CSCI 520 - ASSIGNMENT 2

Motion Capture Interpolation

Key Points:

1) Implemented Bezier Euler Interpolation using DeCasteljau Construction for Control Points Calculations.
2) Implemented Linear Quaternion Interpolation (SLERP).
3) Implemented Bezier Quaternion Interpolation using DeCasteljau Construction for Control Points Calculations

Demo Videos

Martial_Arts_Linear_Quaternion.mp4

Martial_Arts_Bezier_Euler.mp4

Martial_Arts_Bezier_Quaternion.mp4

Submission:

In my Submission,

- You can check all graphs, videos and other result files in the 'Results' folder.
- Other than requested Graphs and Videos there are some other Extra Results too.
- I have also implemented a Graph Generator project which can be used to generate graphs based on different motion files. [You can use GraphGenerator.exe in 'TestStage' folder]
- [Extra] I have implemented Non-Uniform Keyframe Interpolation in the Interpolation project. In this project if you set N = -1. It will check all frames and see if rotation of any bone is missing and interpolate those values based on given rotations for the respective bone.
- [Extra] I have added all the results of Non-Uniform Keyframe Interpolation AMC files in the 'Results/KF' folder.
- I also included all executables with some Test case files in the 'Test Stage' folder. You can use Test file 'X' in this folder to also test Non-Uniform Keyframe interpolation too.

Interpolation Techniques

Linear Euler:
In Linear Euler we interpolate individual Euler angles (pitch, yaw, roll) linearly between start point and endpoint to calculate points in between.

Bezier Euler:
In Bezier Euler we interpolate individual Euler angles using Bezier spline interpolation.
At first we calculate control points using DeCasteljau Construction. Using these 4 points we can interpolate points on the Bezier Spline.

Linear Quaternion(SLERP):
In SLERP, instead of using euler angles we use Quaternions which is a different and much better way of representing a rotation using imaginary numbers**.** The Quaternions are interpolated linearly over the unit Quaternion Sphere instead of interpolating them linearly. Interpolating them on Sphere maintains the rotation as each quaternion rotation should have magnitude of 1.

Bezier Quaternion:
In Bezier Quaternion, we interpolate Quaternions over unit Quaternion Sphere but instead of doing it linearly we use Bezier splines. First we need to calculate control point quaternions using DeCasteljau Construction. Then using these four quaternions we calculate new interpolated points on the Bezier spline.

Non-Uniform Keyframe Interpolation

I have implemented Non-Uniform Keyframe Interpolation, you can test this in the Interpolate project, just by setting N = -1 while passing arguments and also using the AMC test file which only contains Keyframes.
In this we iterate through each bone separately, then find the start frame and end frame for this bone to interpolate between using data from the AMC file.
After finding the start frame and end frame, we can interpolate between these two frames for this particular bone based on the given technique(any technique from above).

Example: interpolate skeleton.asf motion.amc l e -1 outputMotion.amc

Testing

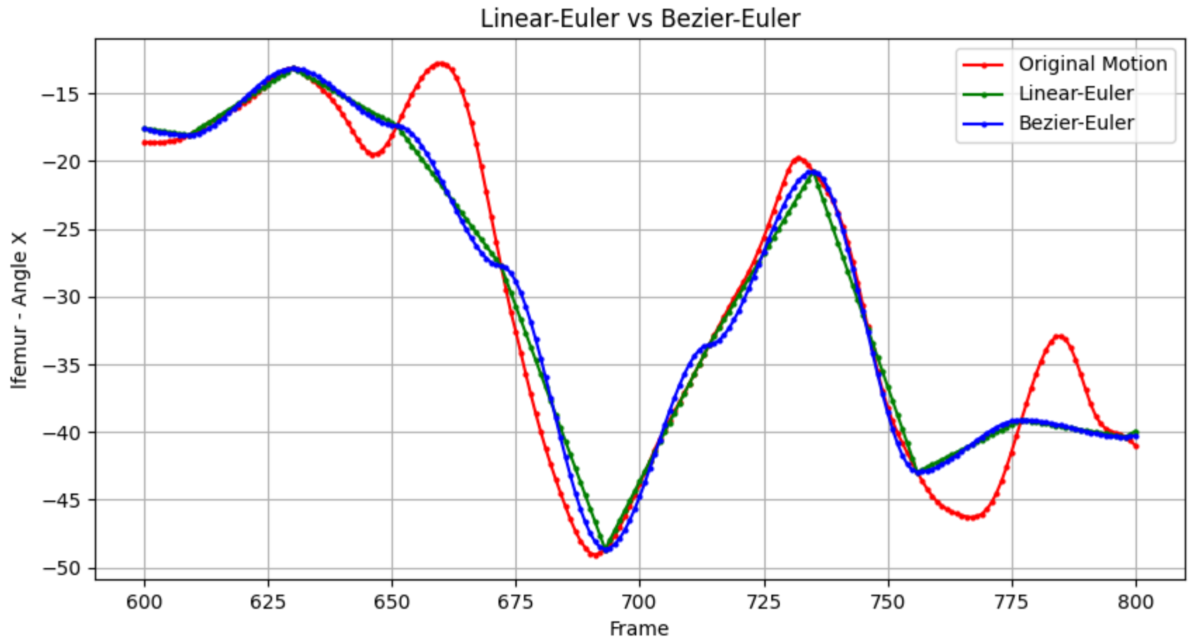
- The Test Stage folder contains two test files which were already created as a keyframe animation file from martial arts.
- Did not have enough time to create original key frame animations, so I used martial arts amc file, and removed many joint rotations from almost all the frames and were randomly kept in every 20-40 frames.

- For example in ‘TestStage/martial_arts_keyFrame_legs.amc’, ["rfemur", "rtibia", "rfoot", "rtoes", "lfemur", "ltibia", "lfoot", "ltoes"] these joints wee removed from all the frames and were kept randomly in every 20-30 frames
- The two test files are ‘TestStage/martial_arts_keyFrame_legs.amc’ and ‘TestStage/martial_arts_keyFrame_hands_legs.amc’
- The results for both these test files are stored in ‘Results/KF’
- I have also created a ‘TestStage/KeyFrameFileExtractor.py’ just in case you need to create more files similar to this with different animations.

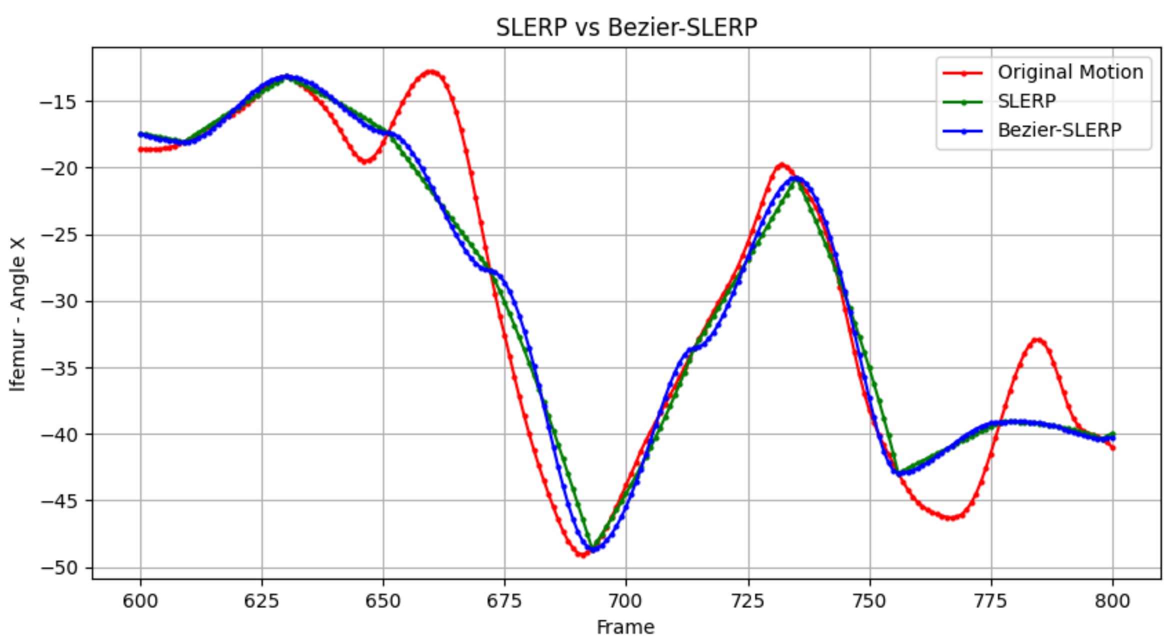
```
#!OML:ASF H:\Martial Arts Walks\Project 1\Capture day 1\Session 1\male.ASF
:FULLY-SPECIFIED
:DEGREES
1
root -2.95645 16.8291 -36.2757 3.79203 -0.409345 -2.77144
lowerback 7.02591 2.55716 -1.97249
upperback -3.14202 3.31583 1.27229
thorax -7.43554 1.75645 1.99661
lowerneck -4.68188 -4.04088 -6.84108
upperneck 4.69893 -5.30737 6.47589
head 2.76948 -2.5844 3.14763
rclavicle 1.91563e-015 -9.24349e-015
rhumerus -32.8355 4.71181 -81.727
rradius 32.9783
rwrist 3.84135
rhand -24.1523 -25.0894
rfingers 7.12502
rthumb 2.33029 -55.0564
lclavicle 1.91563e-015 -9.24349e-015
lhumerus -27.0689 -5.33108 89.6453
lradius 25.7364
lwrist -30.6252
lhand -28.8405 29.9909
lfingers 7.12502
lthumb -2.19771 59.9594
rfemur -19.1521 0.15077 14.7318
rtibia 40.3652
rfoot -23.1652 -2.86285
rtoes -21.0274
lfemur -18.1857 -3.85478 -7.37129
ltibia 37.6337
lfoot -17.8071 6.78532
ltoes -26.1578
2
root -2.95647 16.8281 -36.2789 3.83496 -0.515041 -2.77044
lowerback 7.0168 2.56529 -1.92512
upperback -3.13081 3.33086 1.26317
thorax -7.41553 1.76102 1.95907
lowerneck -4.72931 -3.99134 -6.7546
upperneck 4.68931 -5.23907 6.40124
head 2.76328 -2.55005 3.10787
3
root -2.95654 16.8272 -36.2808 3.87536 -0.609706 -2.74215
lowerback 6.97964 2.56886 -1.91256
upperback -3.12187 3.33669 1.25357
thorax -7.38337 1.76276 1.94295
lowerneck -4.65596 -3.96919 -6.78933
upperneck 4.65659 -5.21207 6.43618
head 2.73923 -2.5375 3.12874
```

In the above image you can see few bones were removed in some frames

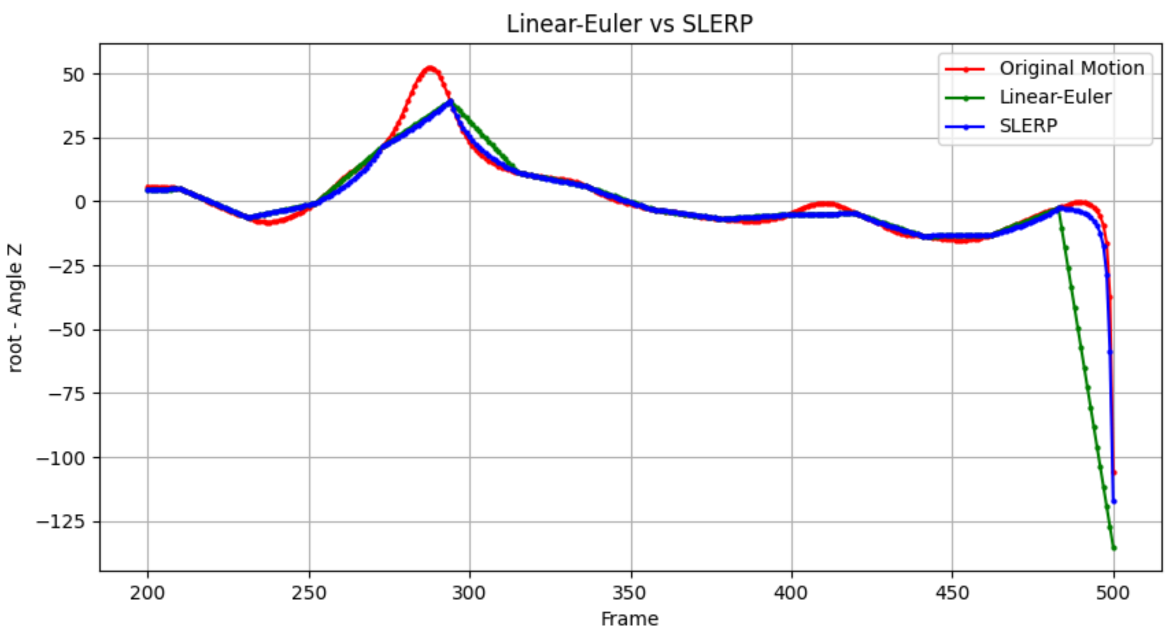
Interpolation Techniques Comparison



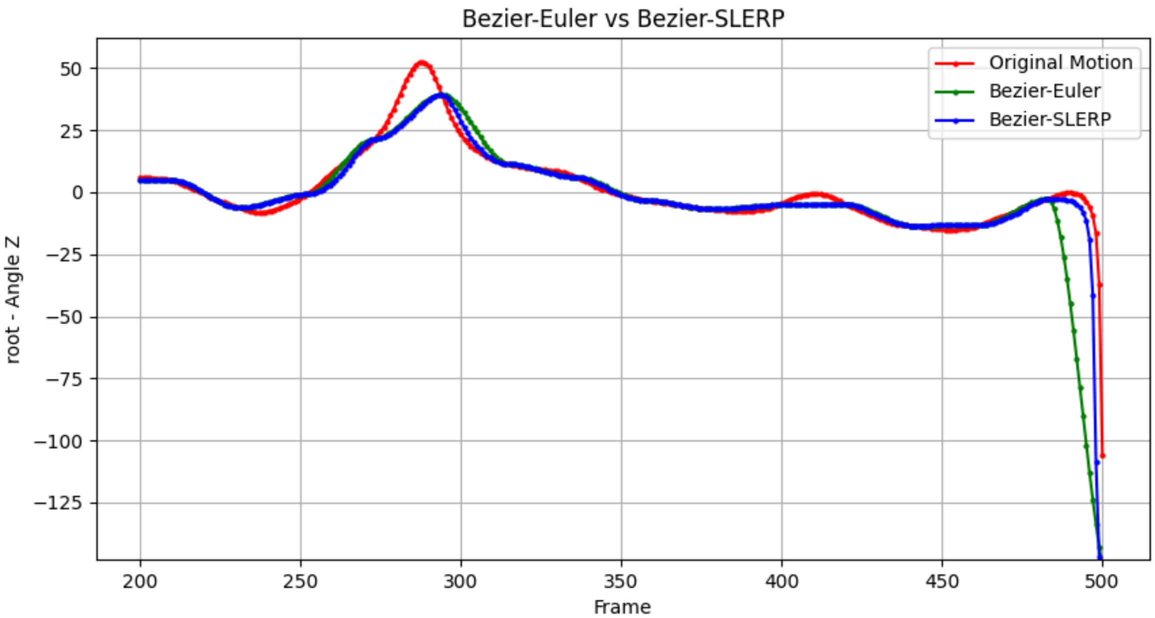
In the above graph, we can compare Linear Euler and Bezier Euler to the original motion. As you can see Bezier Euler looks much more smoother compared to Linear Euler as it is interpolated on Bezier Spline which is constructed using control points considering past and future points based on DeCasteljau Construction.
You can observe this especially between 675-700 frames and 725-750 frames.



In the above graph, we can compare SLERP and Bezier SLERP to the original motion.
As you can see Bezier Euler looks much more smoother compared to Linear Euler as it is interpolated on Bezier Spline which is constructed using control points considering past and future points based on DeCasteljau Construction.
You can observe this especially between 675-700 frames and 725-750 frames.



In the above graph we can compare Linear Euler and SLERP to the original motion.
In this you can see SLERP is much more smoother and similar to original motion compared to Linear Euler.
This is because we spherically interpolate Quaternions in SLERP whereas we linearly interpolate individual Euler angles in Linear Euler.
You can observe this especially between frames 475-500.



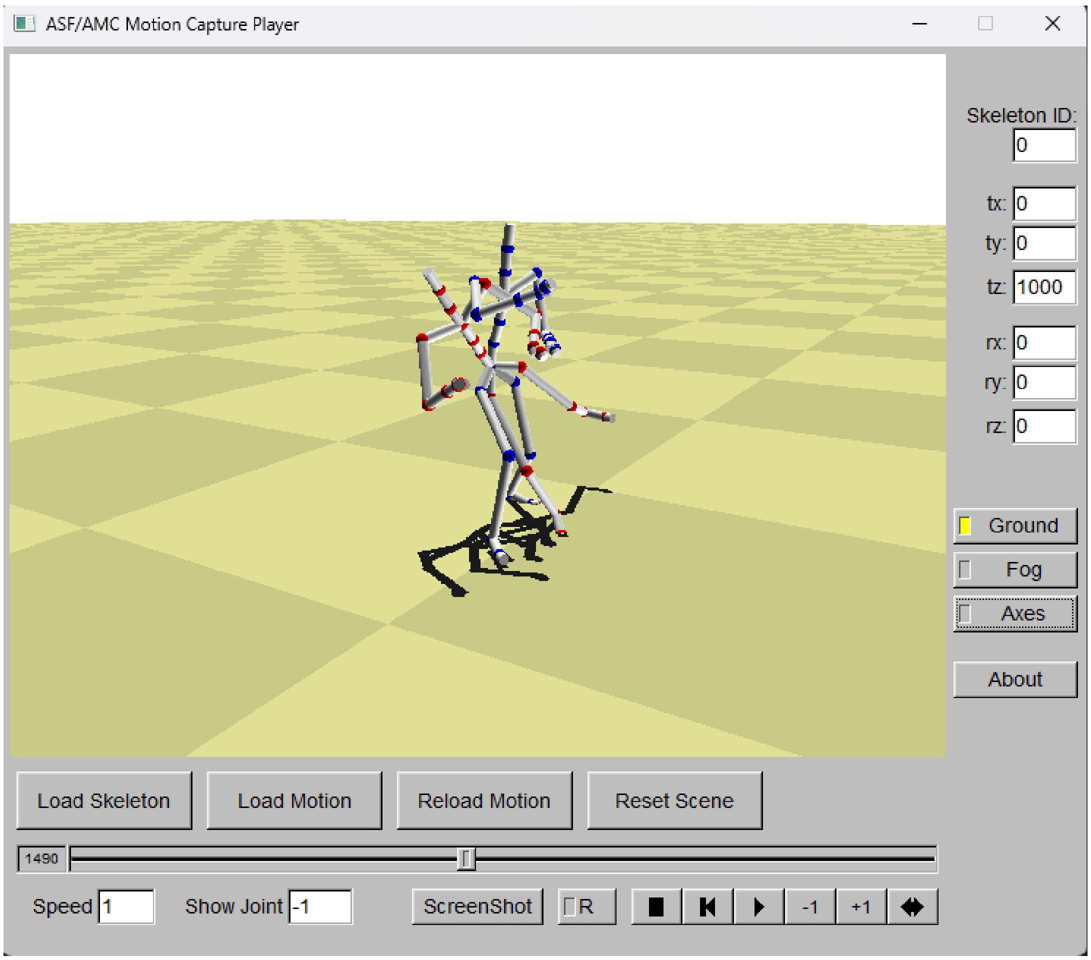
In the above graph you can compare Bezier Euler and Bezier Slerp interpolation techniques with the original motion.
The Bezier SLERP is much more similar to Original motion than Bezier Euler and is also much smoother in comparison.
Just like mentioned above here we interpolate Quaternions instead of Euler angles for Bezier SLERP so that is the reason they are much smoother,

Euler Angles Gimbal Lock Example

Test Animation: Martial Arts Animation Between Frame 1470-1510

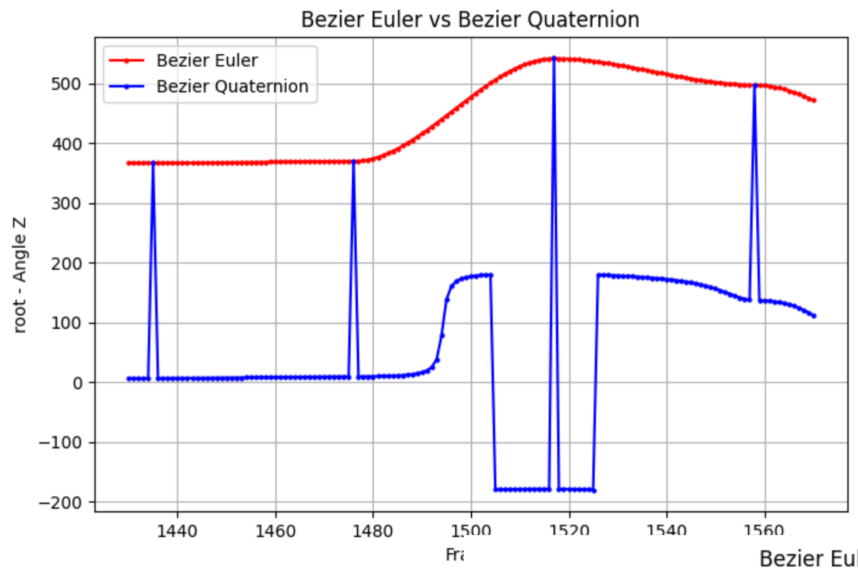
There is a gimbal lock possible between these frames when euler angles are interpolated . You can observe the difference between Bezier Euler interpolation and Bezier Quaternion interpolation.

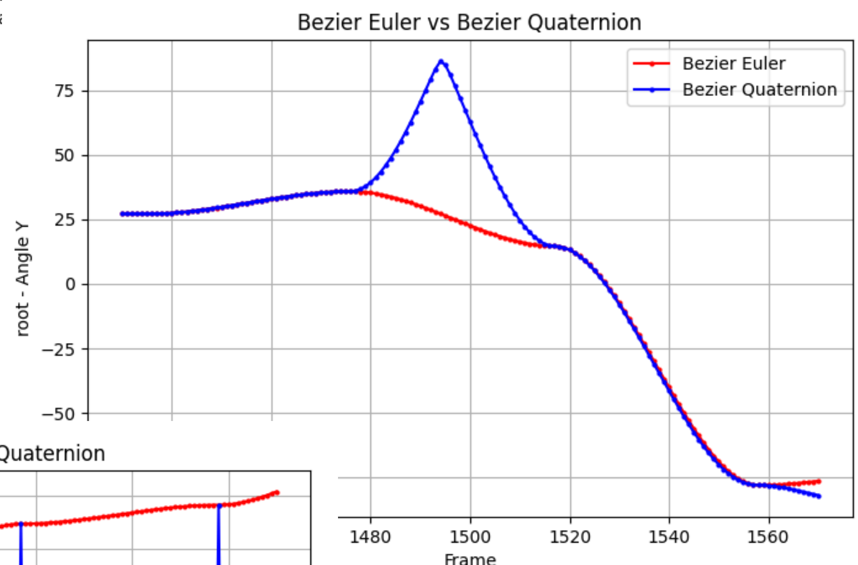
Red -> Bezier Euler Interpolation
Blue -> Bezier Quaternion Interpolation

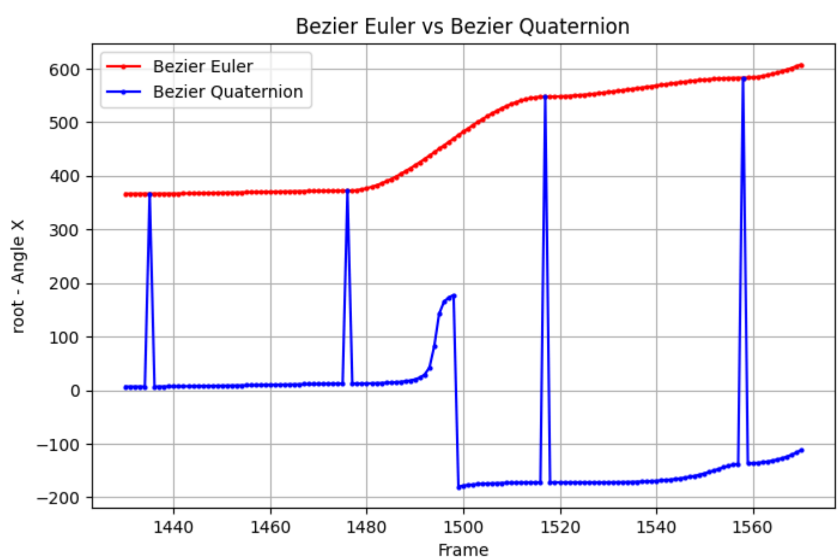


Following are the graphs of X,Y,Z euler angle when interpolating using these 2 techniques.

- You can observe while interpolating Euler angles directly all three angles change continuously during the frames based on Bezier Spline.
- But while interpolating Quaternion, you can observe each angle has different behaviour during interpolation, because Quaternion is interpolated as whole, avoiding gimbal locks.


Bezier Euler vs Bezier Quaternion
Bezier Euler
Bezier Quaternion
root - Angle Z
500
400
300
200
100
0
−100
−200
1440
1460
1480
1500
1520
1540
1560


Bezier Euler vs Bezier Quaternion
Bezier Euler
Bezier Quaternion
root - Angle Y
75
50
25
0
−25
−50
1480
1500
1520
1540
1560
Frame


Bezier Euler vs Bezier Quaternion
Bezier Euler
Bezier Quaternion
root - Angle X
600
500
400
300
200
100
0
−100
−200
1440
1460
1480
1500
1520
1540
1560
Frame