Development of a Self-Locking Joint for Composite Shell Structures

Written By: Nikita Dolgopolov, Cornell MAE

Faculty Advisor: Fabien Royer, Cornell Space Structures Lab

Date: 8/16/2023

Research Award Funding Source: Cornell University Engineering Learning Initiatives Program

Abstract

This paper covers the first iteration of development of a foldable carbon fiber joint that can be attached to other existing carbon fiber structures, such as NASA`s Deployable Composite Booms. The main technique used for evaluating the performance of the joint was Abaqus software simulations. The joint was optimized for maximizing the ratio of force needed to disconnect the joint to force needed to connect the joint. After that, a prototype was created and tested to verify the performance under real-life conditions.

1 Introduction

In 2020, NASA released its Deployable Composite Boom (DCB) structure [1] that is based around Collapsible Tubular Mast (CTM) geometry. The unique property of CTMs is their bi-stability – the structure is stable in expanded and collapsed states, which – for example - allows to compactly pack the composite shell by rolling it onto a spool, as in Figure 1 [1]. Additionally, optimization studies were performed on CTM geometry to emphasize bistability while maintaining high rigidity of the structure [2].

Figure 1: NASA`s Deployable Composite Boom [1]

The CTMs are commonly made from Carbon Fiber and have a uniform cross section throughout their length. The geometry is symmetric about the centerline and can be described by four parameters: two radii and two connection angles between the arcs, as in Figure 2 [2]. To ensure the continuity of the surface, the angles are generally set to equal each other. This ensures the smooth connection of the arcs. The typical scale of a CTM cross section varies from several mm to a few centimeters.

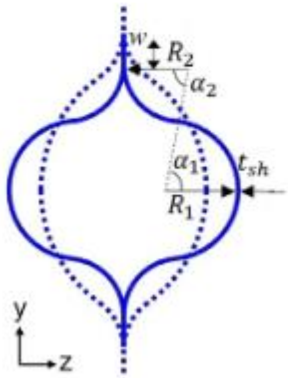

Figure 2: CTM Cross Section Design [1]

The CTMs are an extremely promising type of structure particularly due to their bi-stability property. In the expanded state, the stiffness of CTMs is very high for the given amount of material, yet the foldability of CTMs allows to package them compactly. This leads to lower weight and space requirements and the cost of delivering the structure to Earth orbit. Rolling the CTM on a motor-actuated spool allows to deploy CTM in space to construct solar panel structures, solar sails, and other collapsible and deployable structures. Expanded CTM serves as a skeleton – a holding frame for otherwise unsupported parts.

Additionally, CTMs are extremely lightweight and easy to produce since they only utilize several sheets of 2-3 layers of Carbon Fiber and are symmetric about cross sectional centerline and lengthwise. Furthermore, NASA`s DCBs, for example, experience up to 100 times less thermal distortion than analogous thin-shell metal booms [1].

As this technology progresses and new techniques are developed, it can be used to create much bigger structures with complex geometries. For example, certain modifications can be made to CTMs via laser cutting to create structures by bending CTMs and connecting multiple CTMs to each other. Since such structures can become extremely useful and advantageous over traditional metal structures, it is of great importance to develop mechanisms for connecting carbon fiber structures to each other.

Therefore, the focus of this research is the development of a carbon-fiber joint that can be attached to CTMs and other kinds of composite structures. For that reason, universality/modularity was one of the criteria for the design. Additional criteria were compactness/foldability and light weight of the design.

The joint was designed to be foldable to ensure that it can be attached to and rolled onto a spool with DCB. The material for the joint was chosen to be carbon fiber since it would simplify the attachment process to existing carbon fiber structures, and the mechanism of the joint is based on deformation of the joint under external load.

2 Materials and methods

The main approach used in this study was developing a concept for the joint and performing a FEA simulation of the design in Abaqus CAE software. Successful designs were then optimized by varying values of the geometric parameters and recording the values of significant outputs of the model. Finally, a prototype of one of the designs was created and tested in the lab. The comparison of the simulations and physical results are outlined in the "Data Processing" section.

The inspiration for the joint was taken from a snap joint that can often be found on backpacks. As in Figure 3, the joint consists of a female part and a male part. While female part can be laser cut directly out of DCB or a CTM, the male part also includes two additional "Double-section" attachments. The attachment can be described by the following variables: R1 – radius of the lower arc, R2 – radius of the top arc, R – radius of the front slope, d – distance between doubled arcs, and θ – the angle of arc connection with the horizontal. The doubled section is used to create asymmetry, which affects the forces required to connect and disconnect the joint.

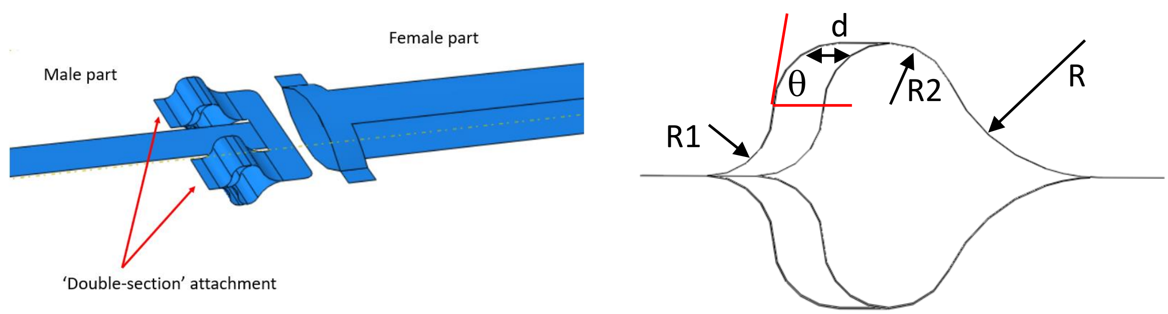

Figure 3: Joint Design

In the study, the "Double-section" attachment was tested: the joint was first pulled into an obstacle by the front edge and the force required for that was recorded. After that, the joint was pushed by the same front edge into the obstacle and the force required for that was recorded. The pulling and pushing actions simulate the connection and disconnection processes when the "Double-section" part is attached to the male part.

2.1 Abaqus CAE Simulation

The obstacle geometry is as follows: two plates placed parallel to each other with circular arcs connected to plates on each side, as in Figure 4. The arcs are connected to ensure better simulation conversion – if the arcs were to be removed, there is a high chance of simulation failure due to ambiguity in contact recognition.

The obstacle geometry was fixed in all simulations. The plate dimensions are 5mm x 20mm, and they are separated by the distance of 5mm. The arcs have radii of 3mm and an opening angle of 90 degrees; they each extend by 20mm along z-axis into the page.

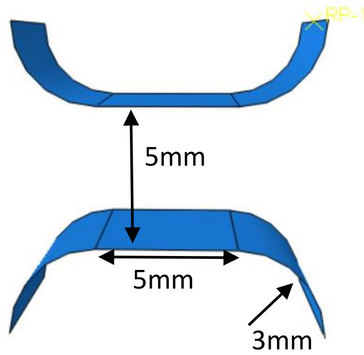

Figure 4: Obstacle geometry

The process for creating and running Abaqus simulation includes defining the geometry of the "Double-section" part and the obstacle geometry, creating materials and composite sections, assigning sections to parts, positioning the objects in the assembly, defining interactions, and assigning boundary conditions.

2.1.1 Part geometry

The "Double-section" part was parametrized around d and θ values – d was varied from 0.5mm to 2.5mm with a step of 0.5mm and θ was varied from 4π/16 to 12π/16 with a step of π/16. The effects of parameters on geometry are showcased in Figure 5. This resulted in a total of 45 different geometries for the optimization study. First, values for R, R1, and R2 were fixed to 5mm, 2mm, and 2mm respectively. The height of the joint was calculated and R, R1, and R2 were scaled by

$(4.0\times R) \div ((R1 + R2 - R1 \times \cos(\theta) - R2 \times \cos(\theta)))$

to make the total height of "Double-section" part equal to 8mm. Flat edges at the front and the back of the part were each set to 2 mm, and their value does not affect the simulation results. The whole part is 10mm wide along z-axis into the page.

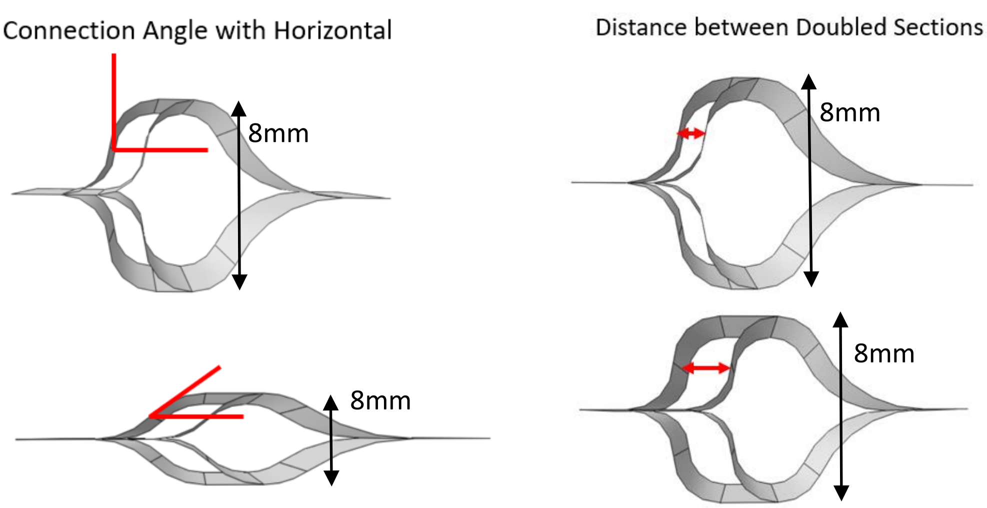

Figure 5: a) Effect of connection angle (θ) on geometry, b) Effect of distance (d) on geometry

2.1.2 Material and Section definition

The material was defined to be Unidirectional Carbon Fiber with properties in accordance with Andrew J. Lee`s study on CTM structures [2] and are outlined in Table 1. Values of Young`s modulus and shear modulus in different directions E1, E2, G12, G13, and G23 are in MPa, while Poisson`s ratio Nu12 is unitless. The axes for moduli are defined in Figure 7.

E1	E2	Nu12	G12	G13	G23
174400	8390	0.259	6400	6400	6400

Table 1: Material Properties

Sections of 3 thicknesses were created: single, double, and quadruple sections. The exact assignments to regions are reflected in Figure 6. This is done to simulate multiple layers of carbon fiber laying on top of each other as the part is created.

Single section includes two carbon fiber layers: 0-degree and 90-degree layer such that the layers are perpendicular to each other while lying in the same plane. This significantly boosts the performance of the composite section by creating a mesh that stacks E1 and E2 moduli. The thickness of a single carbon fiber layer is 0.05mm. Double section and quadruple section include single sections stacked on top of each other two and four times, respectively.

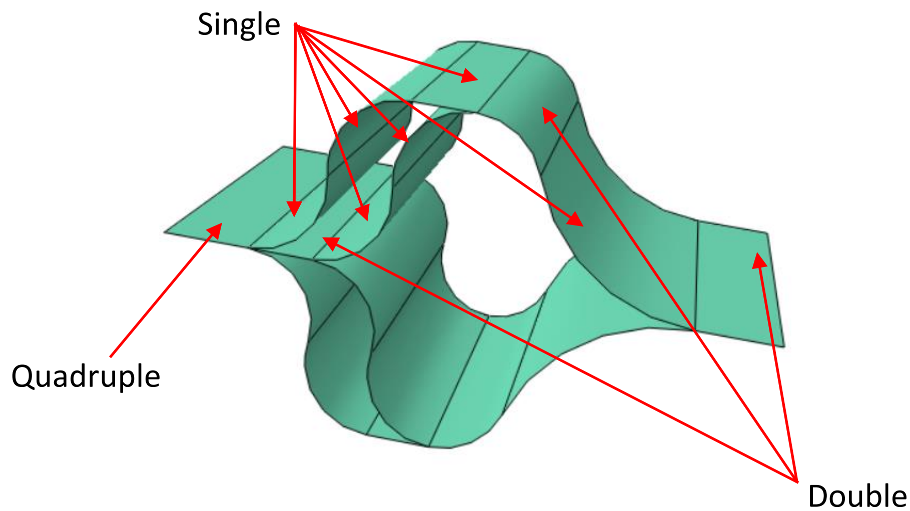

Figure 6: Section assignments

To assign sections, the following orientations were used. Direction #1 is parallel to the 0-degree layer of sections, direction #2 – 90-degree layer, and direction n defines shell normal and ensures that first layer (0 degree) of sections is on the inside of the section. The obstacle part was assigned a single section of unidirectional carbon fiber.

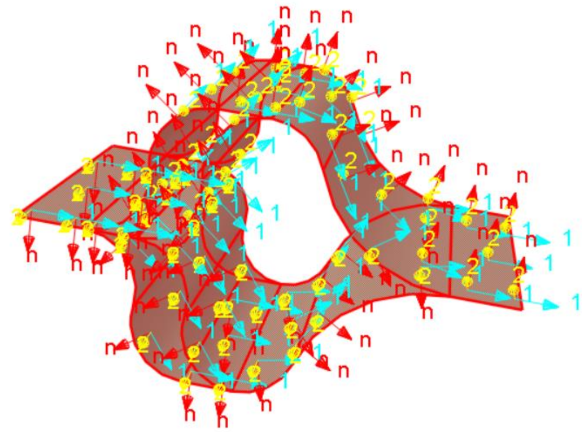

Figure 7: Surface orientations

Certain sections were offset from their respective surfaces to ensure smooth connections of layers, as illustrated in Figure 8.

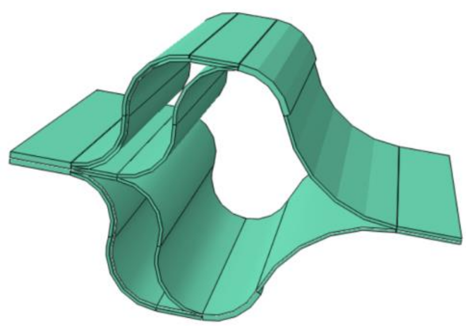

Figure 8: Surface offsets

The "Double-section" part was meshed with an approximate seed size of 0.3mm and rectangular shape.

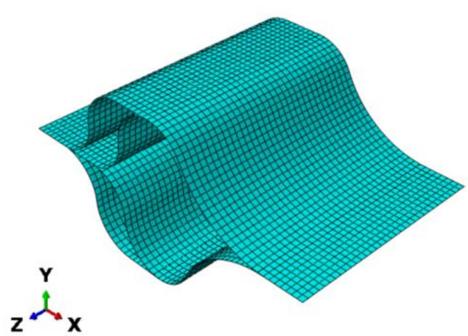

Figure 9: Mesh

2.1.3 Assembly and Analysis Steps

Two separate assemblies were created to analyze connection and disconnection of the joint.

2.1.3.1 Connection setup

The “Double-section” part was positioned in front of an obstacle in a way that the part centers are aligned along x-axis – this ensures symmetry, as in Figure 10. The obstacle was assigned a Rigid constraint and a 0-displacement boundary condition to prevent it from deformation and movement.

The frontmost edge of the “Double-section” part was assigned a rigid constraint and a reference point RP-3. The boundary condition of +60mm x-axis, 0mm y-axis, and 0mm z-axis displacements was assigned to RP-3. The backmost edge was assigned a 0mm y-axis, and 0mm z-axis displacement boundary condition to ensure that the rear edge only moves along x-direction for symmetry purpose – there is a chance of non-uniform mesh that would make deformation asymmetric.

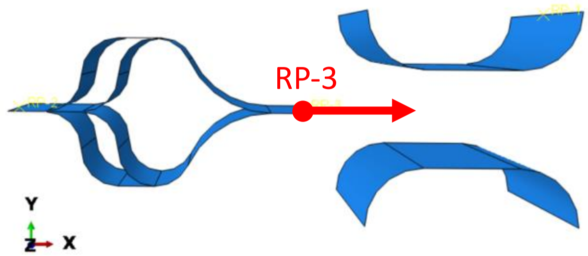

Figure 10: Connection simulation assembly

2.1.3.2 Disconnection setup

The “Double-section” part was positioned in the back of an obstacle in a way that the part centers are aligned along x-axis – this ensures symmetry, as in Figure 11. The obstacle was assigned a Rigid constraint and a 0-displacement boundary condition to prevent it from deformation and movement.

The frontmost edge the “Double-section” part was assigned a rigid constrained and a reference point RP-3. The boundary condition of -60mm x-axis, 0mm y-axis, and 0mm z-axis displacements was assigned to RP-3. The rear edge was assigned a 0mm y-axis, and 0mm z-axis displacement boundary condition to ensure that the edge only moves along x-direction for symmetry purpose – there is a chance of non-uniform mesh that would make deformation asymmetric.

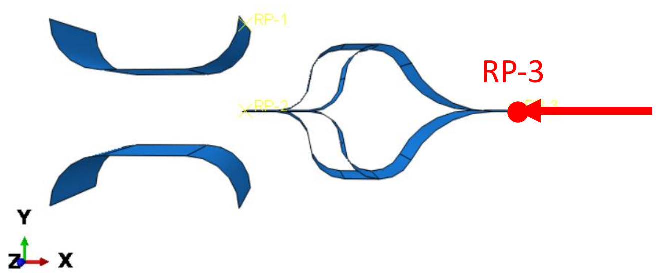

Figure 11: Disconnection simulation assembly

2.1.4 Abaqus data extraction

In both cases, the contact was defined between surfaces to prevent them from intersecting each other and for causing deformation to the part when it touches the obstacle. While the boundary condition is active and part travels through the obstacle, the force is recorded on the RP-3. As a result, tables with force values along the displacement are created.

2.2 Manufacturing

For manufacturing of the "Double-section" part, a silicone mold was first created. After that, fiberglass was used for prototyping, and the "Double-section" part was created via a custom manufacturing process that is described later in section 2.2.2 and in the publication by Arthur Scholothauer and Sergio Pellegrino [3].

2.2.1 Mold manufacturing

The mold was designed with the intention of creating 3 scales of the "Double-section" part – 1:1, 2:1, and 4:1. Eventually, only the 4:1 scale part was created due to issues with manufacturing smaller parts. Silicone is very viscous, so air bubbles are often formed when several-mm scale parts are made. Figure 12 shows the dimensions of the shell to be manufactured at the 4:1 scale and the mold.

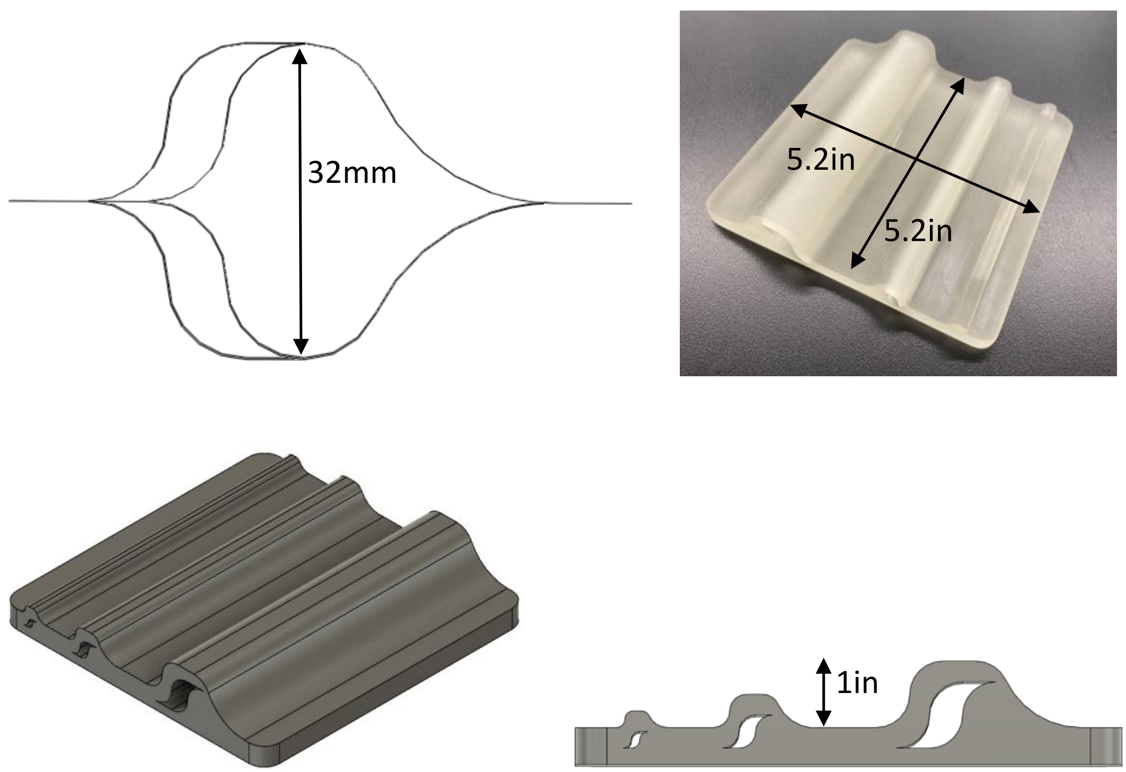

Figure 12: a) Shell dimensions, b) Printed mold, c) & d) Printed mold design

To manufacture the mold, the mold negative was first printed on the formlabs Form 3 resin 3D printer (Figure b). The outer dimensions of the print and the molds are determined by the metal cages owned by the lab. There are 2 middle pieces with inner dimensions of 5.2 x 5.2 x 2 in; additionally, there are 2 plates for top and bottom of the metal box – all 4 metal pieces can be secured together with bolts, as in Figure 13.

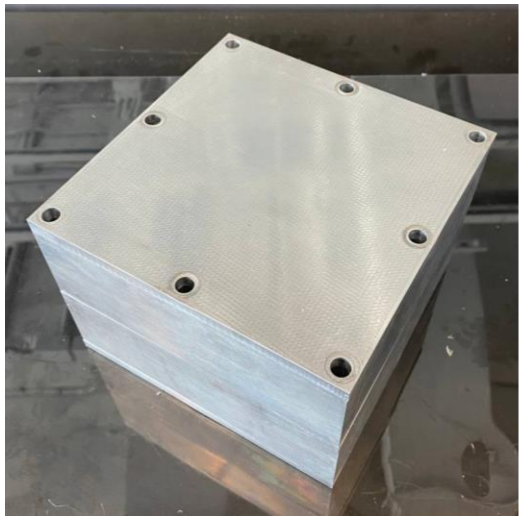

Figure 13: Assembled metal cage

To match the height of such middle piece, the first two silicone mold pieces were made with the height of 1 in each. After them, the rest of silicone pieces were molded by pouring liquid silicone between other pieces, as in Figure 14. The silicone was made by mixing Mold Max XLS compounds in a 10:1 proportion, removing air bubbles from the mixture via a vacuum chamber, and letting the liquid stand in the mold for 24 hours. To ensure that silicone does not stick to other parts of mold, a layer of release agent EASE RELEASE 200 was applied to the molds.

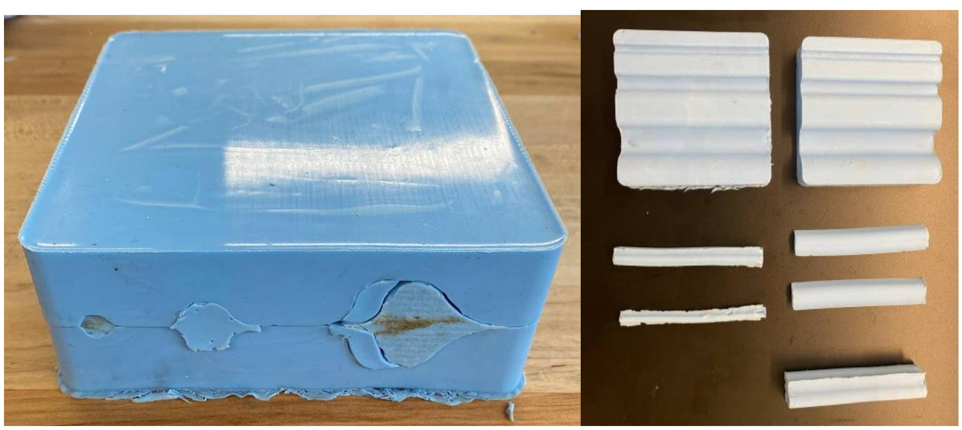

Figure 14: a) Silicone mold assembled, b) Silicone mold pieces

2.2.2 Shell manufacturing

The shell prototype for testing was made from glass fiber. To form a single section, two layers: 0-degree and 90-degree were used with the 90-degree layer on the inside of the shell. The layers were laid between silicone parts of the mold, and the width of the shell was made to be 50mm.

Next step in shell manufacturing is curing the glass fiber in the oven. For that, silicone mold with fiberglass is tightly secured inside the metal cage with the bolts. When silicone is heated in the oven, it tries to expand. However, full volume of the metal cage is occupied, so silicone expansion under high temperatures causes high pressure inside the mold. Therefore, it is important that silicone occupies the full volume of the metal cage to prevent the silicone mold from changing shape under expansion of silicone into voids and that the bolts are secured properly.

The curing cycle used includes four steps: temperature rise, high temperature hold, temperature drop, and low temperature hold. After that, the temperature automatically drops to 0.

The full cycle consists of the following steps, as in Figure 15:

1) Heat to 160°C over 60 minutes
2) Stay at 160°C for 33 minutes
3) Cool to 122°C over 30 minutes
4) Stay at 122°C over 5 hours

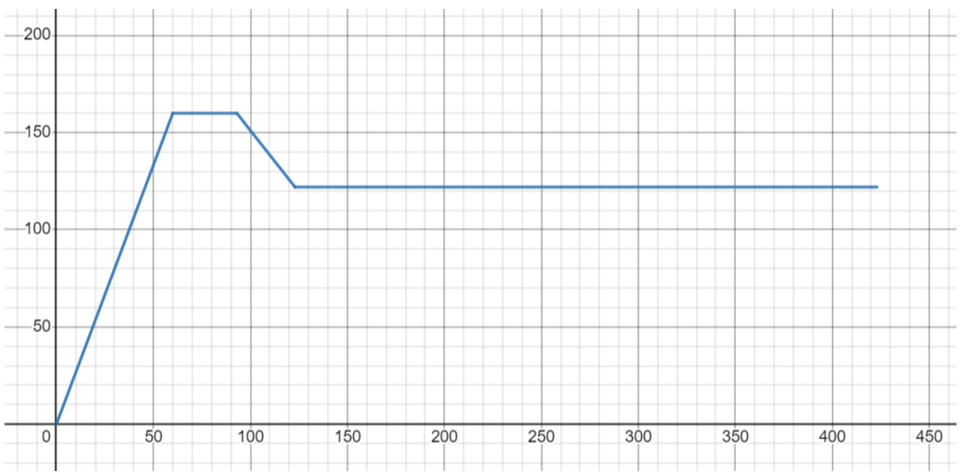

Figure 15: Curing cycle (Temperature(°C) vs Time(minutes))

On Figures 16 and 17, silicone mold and the shell can be seen after the curing was finished. Crumbs of silicone on the edges are the result of silicone`s expansion into the gaps between pieces of the metal cage during curing. The shell was cured true to the intended size: 32mm in height. The small pieces of glue at the edges of the shell are hardened glue that was used to fix some parts of the mold – they can be cut off with scissors. That glue also gave the shell a yellowish color. The blue marks on the shell are leftovers of pen ink that was used to mark glass fiber sheets when they were cut from the roll.

Figure 16: Silicone mold post curing

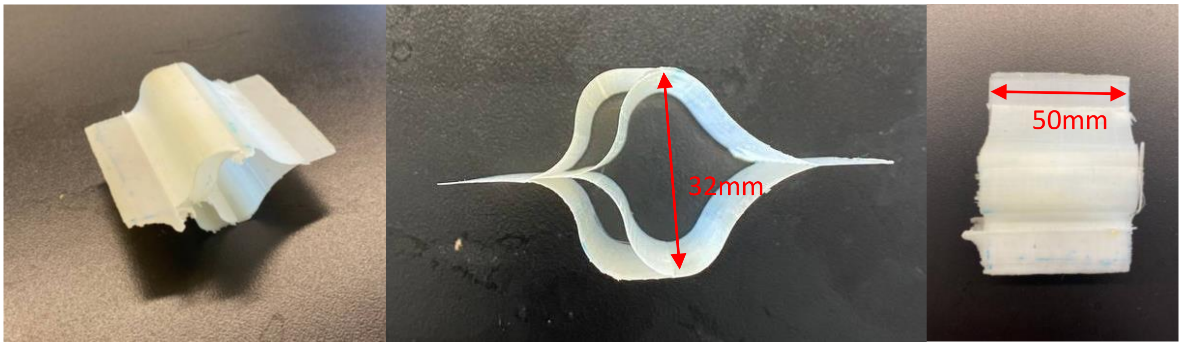

Figure 17: Glass fiber shell

2.3 Experimental testing

The testing was done on the 4:1 scaled version of the “Double-section” part using Instron 68TM-5 machine and a 100N capacity electric force gauge.

2.3.1 Experimental setup

For experimental testing, a hold for the shell was printed with a slot to glue the shell to it, as in Figure 18. The obstacle was also 3D printed with PLA. The complete dimensions of the obstacle are depicted in Figure b. It was designed to align the center of two plates (the obstacle) with the center of flat leg at the top of the piece that is clamped into the testing machine at the top. Each plate is 5 x 20 x 40mm. The simulations of the connection and disconnection were each completed once. The speed of relative motion of parts was set to 2mm/minute. The shell was cut to a width of 10mm.

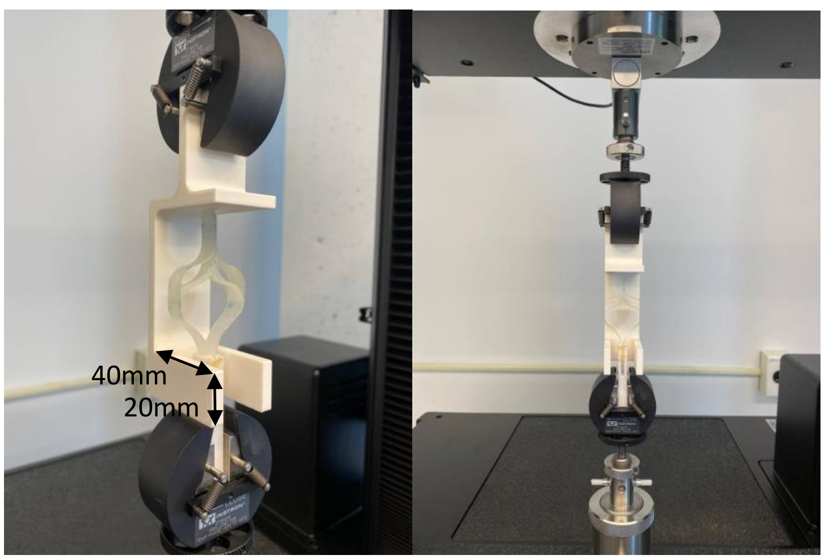

Figure 18: Experimental setup for connection step

2.3.2 Experimental data extraction

The data was recorded by the testing machine and the maximum force was extracted.

3 Data processing

MATLAB code was written to process data extracted from Abaqus. It can be found in Appendix 2. Text files with force vs displacement table were loaded into the code, and max() function was used to retrieve the highest force values from the arrays with force values. The simulations were also filtered by their convergence: displacement of the shell was tracked and compared to a "cutoff" value. If it was smaller than the "cutoff" value, the simulation failed too early – the force ratio value for the combination of parameters d and θ was set to NaN.

To obtain values maps against two parameters, meshgrid() and griddata() MATLAB functions were used. This allows to create continuous graphs with levels that highlight areas of lower and higher force and force ratio values.

The data from experimental testing was processed by finding average force values over all trials.

4 Results

4.1 Abaqus

The Abaqus simulations over distance and angle of connection yielded the following unfiltered maps, as in Figure 19.

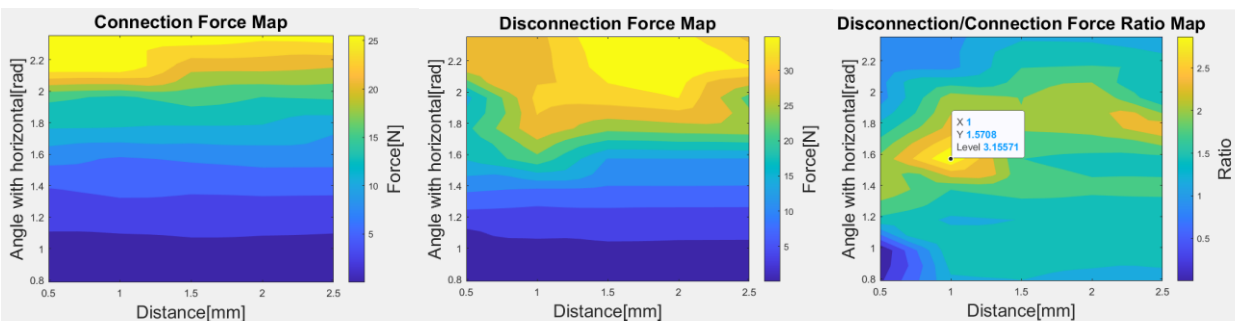

Figure 19: Simulation results

The connection force values vary from 0.8117 N to 28.1055 N, and disconnection force values lie between 1.0941 and 38.2861. The maximum force ratio was found at d=1mm, and θ = 90 degrees. There is a clear variation in forces due to angle of connection: the higher the angle is, the higher is the force. Variation of forces with distance values is not so evident: in most cases distance does not significantly affect the force. The force ratio map, however, shows a clear peak at d=1mm and θ = 90degrees accompanied by an area of relatively high force ratio values at θ values between 75 degrees and 120 degrees. There is an additional small peak at d =2.5mm, θ = 105 degrees.

Unfortunately, setting cutoff value to approximately 0.5, which would imply shell passing halfway through the obstacle, sets absolute majority of results to NaN – the simulations fail too early. Therefore, the results are inconclusive.

4.2 Experimental testing

For connection step, the following Force vs Displacement graph was obtained, as in Figure 20. The force values are discrete due to automatic rounding of the sensor measurements to account for errors. The maximum force value reached is 0.5N – very low in comparison with Abaqus simulation values.

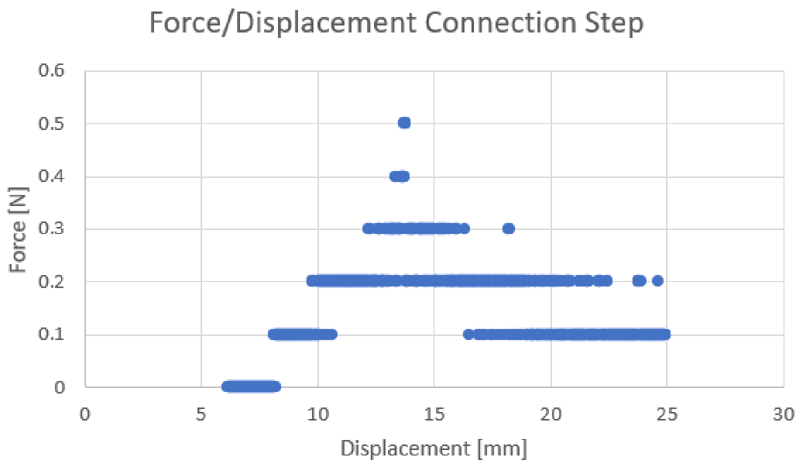

Figure 20: Force vs Displacement plot

On the disconnection step (Figure 21), the shell was prone to folding and locking completely, which means that the joint would not disconnect unless the material fails. This is not the expected result; however, such behavior can be helpful for structures that should not be disconnected easily and that can stay in the expanded state for an extended period of time.

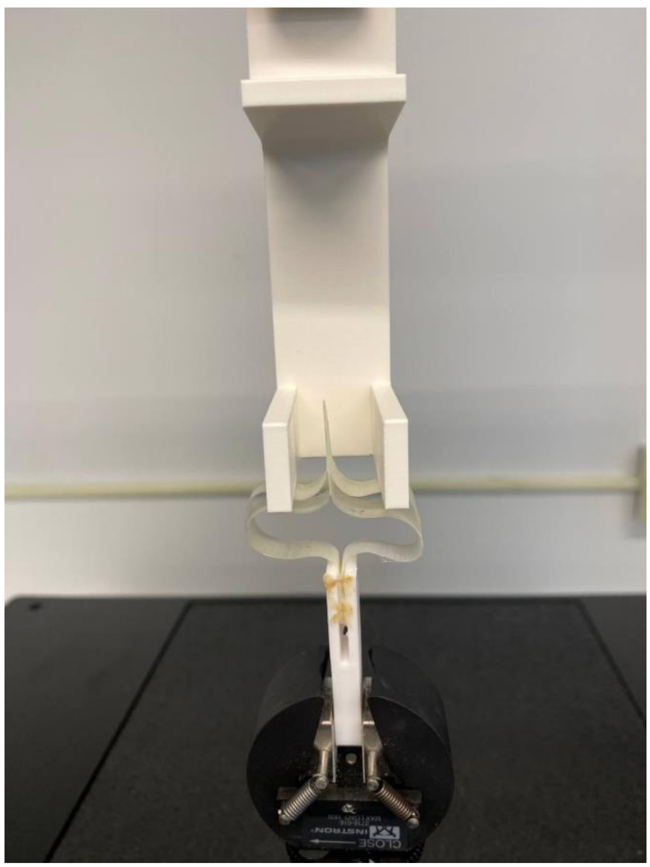

Figure 21: Disconnection step deformation

5 Discussion

An important fact to consider during the analysis of simulation results is that obstacle arcs can significantly alter the results of simulations. The arcs on the obstacle were used for better simulation convergence since it is challenging for Abaqus to process edge-surface contacts. As a result, there is almost no possibility of the joint locking completely, as was in the experimental disconnection step. Varying the height of the obstacle with respect to the joint also plays a big role: a very small height may not allow the joint to be connected at all, while a big height may not lead to sufficiently high disconnection force.

An important consideration for the analysis of experimental results is friction between plastic of the obstacle and the surface of the shell in physical models. The friction could have been the main cause for the deformation observed in the disconnection step: friction resists the movement of surfaces of the shell inward and fixes them at the contact points. Essentially, the shell sticks to the obstacle.

Another possible consideration for experimental testing involves the fact that the obstacle is 3D printed out of PLA plastic. Thin walls of it are prone to deformation, so the tests do not necessarily depict the actual deformation. However, the effect would be rather insignificant since the forces involved are too small to cause any visible deformation. Furthermore, the material for experimental testing was chosen to be glass fiber and not carbon fiber seen in the simulations. Their properties differ, though the behavior is very similar. Glass fiber is more flexible than carbon fiber of the same thickness.

The reason for small forces observed in the disconnection step experimental testing is that the shell size to shell thickness ratio is 4 times higher in the test than in simulations. Because of this, the surfaces of the shell do not have to bend or deform locally as much as they would for a smaller shell – there are higher curvatures involved. Lastly, glass fiber is much more flexible than carbon fiber, so lower force was required to deform it to the same state. Therefore, having joint made from carbon fiber would increase forces involved and result in a more stable joint design.

6 Conclusion

This paper has presented a design for a joint made from carbon fiber that can be easily attached to other composite structures. The joint utilizes asymmetric deformation as a source of producing varying force on connection and disconnection of two structures. Optimization study around maximizing the disconnection force over connection force ratio was conducted to find an optimal configuration and an experimental testing was completed to evaluate the applicability of simulation results in real life scenarios. Though methodology was not completely successful, certain useful conclusions were made and a foundation was laid for future design iterations. The initial design, the code for parametric generation and simulation of these geometries, and MATLAB data processing code were created, and they will be extremely useful once the solutions for challenges are developed.

Therefore, future steps would involve improving the current generating code to avoid early simulation failures, evaluating the joint performance over the full connection and disconnection steps, and conducting experimental testing that replicates Abaqus simulations as closely as possible. This would involve either scaling the geometry in simulations to the 4:1 scale or developing a reliable technique for manufacturing small-scale shell joints, eliminating friction in the experimental testing, and creating the joint from carbon fiber instead of glass fiber.

Once the joint is finalized, it would be helpful to connect it to other carbon fiber structures and verify the performance in these more realistic cases. Zero gravity could also be simulated in these scenarios to replicate in-space operation.

7 References

[1] Vitug, E. (2020, August 27). *Deployable composite booms (DCB)*. NASA. https://www.nasa.gov/directorates/spacetech/game_changing_development/projects/dcb

[2] Lee, A. J., & Fernandez, J. M. (2019, June 19). Inducing bistability in Collapsible Tubular Mast booms with thin-ply composite shells. https://www.sciencedirect.com/science/article/pii/S0263822319307706

[3] Schlothauer, A., & Pellegrino, S. (2018, May). Flexible Silicone Molds For the Rapid Manufacturing of Ultra-Thin Fiber Reinforced Structures. https://www.researchgate.net/profile/Arthur-Schlothauer/publication/325781102_Flexible_Silicone_Molds_For_the_Rapid_Manufacturing_of_Ultra-Thin_Fiber_Reinforced_Structures/links/617a69493c987366c3f4dcf9/Flexible-Silicone-Molds-For-the-Rapid-Manufacturing-of-Ultra-Thin-Fiber-Reinforced-Structures.pdf

8 Appendix

1) Abaqus Simulation Script Connection and Disconnection Steps

The Python code is too long to be pasted into the Appendix, so it was included in the attached files.

2) MATLAB data processing code

```
clear all
close all
clc

%Cell geometry (everything in rad and mm)
%constants
Dlist = [0.5,1.0,1.5,2.0,2.5];
Thetalist = [4*pi/16, 5*pi/16 , 6*pi/16, 7*pi/16, 8*pi/16,9*pi/16, 10*pi/16,
11*pi/16,  12*pi/16];
   dsize = size(Dlist);
   tsize = size(Thetalist);

   %Initialize structures
   aspect=[];
   simDataF=zeros(dsize(2),tsize(2),2);
   RatioData=zeros(dsize(2),tsize(2));
   dispData=[];

   R1_val = 2.0;
   R2_val = 2.0;
   R_val = 5.0;
```

```
%extract data

for II=1:dsize(2)
    for JJ=1:tsize(2)

        d1_val=Dlist(II);
        theta_val=Thetalist(JJ);
        baseheight = 5.0/8.0*(R1_val+R2_val-R1_val*cos(theta_val)-
R2_val*cos(theta_val));
        my = baseheight/2.0;

        FolderName='D:/Abaqus License/Models/SimResults';
        FolderNameBack='D:/Abaqus License/Models/SimResults2';

jobname=[[[[[[[['doubleEdgeJoint_d1=',num2str(d1_val*10)],'_R='],num2str(10*R_val
)],'_R1='],num2str(10*R1_val)],'_theta='],num2str(theta_val*16/pi)],'data.txt'];

jobnameBack=[[[[[[[['doubleEdgeJoint_d1=',num2str(d1_val*10)],'_R='],num2str(10*R
_val)],'_R1='],num2str(10*R1_val)],'_theta='],num2str(theta_val*16/pi)],'_backdat
a.txt'];

        R_int = 4.0*R_val/((R1_val+R2_val-R1_val*cos(theta_val)-
R2_val*cos(theta_val)));
        R1_int = 4.0*R1_val/((R1_val+R2_val-R1_val*cos(theta_val)-
R2_val*cos(theta_val)));
        R2_int = 4.0*R2_val/((R1_val+R2_val-R1_val*cos(theta_val)-
R2_val*cos(theta_val)));

        cos_alpha = (R_int-R1_int+(R2_int+R1_int)*cos(theta_val))/(R_int+R2_int);
        sin_alpha = sqrt(1.0- cos_alpha^2);
        length = (R1_int+R2_int)*sin(theta_val)+d1_val+sin_alpha*(R_int+R2_int);
        cutoff_val = (30.0+length)/60.0;
        cutoffback_val = (25.0+length)/60.0;

        simText = readlines(jobname);
        simText = simText(5:end-5);
        ssize = size(simText);

        LastLine = strtrim(simText(end));
        timeVal = extractBefore(LastLine, ' ');
        setNAN = 0;

        valuesArray = [0];
        cutoff = cutoff_val;
        cutoffback = cutoffback_val;

        if str2double(timeVal) > cutoff


            for linenum = 1:ssize(1)
                TextLine = simText(linenum);
```

```
                TextLine = strtrim(TextLine);
                TextLine = extractAfter(TextLine, ' ');
                TextLine = strtrim(TextLine);
                valuesArray(linenum) = str2double(TextLine);
            end
        else
            setNAN = 1;
            simDataF(II,JJ, 1) = NaN;
        end

        simTextBack = readlines(jobnameBack);
        simTextBack = simTextBack(5:end-5);
        ssizeBack = size(simTextBack);

        LastLineBack = strtrim(simTextBack(end));
        timeValBack = extractBefore(LastLineBack, ' ');
        setNANBack = 0;

        valuesArrayBack = [0];

        if str2double(timeValBack) > cutoffback

            for linenumBack = 1:ssizeBack(1)
                TextLineBack = simTextBack(linenumBack);
                TextLineBack = strtrim(TextLineBack);
                TextLineBack = extractAfter(TextLineBack, ' ');
                TextLineBack = strtrim(TextLineBack);
                valuesArrayBack(linenumBack) = str2double(TextLineBack);
            end
        else
            setNANBack = 1;
            simDataF(II,JJ, 2) = NaN;
        end


        %filtering

        if (setNAN == 1) || (setNANBack == 1)
            RatioData(II,JJ) = NaN;
        else
            maxVal = max(valuesArray);
            minVal = min((valuesArrayBack));
            %pick out the data by extracting the max value
            simDataF(II,JJ, 1)=(maxVal);
            simDataF(II,JJ, 2)=abs(minVal);


            RatioData(II,JJ)=simDataF(II,JJ, 2)/simDataF(II,JJ, 1);
        end
    end
end
```

```matlab
dataIn = simDataF(:,:, 1);
dataOut = simDataF(:,:, 2);

%interpolate data
[xq,yq]=meshgrid(0.5:0.1:2.5, pi/4:pi/32:24*pi/32); %create a grid
fqIn=griddata(Dlist,Thetalist,dataIn', xq,yq); %interpolate my data to grid
fqOut=griddata(Dlist,Thetalist,dataOut', xq,yq); %interpolate my data to grid
fqRatio=griddata(Dlist,Thetalist,RatioData', xq,yq); %interpolate my data to grid

%plot

figure
[C1,h1]=contourf(xq,yq,fqIn, 10);
set(h1, 'EdgeColor', 'none');
title('Connection Force Map', FontSize=18)
c=colorbar;
h1.LineWidth = 2;
box on
xlabel('Distance[mm]','FontSize',18);
ylabel('Angle with horizontal[rad]','FontSize',18);
c.Label.String = 'Force[N]';
c.Label.FontSize = 18;

figure
[C2,h2]=contourf(xq,yq,fqOut, 10);
set(h2, 'EdgeColor', 'none');
title('Disconnection Force Map', FontSize=18)
c=colorbar;
h2.LineWidth = 2;
box on
xlabel('Distance[mm]','FontSize',18);
ylabel('Angle with horizontal[rad]','FontSize',18);
c.Label.String = 'Force[N]';
c.Label.FontSize = 18;

figure
[C3,h3]=contourf(xq,yq,fqRatio, 10);
set(h3, 'EdgeColor', 'none');
title('Disconnection/Connection Force Ratio Map', FontSize=18)
c=colorbar;
h3.LineWidth = 2;
box on
xlabel('Distance[mm]','FontSize',18);
ylabel('Angle with horizontal[rad]','FontSize',18);
c.Label.String = 'Ratio';
c.Label.FontSize = 18;
```