ESW: BattleBots Spring Force Requirement

Hannah Bissonette

I. Constraining the problem:

The flipper mechanism must be capable of generating sufficient force to flip the opposing robots. Therefore, the analysis must consider that we do not know the shape, size, and weight of the opposing robot. The rules of the competition provide limitations to the size and weight of the robot, so those provide the initial constraints.

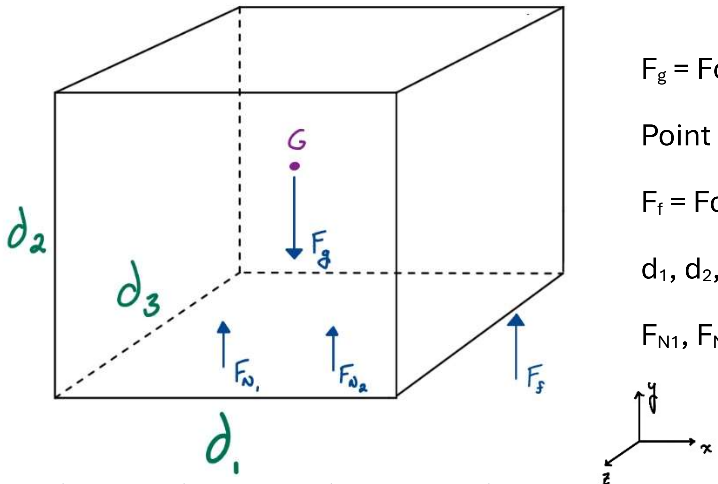
From the Competition Rulebook. Rules_21-22.pdf

a. The maximum weight of a bot is 50 lbs.
    i. Robots will be weighed in fighting condition, which includes ALL batteries and circuitry. To ensure this rule is followed, see Match Rules 1a.
b. Maximum dimensions of a bot shall not exceed 48 inches, and the combined length of three sides shall not exceed 88 inches.
    i. Expansion to a greater size once combat has begun is permitted.
    ii. Robots that expand (or separate into multi-bots) after each match has begun do not need to be able to return to their original smaller state by themselves.
    iii. Multi-bots must start the match as one unit (i.e. the total space that they occupy must be within the maximum dimensions listed above).

We only need to find the maximum force required for any shape, weight, and size. It is far too time-consuming to test all possibilities, so the problem needs to be simplified.

II. Simplify the problem:

a. The force of gravity will act against the force we need to flip the robot. The most difficult robot to flip will be the heaviest. We will analyze only the maximum allowable weight of 50 pounds.
b. Assuming any expansion of the opposing robot will aid rotation. Therefore, we will exclude expansion beyond given dimension constraints in the analysis.
c. Consider the drawing below. The box is representative of the dimensions of the opposing robot. This is a static equilibrium drawing of the point when the flipping mechanism first touches the robot. It represents the point in time before the attack is initiated.



F_g = Force of gravity

Point G = Center or gravity

F_f = Force of flipping mechanism

d_1, d_2, d_3 = Length dimensions of robot

F_{N1}, F_{N2} = Normal forces from tires

Figure 1: Static Free Body Diagram - Opposing Robot

All dimensions and locations of forces are arbitrarily defined. The diagram assumes 2 normal forces from the tires, even though there can be more than two tires. This will be further explained later. Now consider that we do not want the effectiveness of the mechanism to be dependent on F_f's position in the z direction. Therefore, we simplify this into a 2-D problem, making the force calculation independent of the F_f's z position.

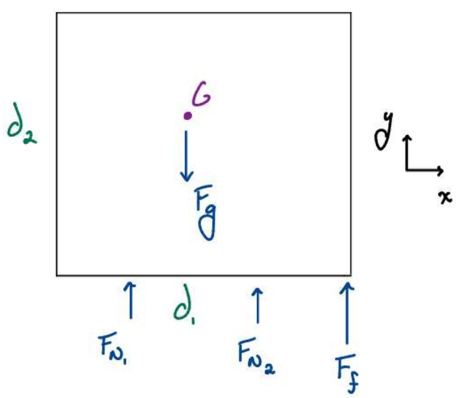


Figure 2: Simplified Free Body Diagram - Opposing Robot

d. The normal force on the tires acts in the same direction as the force of the flipping mechanism. Therefore, a greater force will be required once the tires have been lifted. The analysis will now be done at the point in time after the tires have been lifted.
e. The rotation of the opposing robot will go around a circular path. Instead of using the sum of forces in x and y directions, normal and tangential directions will be used. Acceleration in the tangential direction is equivalent to the radius of the circular path multiplied by the angular acceleration.
f. The robot will be rotated around the point on the opposite side of the flipping force. This will make that point the center of the circle the robot is moving along. This simplifies the radius of the circular path to d_1.
g. It is not feasible for the robot to have one of the three dimensions go below 5 inches and still be 50 pounds. The minimum for all dimensions will be set to 5 inches.
h. The angular acceleration is the same anywhere on the opposing robot and for the flipping mechanism.
i. In the design process, we may choose to use multiple springs and/ or use tension from a cable to satisfy the force requirement. For this analysis, we will treat it as one force and more can be added later as necessary.
j. The flipping mechanism rotates around the pivot point, which is where the mechanism is secured. This point has purely rotational motion and therefore does not have any acceleration in the y direction.
    i. This can be used later to ensure that the pin used can handle the stress. It is not relevant for this calculation.

III. <u>Developing Equations:</u>

a. <u>Equation for flipping force:</u>

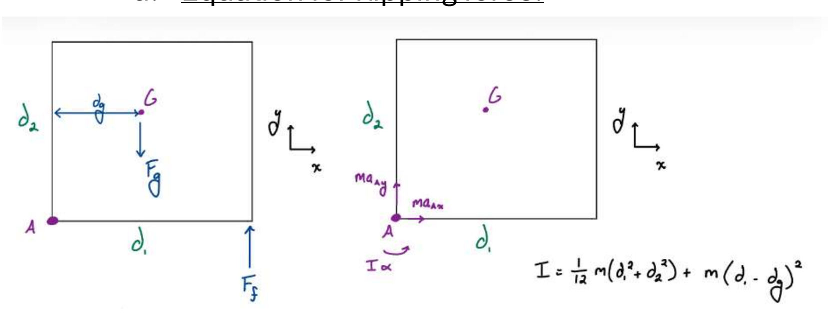


Figure 3: First Dynamic Analysis - Opposing Robot

d_g = X distance from the center of gravity to point A

I = Mass moment of inertia

$a_{ay,x}$ = Acceleration in the y & x directions respectively

α = angular acceleration

Solving for the sum of moments equation about point A, and using the equation for mass moment of inertia for a cuboid with parallel axis theorem

$F_f = (I*\alpha)/d_1 + F_g*(d_g/d_1)$

$I = (1/12)*m*(d_1^2 + d_2^2) + m*(d_1 - d_g)^2$

We still need to solve for α.

b. <u>Equation for angular acceleration:</u>

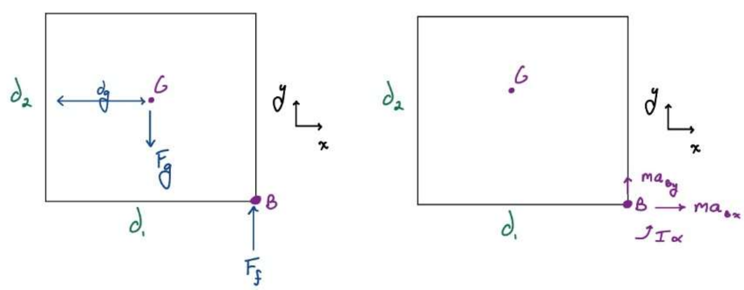


Figure 4: Second Dynamic Analysis - Opposing Robot

Sum of moments, $\alpha = F_g*(d)/I$. When plugged back in, it cancels to show that $F_f = F_g$. These two equations are necessary for the spring force requirement calculation.

c. <u>Equation for spring force requirement:</u>

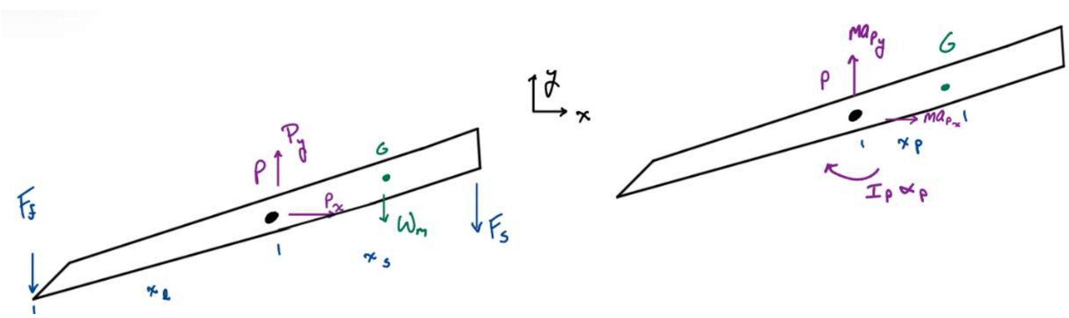


Figure 5: Dynamic Analysis - Flipper Mechanism

W_m = Weight of flipping mechanism

F_s = Spring force requirement

x_l = Distance from interfacing point to pivot point

x_s = Distance from pivot point to acting point of spring force

x_p = distance from the center of gravity to the pivot point

Using the sum of moments, the equation $F_s = (F_g x_l - W_m x_p + I_p \alpha)/ x_s$

And treating the flipping mechanism as a flat bar,

$I_p = (1/12)*m*L^2 + m*(x_p)^2$

IV. <u>Results:</u>

a. <u>Maximum required angular acceleration:</u>

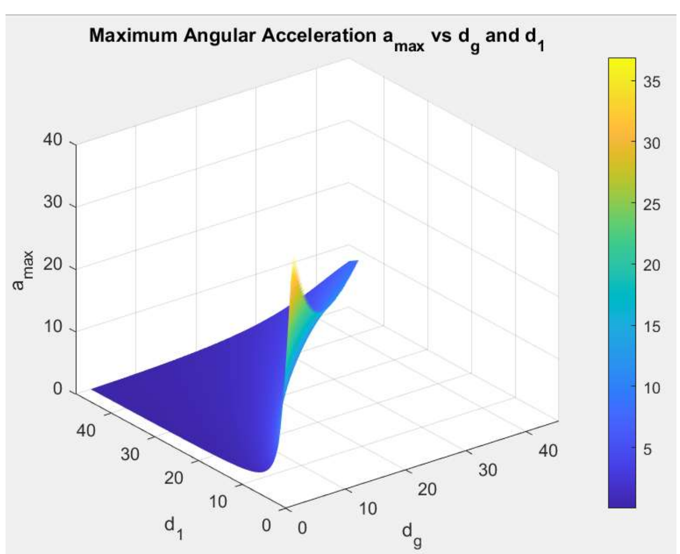
MATLAB Code:

```
clear; clc;

% Define ranges & constants
d_g = linspace(0, 48, 150);
d_1 = linspace(5, 48, 150);
d_2 = linspace(5, 35, 150);
F_g = 50;
m = 50 / 32.2;

% Create grid
[Dg, D1, D2] = meshgrid(d_g, d_1, d_2);

% Compute angular acceleration for all values
a_max = (F_g .* Dg) ./ ((1/12) * m .* ((D1.^2) + (D2.^2)) + m .* (D1 - Dg).^2);

% d_g < d_1
a_max(Dg >= D1) = NaN;

% Find the maximum across all d_2 values
a_max = max(a_max, [], 3, 'omitnan');

% Find overall maximum and its location
[max_alph, idx] = max(a_max(:), [], 'omitnan');
[row, col] = ind2sub(size(a_max), idx);
max_dg = d_g(col);
max_d1 = d_1(row);

% Values
fprintf('Maximum a_max = %.4f\n', max_alph);
fprintf('At d_g = %.4f and d_1 = %.4f\n',max_dg, max_d1);

% Plot
figure(1);
surf(d_g, d_1, a_max);
xlabel('d_g');
ylabel('d_1');
zlabel('a_{max}');
title('Maximum Angular Acceleration a_{max} vs d_g and d_1');
shading interp;
colorbar;
```

Return:

Maximum a_max = 36.8783

At d_g = 5.0973 and d_1 = 5.2886



b. Spring force required:

This equation, even with the maximum angular acceleration determined, is still entirely dependent on the geometry of the mechanism.

The equation:

$F_s = (F_g x_l - W_m x_p + \alpha_{max}(mL^2/12 + mx_p)^2)/x_s$

There are a few ways to use this equation meaningfully. Ranges can be plugged into the equation to determine the optimal way to design the mechanism. Loops can be made for reasonable ranges of all the above variables to minimize the spring force requirement. Minimizing the spring force requirement would mean optimizing the geometric design of the mechanism.

V. Next Steps:

a. Further Examination:

i. Considering previous errors in my work, it is important that, before design decisions are made that somebody capable of affirming the sensibility of my assumptions and the factuality of the derived equations.

ii. I believe I have already found someone willing and capable. Hopefully, this step can be met soon.

b. Reasonable Ranges:

i. Below is a MATLAB code to calculate the required spring force when the relevant dimensions are provided. This is a guess-and-check approach that is inefficient and does not optimize geometry. I would rather provide ranges, but I do not know how much space we want to allocate to the mechanism.

ii. Silly Code:

```
clear;clc
% Pre-defined values
F_g = 50;
a_max = 37.6817;
% Plug in dimensions
W_m = input('The weight of the flipping mechanism is (lbs) ');
L = input('The length of the flipping mechanism is (in) ');
x_p = input('The distance from the pinned point to the center of mass is (in) ');
x_l = input('The distance from the interface point to the pinned point is (in) ');
x_s = input('The distance between the pinned point and the spring force is (in) ');
% Constants to calculate
m = W_m/32.2;
% Calculate spring force required
F_s =(1/x_s)*(F_g*x_l - W_m*x_p + a_max*m*((1/12)*(L^2)+(x_p^2)));
disp('The force required of the spring is ');disp(F_s);
```