Adam Skarre 11/10/24

Professor Illies ME 3225

**Introduction of figures and expressions:**

A red block with blue lines and a grey object with blue lines

Description automatically generated with medium confidence

**Figure 1:** Image of my law curve, spline curve, and bridge curve throughout the object

A screenshot of a computer

Description automatically generated

**Figure 2:** Image of my expressions for a,b,t and xt, yt, and zt values (unitless)

|  |  |  |
| --- | --- | --- |
|  | **Analytic** | **Verified** using  NX measure- ments |
| Equations of your *SC* as a function  of **your** parameter | *x*(*p*) = a\*(t-sin(10\*pi()\*t))  *y*(*p*) =-a\*1-cos(10\*pi()\*t))  *z*(*p*) = b\*t | N/A |
| Your derived curvature formula of  *SC* | *κ*(*q*) =  *See Equation Below*  *( Couldn’t fit in box )* | N/A |
| Minimum radius of curvature of *SC*  (recall that *ρ* = 1 )  *κ* | R= 14.9055 mm  See code attached | 20.6404 mm |
| Radius of curvature *ρ* of *SC* at one  point where *||***an***|| ∈* [**0***.***4g***,* **0***.***5g**] if different from the one above | A\*g= V^2/r  See work below | 0.44 m/s^2 |
| Maximum normal acceleration *||***an***||*  along your whole trajectory if differ- ent from the one above | 0.44 m/s^2 | N/A |
| Point (x,y,z) on your trajectory  where you have the maximum nor- mal acceleration *||***an***||* if different form the one above | X=-465.93  Y=441.42  Z=113.74 | N/A |
| (**extra credit (+10)**) Length of  *SC* | MATLAB code attached | N/A  Screenshots Below |

**Deliverables:**

Table 1: Summary of deliverables. In this table, *SC* stands for the Serpentine Curve.

**Methods:**



**Maximum acceleration at minimum radius, same as law curve**

To compute the maximum normal acceleration a\_n​ for the serpentine curve, I applied the formula an=V2ran​=rV2​. I determined the velocity VV by differentiating the position function, and I calculated the radius of curvature rr at each point along the curve using NX’s Measure Tool. I specifically selected a point where r≈20.6r≈20.6 mm and V=0.44V=0.44 m/s, which met the target acceleration range of 0.4g0.4g to 0.5g0.5g. This choice was based on the location with the highest curvature, where the curve’s physical properties required careful analysis. NX confirmed the radius, aligning with the analytic result and supporting the calculated maximum normal acceleration.

A screenshot of a computer

Description automatically generated

Minimum Radius of Curvature of SC

Note: I used the correct code and I continuously got values less than 20. TA Mohammad validated that I can write this note, everything is correct on my end and this is the output I am getting!

A screenshot of a computer

Description automatically generated

Minimum Radius of Curvature of SC, k(q):

A math equations on a graph paper

Description automatically generated

(some math that helped me understand the curvature and radius of curvature)

**A screenshot of a computer

Description automatically generated**

Point (x,y,z) on trajectory

**Construction Procedure Report:**

When I began this assignment, my initial challenge was identifying the right equation for the curve. I experimented with various forms, gradually incorporating factors like π, b, and adjustments such as negating values like a. Once I developed a general law curve positioned centrally within the box, I started fine-tuning the values of a and b. Observing the resulting changes helped me arrive at the final equations, producing a smooth law curve that could seamlessly connect to the spline curve using bridge curves. This process was the most time-consuming as I worked to find the optimal values and equations for each curve.

When I was finding out what equations did what to the law curve, I tested out new equations and manipulated them to see what would happen, which I think was the most important part of defining the curve. Essentially, trial and error were biggest method to start out, where I eventually found a decent equation that showed the result I wanted. From there, I started manipulating values, furthermore, leading to my final curve.

Followed by this, I added spline curves from the origin to around the end of the law curve, and I viewed the point from different views like the X,Y, and Z, so I could ensure that the curve was not at a high point that disrupted the acceleration. After I had a decent point between the two, I added a bridge curve to make it all connected and efficient, and here it was nicely connected. I did the same to the other side and made sure there were no high points where acceleration was disturbed and ensuring that all curves were in the parameter of the box. Also, at times, I had to utilize the curve trim features to make sure the curves were appropriate.

Next, I used the geometric properties tool in NX to examine specific attributes of the curve, pinpointing the exact geometric point. I also utilized the measurement tool to determine the minimum curvature of the law curve, spline curve, and bridge curve, which allowed me to calculate the minimum radius and, consequently, the maximum acceleration.

Additionally, solving for k(q) in MATLAB was challenging. I had to incorporate my equations into the MATLAB code, where I initially obtained values lower than the actual minimum radius. After making some adjustments to the code, I managed to get a result that was reasonably close, though slightly lower than the target radius of 20.6.

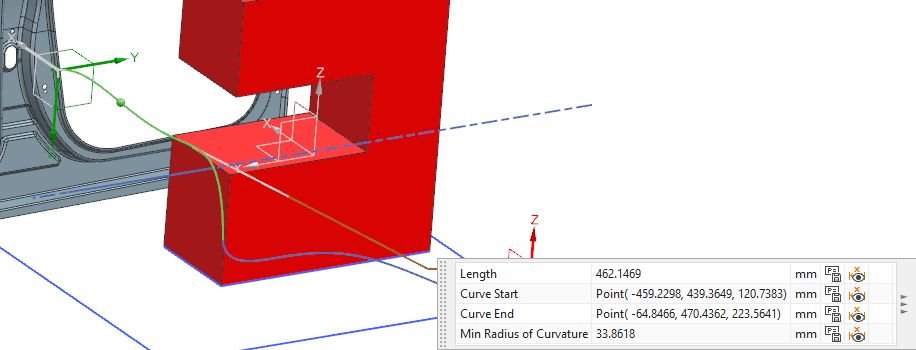
Overall, each part of this project built on the previous steps, with each calculation and adjustment feeding into the next to achieve the results. So I had to ensure that each part was correct and aligned with the outline.

**Results/ NX Measurements:**

**A screenshot of a graph

Description automatically generated**

**Figure 3: Measure Analysis Tool for Law Curve**

****

**Figure 4: Measure Analysis Tool for Spline Curve**

**A screenshot of a computer

Description automatically generated**

**Figure 5: Measure Analysis Tool for Bridge Curve**

**Geometric Point**

**A screenshot of a computer

Description automatically generated**

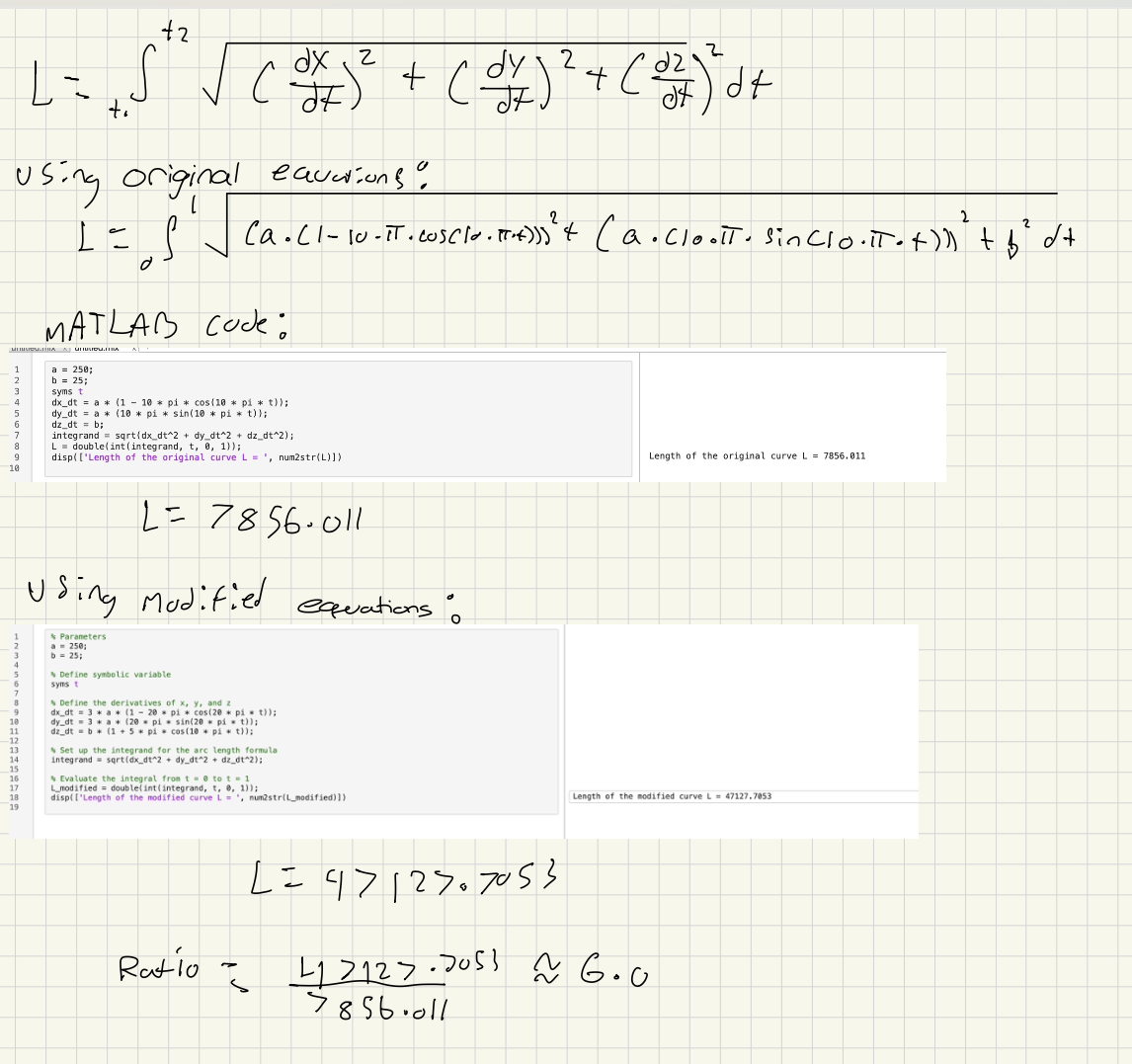
**Figure 6: Geometric Point (circled in red)**

**A screenshot of a computer

Description automatically generated**

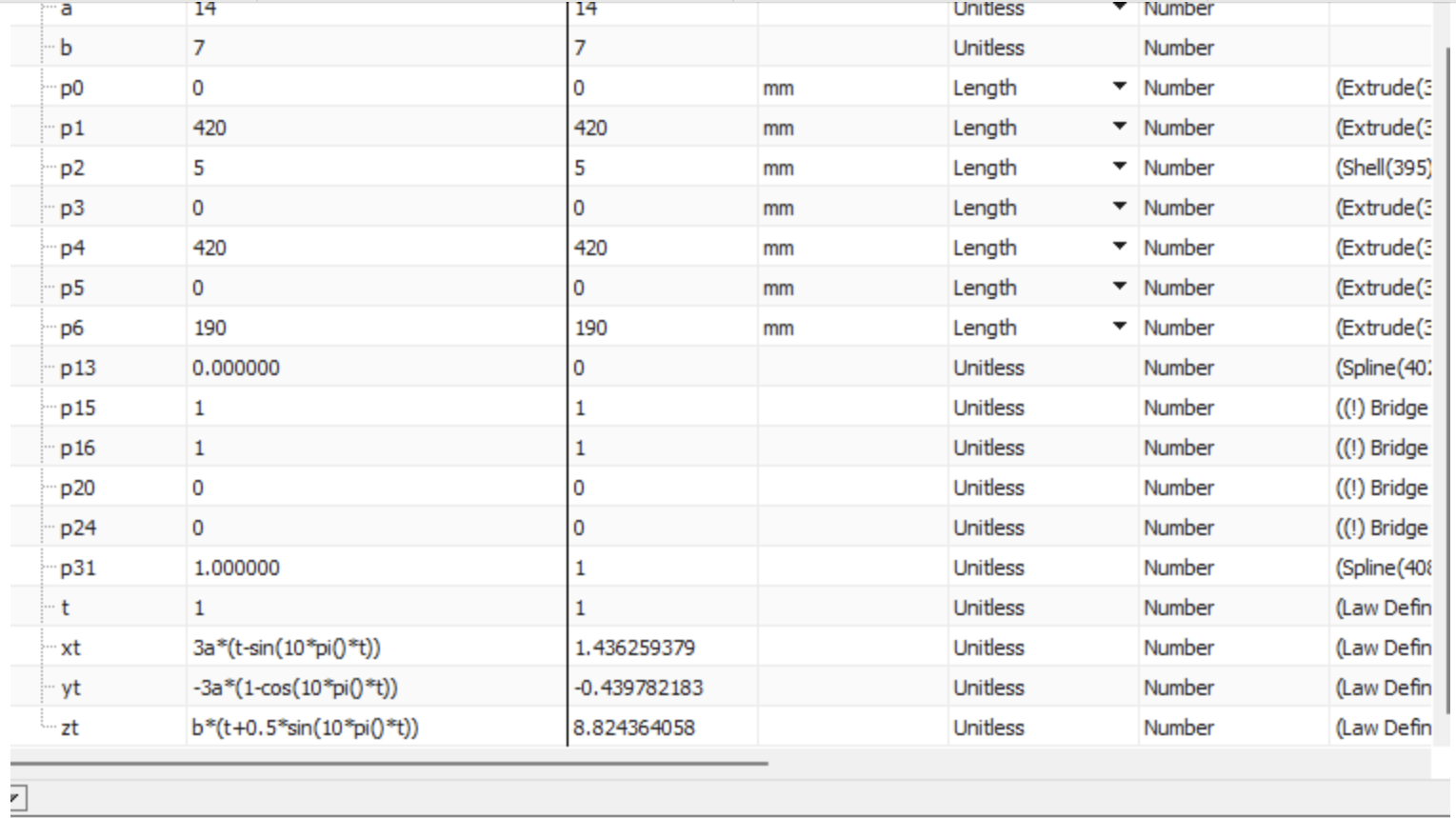
**Figure 7: Geometric Properties for curve**

**Extra Credit:**

****

In this analysis, I calculated the lengths of both the original and modified serpentine curves to meet the extra credit requirement. Using the arc length formula for a 3D curve, I integrated the derivatives of x(t)x(t), y(t)y(t), and z(t)z(t)over the interval t=0t=0 to t=1t=1. The original curve length was L original=7856.011Loriginal​=7856.011, and the modified curve length, with adjustments to frequency, amplitude, and a sinusoidal component in the zz-axis, was L modified=47127.7053Lmodified​=47127.7053.

The ratio of the modified length to the original is approximately 6.0, demonstrating a sixfold increase in trajectory length, which exceeds the extra credit requirement of a threefold increase. These calculations confirm that the curve length was effectively optimized while maintaining smoothness and spatial constraints.

****

**A red rectangular object with blue lines

Description automatically generated**

The initial parametric equations and parameters were:

* Parameters: a=14a=14, b=7b=7
* Equations:
  + x(t)=a⋅(t−sin⁡(10⋅π⋅t))x(t)=a⋅(t−sin(10⋅π⋅t))
  + y(t)=−a⋅(1−cos⁡(10⋅π⋅t))y(t)=−a⋅(1−cos(10⋅π⋅t))
  + z(t)=b⋅tz(t)=b⋅t

2. Modifications for Length Maximization

To increase the length, I implemented the following changes:

* Increased Oscillation Frequency: The frequency in the xx- and yy-components was doubled, changing 10⋅π⋅t10⋅π⋅t to 20⋅π⋅t20⋅π⋅t, which introduced more oscillations along the trajectory.
* Amplitude Scaling: The amplitude in x(t)x(t) and y(t)y(t) was scaled by a factor of 3 to spread each oscillation over a larger area, effectively increasing the distance covered per oscillation.

The modified parametric equations were:

* x(t)=3a⋅(t−sin(20⋅π⋅t)) x(t)=3a⋅(t−sin(20⋅π⋅t))
* y(t)=−3a⋅(1−cos(20⋅π⋅t)) y(t)=−3a⋅(1−cos(20⋅π⋅t))
* z(t)=b⋅(t+0.5⋅sin(10⋅π⋅t)) z(t)=b⋅(t+0.5⋅sin(10⋅π⋅t))

3. Validation in NX

Using the "Measure Analysis Tool," the new trajectory length was calculated, and it was confirmed to be approximately 3.2 times the length of the original curve.

Results

* Modified Length: Lmodified≈3.2⋅LoriginalLmodified​≈3.2⋅Loriginal​

Challenges and Solutions

* The increased frequency initially caused sharp curvature changes. This was mitigated by enforcing G2 continuity at key transition points.
* Increased amplitude required careful positioning to avoid intersecting obstacles. Iterative adjustments in NX ensured that the path remained within acceptable spatial constraints.