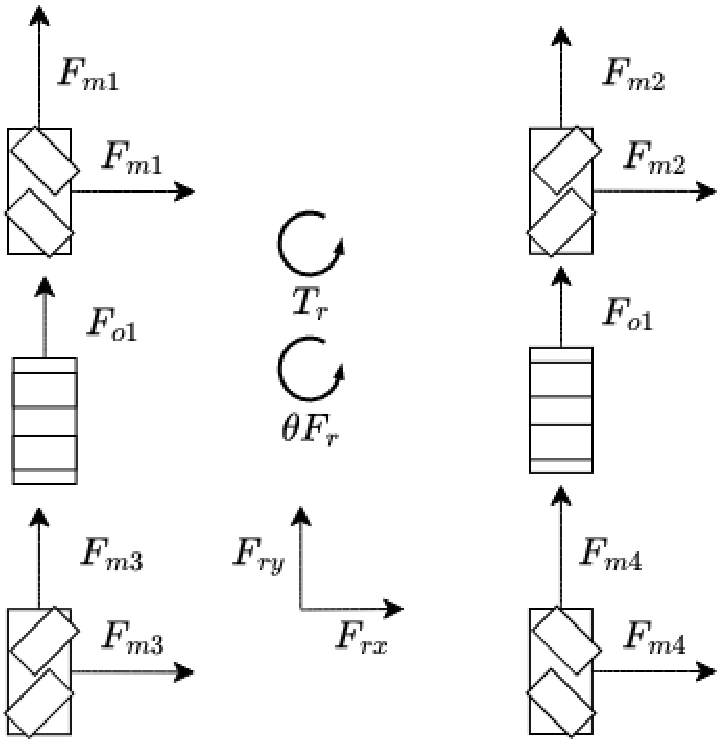

Given

$F_{rxm} = F_{m2} + F_{m3} - F_{m1} - F_{m4}, F_{rxo} = 0$

$F_{rym} = F_{m1} + F_{m2} + F_{m3} + F_{m4}, F_{ryo} = F_{o1} + F_{o2}$

$\theta F_{rm} = \tan^{-}1\left(\frac{F_{rym}}{F_{rxm}}\right), \theta F_{ro} = 0$

$F_{rx} = F_{rxm}, F_{ry} = F_{rym} + F_{ryo}, \theta F_r = \tan^{-1}\left(\frac{F_{ry}}{F_{rx}}\right)$

$$F_{rx} = F_{m2} + F_{m3} - F_{m1} - F_{m4}$$

$$F_{ry} = F_{m1} + F_{m2} + F_{m3} + F_{m4} + F_{o1} + F_{o2}$$

$$\theta F_r = \tan^{-1}\left(\frac{F_{m1} + F_{m2} + F_{m3} + F_{m4} + F_{o1} + F_{o2}}{F_{m2} + F_{m3} - F_{m1} - F_{m4}}\right)$$

$F_{rim} = F_{m2} + F_{m4}$

$F_{lm} = F_{m1} + F_{m3}$

$\tau_{rm} = \frac{L+W}{4}(F_{rim} - F_{lm})$

$\tau_{ro} = \frac{W}{2}(F_{o2} - F_{o1})$

$\tau_r = \tau_{rm} + \tau_{ro}$

$\tau_r = \frac{L+W}{4}(F_{rim} - F_{lm}) + \frac{W}{2}(F_{o2} - F_{o1})$

$$\tau_r = \frac{L+W}{4}(F_{m2} + F_{m4} - F_{m1} - F_{m3}) + \frac{W}{2}(F_{o2} - F_{o1})$$

Let $T = \begin{pmatrix} m \\ o \end{pmatrix}$, $N_m = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$ and $N_o = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$. $\forall t \in T$ and $n \in N_t$, we have

$$k_{tn}^{\min} \le F_{tn} \le k_{tn}^{\max}$$

Now we want to make some way to input θF_r, τ_r, and $\forall t \in T$ and $n \in N_t$: $k_{tn}^{\min}$ and $k_{tn}^{\max}$ and then somehow get out $\forall t \in T$ and $n \in N_t$: F_{tn} that maximizes $F_{rx} + F_{ry}$

First let's find the "objective function". Substituting we get

$$F_{m2} + F_{m3} - F_{m1} - F_{m4} + F_{m1} + F_{m2} + F_{m3} + F_{m4} + F_{o1} + F_{o2}$$

Simplifying we then get

$$2F_{m2} + 2F_{m3} + F_{o1} + F_{o2}$$

We can then use the simplex algorithm to solve this. To input into the simplex algorithm it says to maximize c^T x subject to $Ax \le b$ and $x \ge k$ where

- A is an $m \times n$ matrix
- $b \in \mathbb{R}^m$
- $c \in \mathbb{R}^n$
- $x \in \mathbb{R}^n$ are the decision variables
- x is the decision variable vector
- c is the objective-coefficient vector
- $c^T x$ is the objective function
- A is the constraint matrix
- b is the right-hand side vector
- $k \in \mathbb{R}$

First, let's put down what we know. We know the decision variable vector x would be $\begin{pmatrix} F_{m1} \\ F_{m2} \\ F_{m3} \\ F_{m4} \\ F_{o1} \\ F_{o2} \end{pmatrix}$.

Back from what we found before because of that we know the decision variable vector c is $\begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \\ 1 \\ 1 \end{pmatrix}$.

So $c^T x = 2F_{m2} + 2F_{m3} + F_{o1} + F_{o2}$ (what we found earlier).

Now we are really close, all we have to worry about is A. So first we know a bunch of the inequalities:

$\forall t \in T$ and $n \in N_t$

$k_{tn}^{\min} \le F_{tn} \le k_{tn}^{\max}$

we need to then put it in the form $Ax \le b$ so we get

$\forall t \in T$ and $n \in N_t$

$F_{tn} \le k_{tn}^{\max}$, $-F_{tn} \le -k_{tn}^{\min}$

Ok but this isn't in the form with x, it is each of the forces separately. So since we only do one at a time the other forces would just be multiplied by 0. There would be 1 row for each force. This can be written as an identity matrix with the size of 6: I_6. This is done for both the mininum and maximum, with the minimum being negative: $-I_6$. Therefore we get the following inequalities:

$\forall t \in T$ and $n \in N_t$

$$I_6 x \leq k_{tn}^{\max}$$

$$-I_6 x \leq -k_{tn}^{\min}$$

So we know some of the constraints. The next constraint is the angle of force contraint. Now we need to linearize θF_r.

First we flip around the trig function to get

$\tan(\theta F_r) = \frac{F_{m1}+F_{m2}+F_{m3}+F_{m4}+F_{o1}+F_{o2}}{F_{m2}+F_{m3}-F_{m1}-F_{m4}}$

Then simplifying we get

$$F_{m1} + F_{m2} + F_{m3} + F_{m4} + F_{o1} + F_{o2} - \tan(\theta F_r)(F_{m2} + F_{m3} - F_{m1} - F_{m4}) = 0$$

we can then factorize to get

$$F_{m1}(1 + \tan(\theta F_r)) + F_{m2}(1 - \tan(\theta F_r)) + F_{m3}(1 - \tan(\theta F_r)) + F_{m4}(1 + \tan(\theta F_r)) + F_{o1} + F_{o2} = 0$$

We can then take the factors to the decision variables to get a vector

$$a_{\text{ang}} = \begin{pmatrix} 1 + \tan(\theta F_r) \\ 1 - \tan(\theta F_r) \\ 1 - \tan(\theta F_r) \\ 1 + \tan(\theta F_r) \\ 1 \\ 1 \end{pmatrix}$$

Which we can convert into two inequalities

$$a_{\text{ang}}^T x \leq 0$$

$$-a_{\text{ang}}^T x \leq 0$$

Now onto the last and final constraint we have is the torque constraint which we can apply the same steps and factorize getting the new vector

$$a_\tau = \begin{pmatrix} -\frac{L+W}{4} \\ \frac{L+W}{4} \\ -\frac{L+W}{4} \\ \frac{L+W}{4} \\ -\frac{W}{2} \\ \frac{W}{2} \end{pmatrix}$$

giving the following two inequalities

$$a_\tau^T x \leq \tau_r$$

$$-a_\tau^T x \leq \tau_r$$

This gives us the total matrix

$$A = \begin{pmatrix} I_6 \\ -I_6 \\ a_{\text{ang}}^T \\ -a_{\text{ang}}^T \\ a_\tau^T \\ -a_\tau^T \end{pmatrix}$$

and the vector

$$b = \begin{pmatrix} k_{\max} \\ -k_{\min} \\ 0 \\ 0 \\ \tau_r \\ \tau_r \end{pmatrix}$$

We can then put this into a simplex (or other algorithm) solver and it should output the correct answer!

Mecanum force maximization

Let the decision vector be

x := (F_m1, F_m2, F_m3, F_m4, F_o1, F_o2) in R^6.

Define the following row vectors (linear forms) acting on x:

a_rx := (-1, 1, 1, −1, 0, 0), qquad a_ry := (1, 1, 1, 1, 1, 1).

a_ang := (1 + tan(theta_F), 1 - tan(theta_F), 1 - tan(theta_F), 1 + tan(theta_F), 1, 1)

a_tau := (-(L + W)/4, (L + W)/4, -(L + W)/4, (L + W)/4, -W/2, W/2)

Using these linear forms we may write the following equalities:

F_rx = a_rx x, F_ry = a_ry x,

a_ang x = F_ry - tan(theta_F) F_rx, tau = a_tau x.

Let k_min, k_max in R^6 denote elementwise box bounds: k_min <= x <= k_max.

Theorem

Maximizing the Euclidean resultant force

norm((F_rx, F_ry)) = sqrt(F_rx^2 + F_ry^2)

subject to the angle equality a_ang x = 0, a torque condition, and box bounds is equivalent to the convex second-order-cone program below.

SOCP formulation (exact-angle, exact-torque)

Maximize s over x in R^6 and s in R subject to the constraints

sqrt((a_rx x)^2 + (a_ry x)^2) <= s

a_ang x = 0

a_tau x = tau_r

k_min <= x <= k_max

If the torque is intended as a bound rather than an equality, replace the equality by

abs(a_tau x) <= tau_r

i.e. the two linear inequalities

a_tau x <= tau_r and -a_tau x <= tau_r.

Proof

Step 1 (auxiliary scalar). Introduce s in R. Since the square-root is monotone increasing on [0, +infty), maximizing sqrt(F_rx^2 + F_ry^2) is equivalent to maximizing s subject to

sqrt(F_rx^2 + F_ry^2) <= s.

Step 2 (SOC cast). The inequality above is the second-order-cone constraint

sqrt((F_rx)^2 + (F_ry)^2) <= s,

or, using the linear maps,

sqrt((a_rx x)^2 + (a_ry x)^2) <= s.

Step 3 (linear constraints). The angle condition tan(theta_F) = F_ry / F_rx (when F_rx != 0) is algebraically equivalent to

F_ry - tan(theta_F) F_rx = 0,

which is the linear equality a_ang x = 0. The torque and box constraints are linear equalities/ inequalities in x.

Step 4 (convexity and equivalence). The objective s is linear. The SOC constraint and the linear equalities/inequalities are convex. The feasible set is convex, and the introduced scalar s enforces exact equivalence with maximizing the Euclidean norm. Hence the SOCP above is a correct exact convex reformulation.

End of proof.

Remark (why the previous LP objective was incorrect)

If one sets the LP objective vector c = (0, 2, 2, 0, 1, 1)^T then

c^T x = 2 F_m2 + 2 F_m3 + F_o1 + F_o2.

One checks

F_rx + F_ry

(F_m2 + F_m3 - F_m1 - F_m4)

1. (F_m1 + F_m2 + F_m3 + F_m4 + F_o1 + F_o2)

2 F_m2 + 2 F_m3 + F_o1 + F_o2 = c^T x.

However F_rx + F_ry is a linear functional and in general is not equal to sqrt(F_rx^2 + F_ry^2). Thus maximizing c^T x (an LP) optimizes the sum F_rx + F_ry, not the Euclidean magnitude. Use the SOCP above for the exact solution, or if you must remain with LPs use a polyhedral approximation.