```
%{
mwave - A water wave and wave energy converter computation package
Copyright (C) 2014  Cameron McNatt

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.

Contributors:
    C. McNatt
%}
classdef ConstraintMatComp

    methods (Static)
        function [P] = HingedBodies(bods, hins, varargin)
            % Inputs:
            %   bods = N x 3 matrix of {x, y, z} body coordinates,
 where N
            %       is the number of bodies
            %   hins = (N-1) x 3 matrix of {x, y, z} hinge
 coordinates. The
            %       y-coordinate is not really necessary as it's a
 hinge
            %       about an axis parallel to the y-axis
            %
            % Optional Inputs:
            %   'Origin', org = org is a 1 x 3 vector indicating the
 origin
            %       of the composite body. If the optional 'Origin'
            %       argument is not provided, the default is is the
 body
            %       coordinates of body 1
            %
            % Returns:
            %   P = the velocity transformation matrix
            %       (not PT, i.e. the transpose)

            [opts, args] = checkOptions({{'Origin', 1}}, varargin);

            if (opts(1))
                org = args{1};
            else
                org = bods(1,:);
```

```
           end

           Nbod = size(bods, 1);
           Nhin = size(hins, 1);

           if (Nhin ~= (Nbod - 1))
               error(['The number of hinges must be one less than the
'...
               'number of bodies']);
           end

           if (Nbod > 1)
               if ((size(bods, 2) ~= 3) || (size(hins, 2) ~= 3))
                   error(['The body coordinates and hinge coordinates
'...
                       'must have x,y,z locations']);
               end
           end

           PT = zeros(6*Nbod, 6 + Nhin);

           sR = zeros(Nbod, 3);
           sL = zeros(Nbod, 3);

           for n = 1:Nbod
               if (n < Nbod)
                   sR(n,:) = hins(n,:) - bods(n,:);
               end

               if (n > 1)
                   sL(n,:) = hins(n-1,:) - bods(n,:);
               end
           end

           sO = bods(1,:) - org;

           for n = 1:Nbod
               PTn = zeros(6, 6 + Nhin);

               % Identity matrices
               PTn(1:3, 1:3) = eye(3);
               PTn(4:6, 4:6) = eye(3);

               % cross-product matrix
               svect = -sO;
               for m = 2:n
                   svect = svect - sR(m-1,:) + sL(m,:);
               end
               Sx = ConstraintMatComp.skewMat(svect);
               PTn(1:3,4:6) = Sx;

               for o = 2:n
                   svect = [-sL(n,3), 0, sL(n,1)];
```

```
                    for m = n:-1:(o+1)
                        svect = svect + [-sL(m-1,3), 0, sL(m-1,1)] ...
                            + [sR(m-1,3), 0, -sR(m-1,1)];
                    end

                    PTn(1:3,o+5) = svect';
                end

                % row of 1's in pitch for flex modes
                PTn(5,7:(5+n)) = ones(1,n-1);

                istart = (n - 1)*6 + 1;
                PT(istart:(istart + 5), :) = PTn;
            end

            P = PT.';
        end
    end

    methods (Static, Access = private)

        function [M] = skewMat(v)
            M = zeros(3, 3);
            x = v(1);
            y = v(2);
            z = v(3);

            M(1,2) = -z;
            M(1,3) = y;

            M(2,1) = z;
            M(2,3) = -x;

            M(3,1) = -y;
            M(3,2) = x;
        end
    end
end
```

```
ans = 

  ConstraintMatComp with no properties.
```

Published with MATLAB® R2015b