```
function [SP,W] = sigmaPoints(x, P, type)
% SIGMAPOINTS computes sigma points, either using unscented transform
 or
% using cubature.
%
%Input:
%   x           [n x 1] Prior mean
%   P           [n x n] Prior covariance
%
%Output:
%   SP          [n x 2n+1] UKF, [n x 2n] CKF. Matrix with sigma points
%   W           [1 x 2n+1] UKF, [1 x 2n] UKF. Vector with sigma point
 weights
%
          P_d2=sqrtm(P);
          n=length(x);
    switch type
        case 'UKF'
            w0=1-n/3;
            wi=(1-w0)/(2*n);
            wp=sqrt(n/(1-w0));

            W=ones(1,2*n+1)*wi;
            W(1)=w0;

            for i=0:1:n
                if i==0
                    SP(:,i+1)=x;
                else
                    SP(:,i+1)=x+wp*P_d2(:,i);
                    SP(:,i+1+n)=x-wp*P_d2(:,i);
                end
            end

        case 'CKF'
            wi=1/(2*n);
            wp=sqrt(n);

            W=ones(1,2*n)*wi;

            for i=1:1:n
                SP(:,i)=x+wp*P_d2(:,i);
                SP(:,i+n)=x-wp*P_d2(:,i);
            end
        otherwise
            error('Incorrect type of sigma point')
    end
end
```

Published with MATLAB® R2020a