
v = np.array([4,0,0])
s = w/np.linalg.norm(w)
theta_d = np.linalg.norm(w)
q = np.cross(w, v)/(np.linalg.norm(w)**2)
h = w.dot(v)/(np.linalg.norm(w))
print('rotation direction={} in deg'.format(s*180/np.pi))
print('axis translation={}'.format(q))
print('screw pitch = {}'.format(h))
print('rotation speed={}'.format(theta_d))
L = 4
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.grid()
ax.quiver(-5,0,0,1,0,0,normalize=True,length=10,arrow_length_ratio=0.1)
ax.quiver(0,-5,0,0,1,0,normalize=True,length=10,arrow_length_ratio=0.1)
ax.quiver(0,0,-5,0,0,1,normalize=True,length=10,arrow_length_ratio=0.1)
ax.quiver(0,0,0,q[0],q[1],q[2],normalize=True,length=np.linalg.norm(q),color=(0,1,0.5,0.7))
ax.quiver(q[0]-s[0]*2*L,q[1]-s[1]*2*L,q[2]-s[2]*2*L,s[0],s[1],s[2],
          normalize=True,length=4*L,arrow_length_ratio=0.1,color=(1, 0, 0, 0.8))
ax.scatter(0,0,0,'linewidth',10)
ax.set_xlabel('X')
ax.set_xlim3d(-5, 5)
ax.set_ylabel('Y')
ax.set_ylim3d(-5, 5)
ax.set_zlabel('Z')
ax.set_zlim3d(-5, 5)
x = np.linspace(-4,4,4)
y = np.linspace(-4,4,4)
z = np.linspace(-4,4,4)
x_,y_,z_ = np.meshgrid(x,y,z)
x_ = np.reshape(x_,(1,np.size(x_)))
y_ = np.reshape(y_,(1,np.size(y_)))
z_ = np.reshape(z_,(1,np.size(z_)))
p = np.vstack([x_,y_,z_])
v_ = np.zeros((3,np.size(x_)))
for i in range(np.size(x_)):
    v_[:,i] = v+np.cross(w,p[:,i])
    ax.quiver(x_[:,i],y_[:,i],z_[:,i],v_[0,i],v_[1,i],v_[2,i],normalize=True,length=0.08*np.linalg.norm(v_
              color=(1,0,0.8,0.3),arrow_length_ratio=0.4)
ax.view_init(elev=20, azim=10)
plt.show()
```

```
rotation direction=[ 0.          40.51423423 40.51423423] in deg
axis translation=[ 0.  1. -1.]
screw pitch = 0.0
rotation speed=2.8284271247461903
```



(b)

Let an arbitary frame {A} rigidly attached to the skew axis S

Let frame {B} be the frame obtained by applying T transformation.

In frame {B} there is a skew axis S' that has the same coordinate ${}^BS'$ with ${}^AS$

$$ {}^AS = {}^BS' $$

Multiply both sides with ${}^AX_B$, then

$$ {}^AX_B{}^AS = [Ad_T]{}^AS = {}^AS' $$

(c)

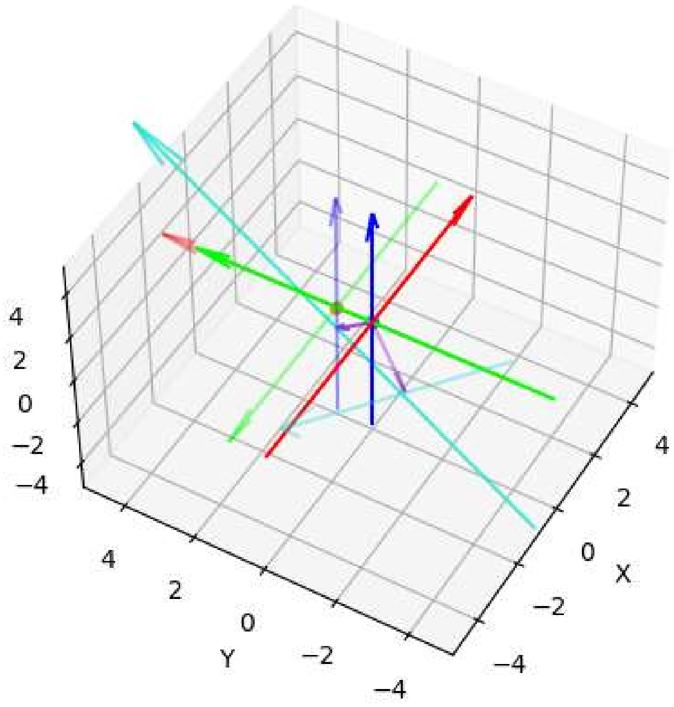
i.

$$ T = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

In [ ]:
```python
import numpy as np
from scipy.spatial.transform import Rotation as R
r = R.from_euler('z', 90, degrees=True).as_matrix()
p = np.array([2,1,3]).transpose()
eijk = np.zeros((3,3,3))
eijk[0,1,2] = eijk[1,2,0]= eijk[2,0,1]= 1
eijk[0,2,1] = eijk[2,1,0]= eijk[1,0,2]=-1
wp = np.einsum('ijk,k->ij', eijk, p)
AdT = np.block([[r,np.zeros((3,3))]
                ,[wp*r,r]])
S = np.array([0,2,2,4,0,0])
S_ = np.dot(AdT,S)
w = S[0:3]
v = S[3:]
s = w/np.linalg.norm(w)
theta_d = np.linalg.norm(w)
q = np.cross(w, v)/(np.linalg.norm(w)**2)
```

```python
h = w.dot(v)/(np.linalg.norm(w))
print('rotation direction={} in deg'.format(s*180/np.pi))
print('axis translation={}'.format(q))
print('screw pitch = {}'.format(h))
print('rotation speed={}'.format(theta_d))
L = 4
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.grid()
ax.quiver(-5,0,0,1,0,0,normalize=True,length=10,arrow_length_ratio=0.1,color=(1,0,0,1))
ax.quiver(0,-5,0,0,1,0,normalize=True,length=10,arrow_length_ratio=0.1,color=(0,1,0,1))
ax.quiver(0,0,-5,0,0,1,normalize=True,length=10,arrow_length_ratio=0.1,color=(0,0,1,1))
ax.quiver(0,0,0,q[0],q[1],q[2],normalize=True,length=np.linalg.norm(q),color=(139/255,31/255,207/255,0.8))
ax.quiver(q[0]-s[0]*2*L,q[1]-s[1]*2*L,q[2]-s[2]*2*L,s[0],s[1],s[2],
          normalize=True,length=4*L,arrow_length_ratio=0.1,color=(36/255,227/255, 208/255, 0.8))
w = S_[0:3]
v = S_[3:]
s = w/np.linalg.norm(w)
theta_d = np.linalg.norm(w)
q = np.cross(w, v)/(np.linalg.norm(w)**2)
h = w.dot(v)/(np.linalg.norm(w))
print('rotation direction={} in deg'.format(s*180/np.pi))
print('axis translation={}'.format(q))
print('screw pitch = {}'.format(h))
print('rotation speed={}'.format(theta_d))
L = 4
ax.quiver(0,-4,0,0,1,0,normalize=True,length=10,arrow_length_ratio=0.1,color=(1,0,0,0.4))
ax.quiver(5,1,0,-1,0,0,normalize=True,length=10,arrow_length_ratio=0.1,color=(0,1,0,0.4))
ax.quiver(0,1,-5,0,0,1,normalize=True,length=10,arrow_length_ratio=0.1,color=(0,0,1,0.4))
ax.quiver(0,0,0,q[0],q[1],q[2],normalize=True,length=np.linalg.norm(q),color=(139/255,31/255,207/255,0.4))
ax.quiver(q[0]-s[0]*2*L,q[1]-s[1]*2*L,q[2]-s[2]*2*L,s[0],s[1],s[2],
          normalize=True,length=4*L,arrow_length_ratio=0.1,color=(36/255,227/255, 208/255, 0.4))
ax.scatter(0,0,0,'linewidth',20)
ax.scatter(0,1,0,'linewidth',20)
ax.set_xlabel('X')
ax.set_xlim3d(-5, 5)
ax.set_ylabel('Y')
ax.set_ylim3d(-5, 5)
ax.set_zlabel('Z')
ax.set_zlim3d(-5, 5)
ax.view_init(elev=47, azim=-150)
plt.show()
```

```
rotation direction=[ 0.         40.51423423 40.51423423] in deg
axis translation=[ 0.  1. -1.]
screw pitch = 0.0
rotation speed=2.8284271247461903
rotation direction=[-4.05142342e+01  8.99596713e-15  4.05142342e+01] in deg
axis translation=[-1.  -1.5 -1. ]
screw pitch = 4.242640687119285
rotation speed=2.8284271247461903
```



4
2
0
−2
−4
4
2
0
Y
−2
−4
4
2
0
X
−2
−4