

```
function [FEM,rebound,theta,cable_post,strap_post,new_F] = build_testing_links_straps_new(
(FEM,tor, straps,total_load,tori_theta,thetal,cable_rad,C)
%% GENERAL
center(1) = (C(1,2) + C(2,2))/2; % Contact between tori z
center(2) = (C(1,1) + C(2,1))/2; % Contact between tori r
test_z = center(1)+2.84;
test_rad = center(2)-17.6;
cable_z = test_z;

% Instantiate
r = [tor.r]'; % Radius of tori

% Properties for link elements
E = 10e6;
R = 1;
A = pi*R^2;
Izz = pi*(1/16)^4/4;
Iyy = pi*R^4/4;
J = pi*R^4/2;

% Properties for cable elements
E2 = 10e6;
R2 = .1;
A2 = pi*R2^2;
Izz2 = pi*R2^4/4;
Iyy2 = pi*R2^4/4;
J2 = pi*R2^4/2;

% Properties for loading strap boundary Elements
E3 = 20;
R3 = .1;
A3 = pi*R3^2;
Izz3 = pi*R3^4/4;
Iyy3 = pi*R3^4/4;
J3 = pi*R3^4/2;

% Get theta Locations
theta = thetal;

for i = 1:size(theta)
    for j = 1:size(tori_theta)
        if round(theta(i),3) == round(tori_theta(j),3)
            tori_nodes(i) = j;
        end
    end
end

tori_nodes = tori_nodes';

%% NODES
```

```

for i = 1:size(tori_nodes)
    tan1 = tangentPoint([test_rad, test_z],...
        [sqrt(FEM.MODEL.nodes(tori_nodes(i),1)^2+...
            FEM.MODEL.nodes(tori_nodes(i),2)^2),...
            FEM.MODEL.nodes(tori_nodes(i),3)],r(1),0);

    tan2 = tangentPoint([test_rad, test_z],...
        [sqrt(FEM.MODEL.nodes(length(tori_theta)+tori_nodes(i),1)^2+...
            FEM.MODEL.nodes(length(tori_theta)+tori_nodes(i),2)^2),...
            FEM.MODEL.nodes(length(tori_theta)+tori_nodes(i),3)],r(2),1);

    x1 = abs(tan1(1,1))*cos(theta(i));
    y1 = abs(tan1(1,1))*sin(theta(i));
    z1 = tan1(1,2);
    nodes1(i,:) = [x1 y1 z1];
    x2 = abs(tan2(1,1))*cos(theta(i));
    y2 = abs(tan2(1,1))*sin(theta(i));
    z2 = tan2(1,2);
    nodes2(i,:) = [x2 y2 z2];

    % Loading Strap Element Node Locations
    x3 = test_rad*cos(theta(i));
    y3 = test_rad*sin(theta(i));
    z3 = test_z;
    nodes3(i,:) = [x3 y3 z3];
    point1 = [sqrt(FEM.MODEL.nodes(tori_nodes(i),1)^2+...
        FEM.MODEL.nodes(tori_nodes(i),2)^2),...
        FEM.MODEL.nodes(tori_nodes(i),3)];

    point2 = [sqrt(FEM.MODEL.nodes(length(tori_theta)+tori_nodes(i),1)^2+...
        FEM.MODEL.nodes(length(tori_theta)+tori_nodes(i),2)^2), ...
        FEM.MODEL.nodes(length(tori_theta)+tori_nodes(i),3)];

    tangent = circle_tan([point1(1), point1(2)], [point2(1), ...
        point2(2)]], [r(1) r(2)], 1); % Find tangent line between tori

    x4 = abs(tangent(1,1))*cos(theta(i));
    y4 = abs(tangent(1,1))*sin(theta(i));
    z4 = tangent(1,2);
    nodes4(i,:) = [x4, y4, z4];
    x5 = abs(tangent(2,1))*cos(theta(i));
    y5 = abs(tangent(2,1))*sin(theta(i));
    z5 = tangent(2,2);
    nodes5(i,:) = [x5, y5, z5];
    x6 = cable_rad*cos(theta(i))+4*cos(theta(i)+(pi/2));
    y6 = cable_rad*sin(theta(i))+4*sin(theta(i)+(pi/2));
    z6 = cable_z;
    nodes6(i,:) = [x6 y6 z6];
    x7 = cable_rad*cos(theta(i))+4*cos(theta(i)+((3*pi)/2));
    y7 = cable_rad*sin(theta(i))+4*sin(theta(i)+((3*pi)/2));
    z7 = cable_z;

```

```
nodes7(i,:) = [x7 y7 z7];
x8 = cable_rad*cos(theta(i));
y8 = cable_rad*sin(theta(i));
z8 = cable_z;
nodes8(i,:) = [x8 y8 z8];
xyint = FEM.MODEL.nodes(tori_nodes(1),1)+r(1)*cos(1.8);
zint = FEM.MODEL.nodes(tori_nodes(1),3)+r(1)*sin(1.8);
x10 = xyint*cos(theta(i));
y10 = xyint*sin(theta(i));
z10 = zint;
nodes10(i,:) = [x10 y10 z10];
xyint = FEM.MODEL.nodes(tori_nodes(1),1)+r(1)*cos(1.4);
zint = FEM.MODEL.nodes(tori_nodes(1),3)+r(1)*sin(1.4);
x11 = xyint*cos(theta(i));
y11 = xyint*sin(theta(i));
z11 = zint;
nodes11(i,:) = [x11 y11 z11];
xyint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),1)+r(1)*cos(1-0.62*1);
zint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),3)+r(1)*sin(1-0.62*1);
x12 = xyint*cos(theta(i));
y12 = xyint*sin(theta(i));
z12 = zint;
nodes12(i,:) = [x12 y12 z12];
xyint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),1)+r(1)*cos(1-0.62*2);
zint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),3)+r(1)*sin(1-0.62*2);
x13 = xyint*cos(theta(i));
y13 = xyint*sin(theta(i));
z13 = zint;
nodes13(i,:) = [x13 y13 z13];
xyint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),1)+r(1)*cos(1-0.62*3);
zint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),3)+r(1)*sin(1-0.62*3);
x14 = xyint*cos(theta(i));
y14 = xyint*sin(theta(i));
z14 = zint;
nodes14(i,:) = [x14 y14 z14];
xyint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),1)+r(1)*cos(1-0.62*4);
zint = FEM.MODEL.nodes(length(tori_theta)+tori_nodes(1),3)+r(1)*sin(1-0.62*4);
x15 = xyint*cos(theta(i));
y15 = xyint*sin(theta(i));
z15 = zint;
nodes15(i,:) = [x15 y15 z15];
%% U displacement vector
x9 = -0.5*cos(theta(i));
y9 = -0.5*sin(theta(i));
z9 = 0;
nodes9(i,:) = [x9,y9,z9];

end

nodes = [nodes1; nodes2; nodes3; nodes4; nodes5; nodes6; nodes7; nodes8;...
nodes10; nodes11; nodes12; nodes13; nodes14; nodes15];
```

```

%% ORIENTATION
orientation = zeros(size(nodes,1),3);
orientation(:,3) = 1e6;

%% CONNECTIVITIES

% Links
connect_l1 = [tori_nodes(1:size(theta,1)) size(FEM.MODEL.nodes,1) + (1:(size(theta,1)))'];

connect_l2 = [size(tori_theta,1) + tori_nodes(1:size(theta,1)) ...
    size(FEM.MODEL.nodes,1) + size(theta,1) + (1:(size(theta,1)))'];

connect_l3 = [tori_nodes(1:size(theta,1)) size(FEM.MODEL.nodes,1) + ...
    size(theta,1)*3 + (1:(size(theta,1)))'];

connect_l4 = [size(tori_theta,1) + tori_nodes(1:size(theta,1)) ...
    size(FEM.MODEL.nodes,1) + size(theta,1)*4 + (1:(size(theta,1)))'];

connect_l = [connect_l1; connect_l2; connect_l3; connect_l4];

con_link = [connect_l 2*ones(size(connect_l,1),1)];

% Straps
connect_s1 = [size(FEM.MODEL.nodes,1) + (1:(size(theta,1)))' ...
    size(FEM.MODEL.nodes,1) + size(theta,1)*2 + (1:(size(theta,1)))'];

connect_s2 = [size(FEM.MODEL.nodes,1) + size(theta,1) + ...
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1) + size(theta,1)*2 + ...
    (1:(size(theta,1)))'];

connect_s3 = [size(FEM.MODEL.nodes,1) + size(theta,1)*3 + ...
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1) + size(theta,1)*4 + ...
    (1:(size(theta,1)))'];

connect_s = [connect_s1; connect_s2; connect_s3];

strap_link = [connect_s 1*ones(size(connect_s,1),1)];

% Bench
connect_b1 = [size(FEM.MODEL.nodes,1) + size(theta,1)*7 + (1:(size(theta,1)))' size(FEM. ✓
MODEL.nodes,1) + size(theta,1)*5 + (1:(size(theta,1)))'];
connect_b2 = [size(FEM.MODEL.nodes,1) + size(theta,1)*7 + (1:(size(theta,1)))' size(FEM. ✓
MODEL.nodes,1) + size(theta,1)*6 + (1:(size(theta,1)))'];
connect_b = [connect_b1; connect_b2];
bound_link = [connect_b 2*ones(size(connect_b,1),1)];

% Cable
connect_c = [size(FEM.MODEL.nodes,1) + size(theta,1)*2 + (1:(size(theta,1)))' size(FEM. ✓
MODEL.nodes,1) + size(theta,1)*7 + (1:(size(theta,1)))'];
cable_link = [connect_c 1*ones(size(connect_c,1),1)];

```

## % New Links

```
connect_n11 = [tori_nodes(1:size(theta,1)) size(FEM.MODEL.nodes,1)+...  
    size(theta,1)*8 + (1:(size(theta,1)))'];  
  
connect_n12 = [tori_nodes(1:size(theta,1)) size(FEM.MODEL.nodes,1)+...  
    size(theta,1)*9 + (1:(size(theta,1)))'];  
  
connect_n13 = [size(tori_theta,1)+tori_nodes(1:size(theta,1)) ...  
    size(FEM.MODEL.nodes,1)+ size(theta,1)*10 + (1:(size(theta,1)))'];  
  
connect_n14 = [size(tori_theta,1)+tori_nodes(1:size(theta,1)) ...  
    size(FEM.MODEL.nodes,1)+ size(theta,1)*11 + (1:(size(theta,1)))'];  
  
connect_n15 = [size(tori_theta,1)+tori_nodes(1:size(theta,1)) ...  
    size(FEM.MODEL.nodes,1)+ size(theta,1)*12 + (1:(size(theta,1)))'];  
  
connect_n16 = [size(tori_theta,1)+tori_nodes(1:size(theta,1)) ...  
    size(FEM.MODEL.nodes,1)+ size(theta,1)*13 + (1:(size(theta,1)))'];  
  
connect_n1 = [connect_n11; connect_n12; connect_n13; connect_n14;...  
    connect_n15; connect_n16;];  
  
con_newlink = [connect_n1 2*ones(size(connect_n1,1),1)];
```

## % New Straps

```
connect_ns1 = [size(FEM.MODEL.nodes,1) + (1:(size(theta,1)))' ...  
    size(FEM.MODEL.nodes,1)+size(theta,1)*8 + (1:(size(theta,1)))'];  
  
connect_ns2 = [size(FEM.MODEL.nodes,1)+size(theta,1)*8 + ...  
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1)+size(theta,1)*9 +...  
    (1:(size(theta,1)))'];  
  
connect_ns3 = [size(FEM.MODEL.nodes,1)+size(theta,1)*9 + ...  
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1) + size(theta,1)*3 +...  
    (1:(size(theta,1)))'];  
  
connect_ns4 = [size(FEM.MODEL.nodes,1) + size(theta,1)*4 +...  
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1)+size(theta,1)*10 + ...  
    (1:(size(theta,1)))'];  
  
connect_ns5 = [size(FEM.MODEL.nodes,1)+size(theta,1)*10 + ...  
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1)+size(theta,1)*11 + ...  
    (1:(size(theta,1)))'];  
  
connect_ns6 = [size(FEM.MODEL.nodes,1)+size(theta,1)*11 +...  
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1)+size(theta,1)*12 +...  
    (1:(size(theta,1)))'];  
  
connect_ns7 = [size(FEM.MODEL.nodes,1)+size(theta,1)*12 +...  
    (1:(size(theta,1)))' size(FEM.MODEL.nodes,1)+size(theta,1)*13 +...
```

```

(1:(size(theta,1)))'];

connect_ns8 = [size(FEM.MODEL.nodes,1)+size(theta,1)*13 + ...
(1:(size(theta,1)))' size(FEM.MODEL.nodes,1) + size(theta,1) + ...
(1:(size(theta,1)))'];

connect_ns = [connect_ns1; connect_ns2; connect_ns3; connect_ns4;
connect_ns5; connect_ns6; connect_ns7; connect_ns8];

con_newstrap = [connect_ns 1*ones(size(connect_ns,1),1)];

cable_post = [size(FEM.MODEL.connect,1)+size(con_link,1)+...
size(strap_link,1)+size(bound_link,1)+1, size(FEM.MODEL.connect,1)+...
size(con_link,1)+size(strap_link,1)+size(bound_link,1)+...
size(cable_link,1)];

strap_post(1) = size(FEM.MODEL.connect,1)+size(con_link,1)+1;
% strap_post(2) = size(FEM.MODEL.connect,1)+size(con_link,1)+size(connect_s1,1)+1;
% strap_post(3) = size(FEM.MODEL.connect,1)+size(con_link,1)+size(connect_s1,1)+size(
(connect_s2,1)+1;
% strap_post(4) = size(FEM.MODEL.connect,1)+size(con_link,1)+size(bound_link,1)+size(
(cable_link,1)+size(con_newlink,1)+1;
% strap_post(5) = size(FEM.MODEL.connect,1)+size(con_link,1)+size(bound_link,1)+size(
(cable_link,1)+size(con_newlink,1)+size(connect_ns1:connect_ns5,1)+1;

%% ELEMENTS
% Preallocate element structure
EL = [];
EL(size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link,1)).el = [];
EL(size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link,1)).el_in = [];
EL(size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link,1)).el_in0.
break = [];
EL(size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link,1)).el_in0.mat
= [];
EL(size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link,1)).el_in0.geom
= [];
EL = EL';

% Links
for i = 1:size(con_link,1)
    EL(i).el_in0 = instantiate_EL; % Instantiate all element variables

    % Define element functions
    EL(i).el = 'e12'; % Linear, corotational beam

    % Special element input
    EL(i).el_in0.break = 0;
    EL(i).el_in0.mat = [E .3]; % [E nu]
    EL(i).el_in0.geom = [A Izz Iyy 0 J]; % [A Izz Iyy ky J]
end

```

```

% Straps
strap_type = 3;
for i = size(con_link,1)+1:size(con_link,1)+size(strap_link,1)
    EL(i).el_in0 = instantiate_EL; % Instantiate all element variables

    % Define element functions
    EL(i).el = 'el4';

    % Corotational beam with axial lookup table
    EL(i).el_in0.break = 0;
    EL(i).el_in0.mat = straps(strap_type).mat; % [E nu]
    EL(i).el_in0.geom = straps(strap_type).geom; % [A Izz Iyy ky J]
    EL(i).el_in0.axial = straps(strap_type).axial;
    EL(i).el_in0.axial_k = straps(strap_type).axial_k;
    EL(i).el_in0.eps0 = straps(strap_type).eps0;
end

% Bench
for j = size(con_link,1)+size(strap_link,1)+1:size(con_link,1)+size(strap_link,1)+size(
bound_link,1)
    EL(j).el_in0 = instantiate_EL; % Instantiate all element variables

    % Define element functions
    EL(j).el = 'el2'; % Linear, corotational beam

    % Special element input
    EL(j).el_in0.break = 0;
    EL(j).el_in0.mat = [E3 .3]; % [E nu]
    EL(j).el_in0.geom = [A3 Izz3 Iyy3 0 J3]; % [A Izz Iyy ky J]

    % Element prestrain
    EL(j).el_in0.eps0 = 0;
end

% Cable
for j = size(con_link,1)+size(strap_link,1)+size(bound_link,1)+1:size(con_link,1)+size(
strap_link,1)+size(bound_link,1)+size(cable_link)
    EL(j).el_in0 = instantiate_EL; % Instantiate all element variables

    % Define element functions
    EL(j).el = 'el2'; % Linear, corotational beam

    % Special element input
    EL(j).el_in0.break = 0;
    EL(j).el_in0.mat = [E2 .3]; % [E nu]
    EL(j).el_in0.geom = [A2 Izz2 Iyy2 0 J2]; % [A Izz Iyy ky J]

    % Element prestrain
    EL(j).el_in0.eps0 = 0;
end

```

```

% New Links
for i = size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link,1)+1:size(
(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link)+size(con_newlink)
    EL(i).el_in0 = instantiate_EL; % Instantiate all element variables

    % Define element functions
    EL(i).el = 'el2'; % Linear, corotational beam

    % Special element input
    EL(i).el_in0.break = 0;
    EL(i).el_in0.mat = [E .3]; % [E nu]
    EL(i).el_in0.geom = [A Izz Iyy 0 J]; % [A Izz Iyy ky J]
end

% New Straps
for i = size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link)+size(
(con_newlink)+1:size(con_link,1)+size(strap_link,1)+size(bound_link,1)+size(cable_link)
+size(con_newlink)+size(con_newstrap)
    EL(i).el_in0 = instantiate_EL; % Instantiate all element variables

    % Define element functions
    EL(i).el = 'el4';

    % Corotational beam with axial lookup table
    EL(i).el_in0.break = 0;
    EL(i).el_in0.mat = straps(strap_type).mat; % [E nu]
    EL(i).el_in0.geom = straps(strap_type).geom; % [A Izz Iyy ky J]
    EL(i).el_in0.axial = straps(strap_type).axial;
    EL(i).el_in0.axial_k = straps(strap_type).axial_k;
    EL(i).el_in0.eps0 = straps(strap_type).eps0;
end

%% MODEL
FEM.EL = [FEM.EL; EL];
FEM.MODEL.nodes = [FEM.MODEL.nodes; nodes(:,1:3)];
FEM.MODEL.connect = [FEM.MODEL.connect; [con_link;strap_link;bound_link;cable_link;
con_newlink;con_newstrap]];
FEM.MODEL.orientation = zeros(size(FEM.EL,1),1);
FEM.MODEL.orientation(:,3) = 1e6;

%% Boundary
b1 = zeros(size(theta,1)*5,6);
b2 = ones(size(theta,1)*2,6);
b3 = zeros(size(theta,1)*7,6);
b = [b1; b2; b3];
b = b';
bound = b(:);
rebound(1,:) = [(size(FEM.MODEL.B,1) + size(theta,1)*7*6) (size(FEM.MODEL.B,1) + size(
(theta,1)*8*6)];
FEM.MODEL.B = [FEM.MODEL.B; bound];

```



```
%% Loading
f1 = zeros(size(theta,1)*7,6);
load('loads.mat');
new_load = (max(lc)-70)';
f10 = (-new_load).*[cos(theta) sin(theta)];
f2 = (-total_load).*[cos(theta) sin(theta)];
f5 = zeros(size(theta,1),4);
f6 = [f2 f5];
f11 = [f10 f5];
f7 = zeros(size(theta,1)*6,6);
f = [f1;f6;f7];
f = f';
F = f(:);
f12 = [f1;f11;f7];
f13 = f12';
f14 = f13(:);
new_F = [FEM.MODEL.F; f14];
FEM.MODEL.F = [FEM.MODEL.F; F];

%% U Vector Displacements
u1 = zeros(size(theta,1)*7,6);
u2 = zeros(size(theta,1),3);
u3 = [nodes9, u2];
u = [u1;u3];
u4 = zeros(size(theta,1)*6,6);
u = [u;u4];
u = u';
U = u(:);

FEM.MODEL.U_pt = [FEM.MODEL.U_pt; U];

%% Set forces
FEM.MODEL.F_pre = FEM.MODEL.F*0;
FEM.MODEL.F_pt = [FEM.MODEL.F*0 FEM.MODEL.F];
```