# Table of Contents

 ........................................................................................................................ 1
x, y, and psi component of tau_d ........................................................................................ 1
Quadratic Programming ...................................................................................................... 1
RESULTS ........................................................................................................................... 2

```
function [Td, alpha] = quad_prog_basic(tau_d)
```

# x, y, and psi component of tau_d

```
tau = [tau_d(1); tau_d(2); tau_d(6)];
```

```
Not enough input arguments.

Error in quad_prog_basic (line 3)
tau = [tau_d(1); tau_d(2); tau_d(6)];
```

# Quadratic Programming

Compute function arguments Point to Minimize

```
u_p = [0 0 0 0 0 0 0 0].'; % [T1 T2x T2y T3 T4x T4y T5x T5y]
u_m = -[0 0 0 0 0 0 0 0].'; % [T1 T2x T2y T3 T4x T4y T5x T5y]
s_p = [0 0 0].'; % [x1 x2 x6]
s_m = -[0 0 0].'; % [x1 x2 x6]
x0 = [u_p; u_m; s_p; s_m]; % auxiliary variables

% Objective Function
% Quadratic Objective Term
w = [1 1 1 1 100 100 100 100]; % Prioritize control over T4 and T5
q = [1 10 10]; % For slackness (Needs Tuning)
H = 2*diag([w w q q]);

% Linear Objective Term
% f = zeros(1, length(x0)); % No linear term in the objective function
f = [];

% Thruster Equality Constraints
% Thruster 1
x1 = 39.3;
y1 = 0;
% Thruster 2
x2 = 35.6;
y2 = 0;
% Thruster 3
x3 = 31.3;
y3 = 0;
% Thruster 4
x4 = -28.5;
```

```
y4 = 5;
% Thruster 5
x5 = -28.5;
y5 = -5;

% Redistributed force for each thruster
thr1 = [0 1 x1].';
thr2_x = [1 0 y2].';
thr2_y = [0 1 x2].';
thr3 = [0 1 x3].';
thr4_x = [1 0 y4].';
thr4_y = [0 1 x4].';
thr5_x = [1 0 y5].';
thr5_y = [0 1 x5].';

% B redistributed
B_r = [thr1 thr2_x thr2_y thr3 thr4_x thr4_y thr5_x thr5_y];

Aeq = [B_r -B_r eye(3) -eye(3)];
beq = tau;

% Thruster Saturation Constraints
u_max = [125 150 150 125 300 300 300 300].';
u_min = [-125 0 0 -125 0 0 0 0].';

A = [eye(8) -eye(8) zeros(8,3) zeros(8,3);
     -eye(8) eye(8) zeros(8,3) zeros(8,3)];

b = [u_min;
     u_max];

lb = [];
ub = [];

options = optimoptions('quadprog', 'Algorithm', 'active-set');

% Compute optimal thruster control
x0_opt = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0, options);

% Slack Variable from auxiliary variables
s = x0_opt(17:19) - x0_opt(20:22); % STILL DON'T NOW WHAT TO DO WITH THIS
s_abs = x0_opt(17:19) + x0_opt(20:22); % STILL DON'T NOW WHAT TO DO WITH THIS

% Allocation variables from auxiliary variables
u = x0_opt(1:8) - x0_opt(9:16);
u_abs = x0_opt(1:8) + x0_opt(9:16);
```

# RESULTS

Thruster control

```
Td = [u(1);
      sqrt(u(2)^2 + u(3)^2);
      u(4);
      sqrt(u(5)^2 + u(6)^2);
      sqrt(u(7)^2 + u(8)^2)];

% Thruster angle
alpha2 = atan(u(3)/u(2));
alpha4 = atan(u(6)/u(5));
alpha5 = atan(u(8)/u(7));
alpha = [pi/2; alpha2; pi/2; alpha4; alpha5];

end
```

*Published with MATLAB® R2023a*