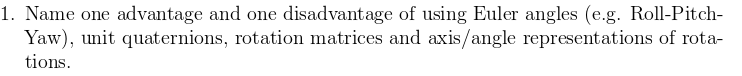
# **Assignment Research Internship**

### **Sec 2.1.3**



**Ans 1.** Various angle representations - their advantages and disadvantages

**Unit Quaternions**

**Advantage** - They offer a compact representation with ease of algebraic application. Allowing efficient interpolation as slerp. Continuous representation. Easy to find inverse and to represent as a combined transform for a chain of rotations.

**Disadvantage** - The representation is not visually obvious, that is hard to visualize. I redundant variable as 3DOF and 4 variables.

**Axis** **Angle** -

**Advantage** - Minimal representation, very intuitive and easy to invert

**Disadvantage** - Hindered algebraic operability. Difficult to Chain.

**Matrix rotation** -

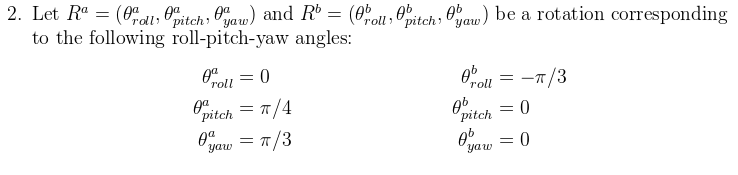
**Advantage** - ease of computer implementation and easy to chain as one combined transformation.

**Disadvantage** - require 9 variable for the representation causing redundancy and are more computationally expensive than others. Hard to interpolate.

**Euler Angles**

**Advantage** - It’s close representation of cases like an airplane, where rotations are restricted along particular axis - like Yaw pitch roll and in gimbal setup of gyroscope. Like the hardware limitation on gimbals Euler angle function with similar restriction and exhibit similar behaviour. Thus making them better for modelling and understanding such behaviour. Since rotation is expressed along fixed orthonormal axis,there easy to visualize and at representing orientation in least number of variables

**Disadvantage** - For rotations they suffer from limitations - if using euler angles. interpolation between two orientation will usually result in a non-optimal path. Also the representation is discontinuous.





**Ans 2.** Taking - ***z-y’-x″* (intrinsic rotations)** or *z-y-x* (extrinsic rotations): the intrinsic rotations are known as: **yaw, pitch and roll** from Wiki.

Rotation matrices can be used to represent a sequence of **intrinsic rotations**. For instance,

R = X(\alpha) Y(\beta) Z(\gamma)

represents a composition of intrinsic rotations about axes *x-y’-z″*, if used to **pre-multiply** column vectors. From - Wiki

With above convention -

1. R(roll, pitch,yaw)

Please Enter Yaw : 600

Please Enter Pitch : 450

Please Enter Roll : 00

Ra  = 0.353553 -0.612372 0.707107

0.866025 0.5 0

-0.353553 0.612372 0.707107

b) R(roll, pitch,yaw)

Please Enter Yaw : 0

Please Enter Pitch : 0

Please Enter Roll : -60

Rb  = 1 0 0

0 0.5 0.866025

0 -0.866025 0.5



**Ans 3.** qa and qb corresponding to Ra and Rb respectively, as calculated above:

qa.w = 0.800103, qa.x = 0.191342, qa.y = 0.331414, qa.z = 0.46194.

qa = (0.800103 + 0.191342 i + 0.331414 j + 0.46194 k )

qb.w = 0.866025, qb.x = -0.5, qb.y = 0, qb.z = 0.

qb = (0.866025 + -0.5 i + 0 j + 0 k )

Yes these representations are unique as far as rotations without scaling are considered.



**Ans 4.**

qc.w = 0.78858, qc.x = -0.234345, qc.y = 0.0560428, qc.z = 0.565759.

qd.w = 0.78858, qd.x = -0.234345, qd.y = 0.517983, qd.z = 0.234345.

No they are not the same. As rotations are not commutative, multiplication of unit Quaternions would in most cases give different result in when order of application is changed.



**Ans 5.**

qa.w = 0.800103, qa.x = 0.191342, qa.y = 0.331414, qa.z = 0.46194.

qf.w = 0.800103, qf.x = 0.191342, qf.y = 0.331414, qf.z = 0.46194.

Verified that qf = qa .

### **CODE**







### **OUTPUT**

Please Enter Yaw : 60

Please Enter Pitch : 45

Please Enter Roll : 0

0.353553 -0.612372 0.707107

0.866025 0.5 0

-0.353553 0.612372 0.707107

Quat.w = 0.800103 Quat.x = 0.191342 Quat.y = 0.331414 Quat.z = 0.46194

End of Rotation Calculations

Qc.w = 0.78858 Qc.x = -0.234345 Qc.y = 0.0560428 Qc.z = 0.565759

Qd.w = 0.78858 Qd.x = -0.234345 Qd.y = 0.517983 Qd.z = 0.234345

Qa.w = 0.800103 Qa.x = 0.191342 Qa.y = 0.331414 Qa.z = 0.46194

Qf.w = 0.800103 Qf.x = 0.191342 Qf.y = 0.331414 Qf.z = 0.46194