PA #1: Calculate DoF using user inputs

Table of Contents

Grubler's Formula

Grubler's Formula can be used to calculate the number of degrees of freedom of a robot (DoF) by substracting the number of independent constraints from sum of degrees of the bodies.

Or in mathematical form,

$$DoF = m(N - 1 - J) + \sum_{i=1}^{J} f_i$$, where

m is the number of freedom of a rigid body ($m = 3$ for planar and $m = 6$ for spatial mechanism)

N is the number of links,

J is the number of joints, and

f_i are the number of freedoms provided by joint i.

Parameters

The parameters should be given interactively, like

m = input("Is the mechanism planar [3] or spatial [6]?");

N = input("How many links are there in the mechanism?");

J = input("What about the number of joints?");

f = input("Finally, give me the number of freedoms provided by these joints. It should be a vector, the dimension of which is same as the number of joints.")

However, we will demonstrate the usage with several sets of fixed input values.

The first example represents open chain mechanism with 3 revolute joints. (RRR). The second example represents closed chain mechanism with 4 revolute joints (RRRR). The third one is a spatial example with 4 universal joints (UUUU). The rest ones are error cases.

```
calcDof(3, 4, 3, ones([3, 1]))
calcDof(3, 4, 4, ones([4, 1]))
calcDof(6, 5, 4, ones([4, 1])*2)
calcDof(4, 5, 4, ones([4, 1])*2)
```

```
calcDof(3, 4, 3, ones([2, 3])*2)
```

Matlab Implementation and Output

```
function dof = calcDof(m, N, J, f)
    if m ~= 3 && m ~= 6
        disp("m should either be 3 or 6!");
    elseif sum(size(f) ~= [J, 1]) ~= 0 && sum(size(f) ~= [1, J]) ~= 0
        disp("The size of f does not match J!");
    else
        dof = m * (N - 1 - J) + sum(f);
    end
end
```

```
ans =

     3


ans =

     1


ans =

     8

m should either be 3 or 6!
The size of f does not match J!
```

Published with MATLAB® R2020b