# Function: find_sf

Computes the Singularity functions for T,V,M and M/I for the bottom shaft. Uses helper functions to mimic the unit step and slope singularity functions (ie, x-a). Parameters

- F - Force Applied to the shaft
- A - location of the pulley
- B - location of Bearing F
- C - location of Bearing G
- x - position vector, represents multiple locations on the shaft

Returns - vectors representing the singularity functions

# Code

```
function [T,V,M,M_over_I] = find_sf(F,A,B,C,x)
 % Find Spring Forces, with A,B and C in cm
 % F is the force applied to the shaft
 [Fk1, Fk2] = spring_forces(F,A,B,C);

 % Given
 L = 40/100; % m
 A = A/100; B = B/100; C = C/100; % Converting from cm to m
 bearing_width = 14/1000; % m
 d1 = 20; % mm
 d2 = 30; % mm
 % Torque calcs
 n = 1000; % rpm
 w = 1000 * pi/30; % rad/s
 H = 1000; % W

 % Mass moment of Interia
 I1 = pi * (d1/1000)^4/64;
 I2 = pi * (d2/1000)^4/64;

 % Creating position vector
 x = x/100; % from cm to m

 % Creating singularity functions for shear
 y1 = sf_step(x,A); y2 = sf_step(x,B); y3 = sf_step(x,C);

 % Creating singularity functions for moment
 y4 = sf_slope(x,A); y5 = sf_slope(x,B); y6 = sf_slope(x,C);

 % Creating a singularity function for the first step due to the change of I
 y7 = sf_step(x,B+bearing_width/2); % step for first shoulder

 % Creating a singularity function for torque
 y8 = sf_step(x,A); y9 = sf_step(x,L);
 % Complete Singularity
 V = F*y1 + Fk1*y2 + Fk2*y3;
 M = F*y4 + Fk1*y5 + Fk2*y6;
 T = H/w*y8 - H/w*y9;

 % Creating slopes for singularity function
 slope1 = max(M)/(I1 * (B-A));
 slope2 = slope1 + max(M)/(I2 * (C-B));
 slope3 = slope2 - slope1;
 step1 = max(M)/I1 - max(M)/I2;

 % Showing that the second step is small and has no significance
 % M_bearing2 = M(floor((C-bearing_width/2)*1000/L))
 % Step2 very. very small, just consider it as insignificant
 % step2 = M_bearing2/I1 - M_bearing2/I2

 M_over_I = slope1*y4 - slope2 *y5 - step1*y7 + slope3*y6;

end

% Helper Functions
function y = sf_step(x,a)
 y = (x>a);
end

function y = sf_slope(x,a)
 y =(x-a).*(x > a);
end
```

```
Error using find_sf (line 18)
Not enough input arguments.
```

*Published with MATLAB® R2014b*