$r = 0$
$q = -0.1$
$p = -0.3$

Thruster Controller

Let's assume we have a body vector

$\vec{F}_d$ & $\vec{M}_d$ which are desired forces & moments

the force of 1 thruster is $\vec{F}_i = \sigma_i \bar{n}_i$ where σ_i is either a 0 or al and $\bar{n}_i$ is a unit vector.

the total force of the thrusters is then simply

$$\vec{F}_T = \sum_{i=1}^{N} \sigma_i \bar{n}_i \quad \text{, similarly} \quad \vec{M}_T = \sum_{i=1}^{N} \sigma_i (\vec{r}_i \times \bar{n}_i)$$

where $\vec{r}_i$ is a vector from the cg to the thruster.

Our challenge then is to find σ_i s.t.

$$\vec{F}_d = \sum_{i=1}^{N} \sigma_i \bar{n}_i \quad \text{and} \quad \vec{M}_d = \sum_{i=1}^{N} \sigma_i (\vec{r}_i \times \bar{n}_i)$$

$$\underset{3\times1}{} = \underset{1\times1}{\sigma_1}\underset{3\times1}{\bar{n}_1} + \sigma_2 \bar{n}_2 = \begin{Bmatrix} \sigma_1 n_{1x} + \sigma_2 n_{2x} \\ \sigma_1 n_{1y} + \sigma_2 n_{2y} \\ \sigma_1 n_{1z} + \sigma_2 n_{2z} \end{Bmatrix}$$

$$= \underset{1\times2}{\begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix}} \underset{2\times3}{\begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \end{bmatrix}}$$

$$= \begin{bmatrix} n_{1x} & n_{2x} \\ n_{1y} & n_{2y} \\ n_{1z} & n_{2z} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix}$$

let's expand these out

$$\vec{F}_d = \sigma_1 \bar{n}_1 + \sigma_2 \bar{n}_2 + \ldots \sigma_N \bar{n}_N$$

$$\vec{M}_d = \sigma_1 \hat{r}_1 \bar{n}_1 + \sigma_2 \hat{r}_2 \bar{n}_2 + \ldots \sigma_N \hat{r}_N \bar{n}_2$$

where $\vec{r}_i \times \bar{n}_i = \hat{r}_i \bar{n}_i$ & $\hat{r}_i$ is the skew symmetric operator for a vector.

in Matrix form

$$\vec{F}_d = \underline{\underline{N}} \vec{\sigma} \quad \text{where} \quad \vec{\sigma} = \begin{Bmatrix} \sigma_1 \\ \vdots \\ \sigma_N \end{Bmatrix} \quad \underline{\underline{N}} = [\bar{n}_1 \ldots \bar{n}_N]$$

since $\underline{\underline{N}}$ is not typically square you must find the pseudo inverse. $\boxed{\vec{\sigma}^* = (N^T N)^{-1} N^T \vec{F}_d}$

$\vec{\sigma}^*$ is deprecated unfortunately. if $\sigma_i < 0$, $\sigma_i = 0$

if $\sigma_i < 1$, $\sigma_i = 0$ ← just these two

if $\sigma_i > 1$, $\sigma_i = 1$ ←

Finding Moment desired is identical

$$\vec{M}_d = \overset{3\times 8}{\underline{\underline{RN}}}\ \overset{8\times 1}{\vec{\sigma}} \quad \text{where } \underline{\underline{RN}} = \left[\hat{r}_1 \bar{n}_1 \cdots \hat{r}_N \bar{n}_N\right]$$

then $$\boxed{\vec{\sigma}^* = \left(\underline{\underline{RN}}^T \underline{\underline{RN}}\right)^{-1} \underline{\underline{RN}}^T \vec{M}_d}$$

$$\underbrace{8\times 3 \quad 3\times 8}_{(8\times 8)} \ (8\times 3) \rightarrow 8\times 3$$

3 eqns & 8 unknowns?
hmm. I'm not sure this will work.
MATLAB?? yea.

See poster note in back of binder

$$\vec{\sigma}^* = \underbrace{\underline{\underline{RN}}^T \left(\underline{\underline{RN}}\,\underline{\underline{RN}}^T\right)^{-1}}_{\text{when } c>r} \vec{M}_d \qquad \underset{3\times 1}{\vec{M}_d} = \underset{3\times 8}{\underline{\underline{RN}}}\ \underset{8\times 1}{\vec{\sigma}}$$

← columns > rows more unknowns than equations

$$(8\times 1) = \underbrace{(8\times 3)(3\times 8)(8\times 3)}(3\times 1)$$

$$8\times 3\ (3\times 3)^{-1}$$

$$8\times 3\ (3\times 3)$$

$$8\times 3$$

Solution of this optimizer is always transpose of the input.