Lab05 - Rocket fuel optymalization, Pawel Drapiewski 11.03.2018 r.

Table of Contents

In this problem we will be searching for the most optimal rocket trace, where there will be the smallest fuel amount burned and the rocket reach all points.

```
clear all;
close all;

% import data for the excersice from outter file
run thrusters_data.m
```

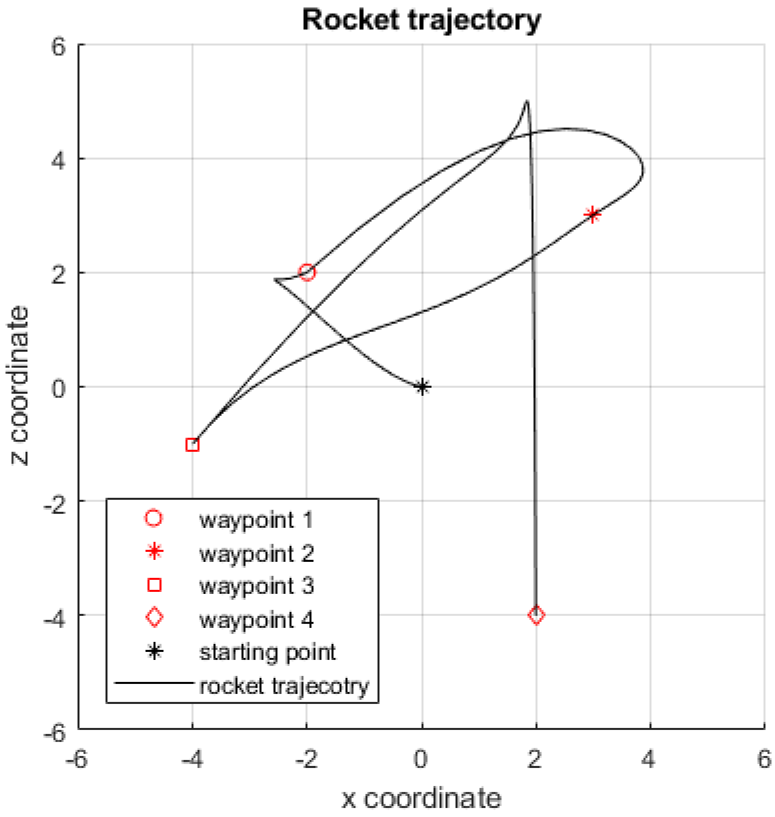
Part 1: No trust constrains

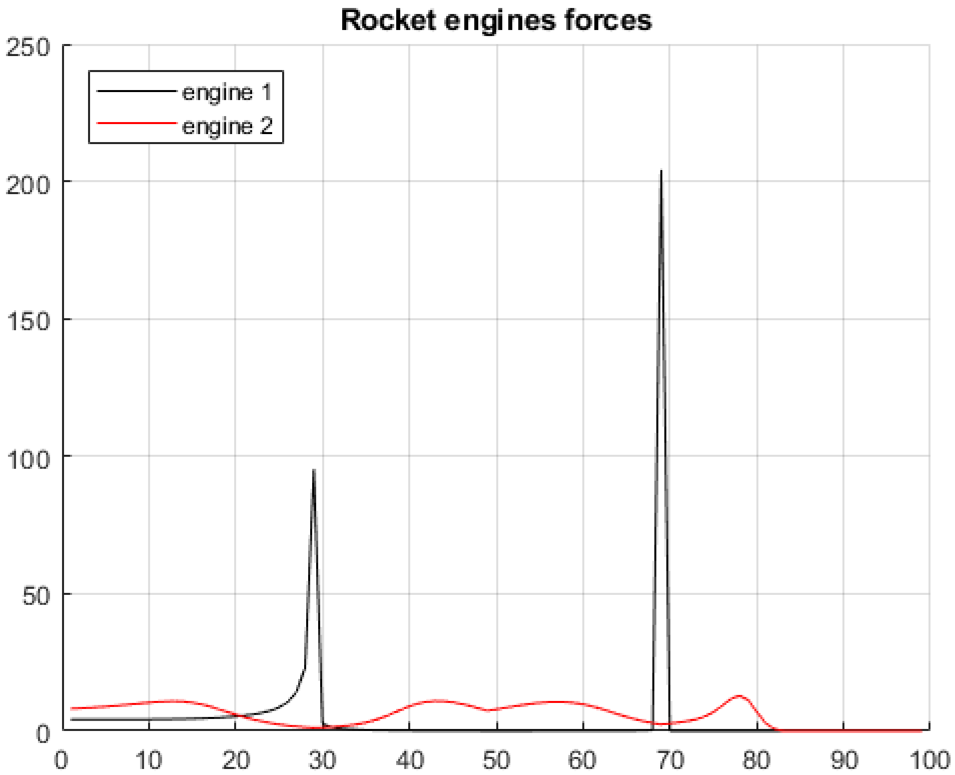
```
cvx_solver('sdpt3');
cvx_begin
variables engine1_power(K-1) engine2_power(K-1) force_vector_x(K) force_vector_y(K
minimize (sum(engine1_power)+sum(engine2_power))
subject to
    % where rocket starts
    points_x(1) == 0;
    points_y(1) == 0;
    % rocket wasn't moving on the start
    velocity_x(1) == 0;
    velocity_y(1) == 0;

    points_x(k1) == w1(1);
    points_y(k1) == w1(2);

    points_x(k2) == w2(1);
    points_y(k2) == w2(2);

    points_x(k3) == w3(1);
    points_y(k3) == w3(2);

    points_x(k4) == w4(1);
    points_y(k4) == w4(2);

    % engine power must be grather than 0!
    engine1_power >= 0; engine2_power >= 0;
    % bounds of the trajecory
    abs(points_x) <= pmax; abs(points_y) <= pmax;

    for k=1:K-1
     % Rocket model
     force_vector_x(k)== cos(theta1)*engine1_power(k) +
 cos(theta2)*engine2_power(k) + 0;
     force_vector_y(k)== sin(theta1)*engine1_power(k) +
 sin(theta2)*engine2_power(k) - m*g;

     velocity_x(k+1)==(1-alpha)*velocity_x(k) + (h/
m)*force_vector_x(k);
     velocity_y(k+1)==(1-alpha)*velocity_y(k) + (h/
m)*force_vector_y(k);

     points_x(k+1) == points_x(k) + h*velocity_x(k);
     points_y(k+1) == points_y(k) + h*velocity_y(k);
    end
cvx_end

total_fuel_consumption = (sum(engine1_power)+sum(engine2_power));
disp('Found minimal fuel consumption: ')
disp(total_fuel_consumption)


% draw the waypoints and rocket trajectory
figure(1);
hold on;
axis equal;
grid on;
title("Rocket trajectory");
xlim([-pmax-1 pmax+1]);
ylim([-pmax-1 pmax+1]);
xlabel("x coordinate");
ylabel("z coordinate");

point1 = plot(w1(1), w1(2) ,'ro');
point2 = plot(w2(1), w2(2) ,'r*');
point3 = plot(w3(1), w3(2) ,'rs');
point4 = plot(w4(1), w4(2) ,'rd');
point_start = plot(points_x(1), points_y(1), 'black*');
rocket_trajectory = plot(points_x, points_y, '-black');
legend([point1 point2 point3 point4 point_start
 rocket_trajectory],'waypoint 1','waypoint 2','waypoint 3','waypoint
 4', 'starting point', 'rocket trajecotry','Location','southwest');
hold off;

% draw plot of enigines trust over the time
figure(2);
hold on; grid on;
title('Rocket engines forces');
engine1 = plot(engine1_power, 'black-');
engine2 = plot(engine2_power, 'red-');
legend([engine1 engine2],'engine 1','engine
 2','Location','Northwest');
```

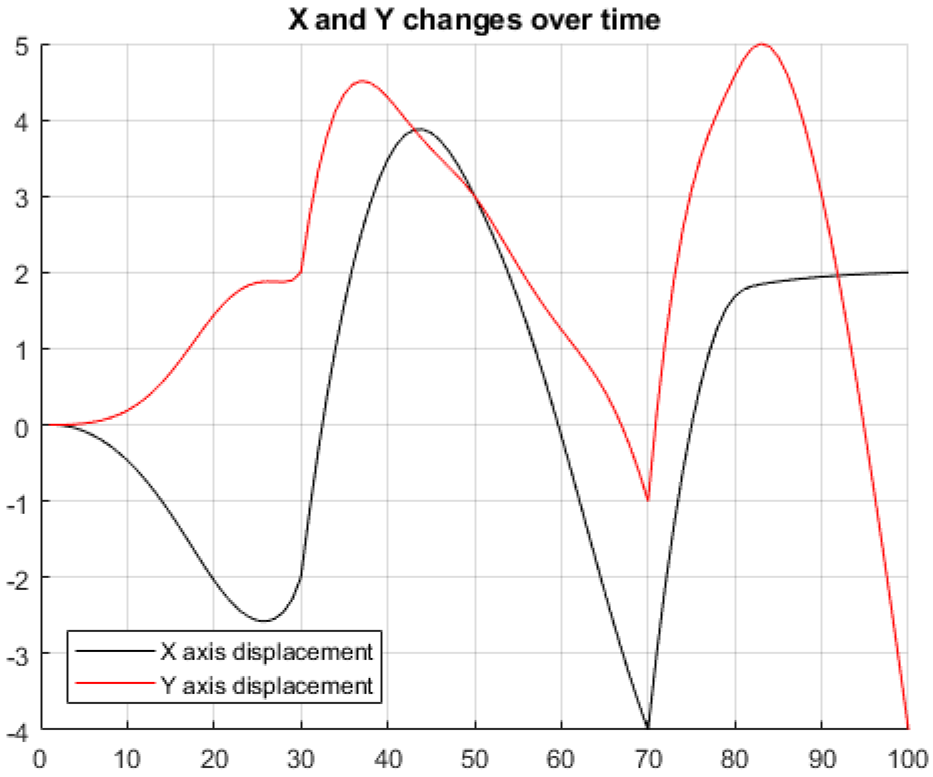
```
hold off

% Draw plot of  the X and Y position
figure(3);
hold on; grid on;
title('X and Y changes over time');
points_x = plot (points_x, 'black-');
points_y = plot(points_y, 'red-');
legend([points_x points_y],'X axis displacement','Y axis
 displacement','Location','Southwest');
hold off
```

```
Calling SDPT3 4.0: 800 variables, 408 equality constraints
------------------------------------------------------------

 num. of constraints = 408
 dim. of socp   var  = 400,   num. of socp blk  = 200
 dim. of linear var  = 398
 dim. of free   var  =  2 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-------------------------------------------------------------------
 0|0.000|0.000|9.4e+01|3.1e+01|5.9e+06| 1.459619e+05  0.000000e+00|
 0:0:00| chol  1  1
 1|0.588|0.122|3.9e+01|2.8e+01|2.6e+06| 6.248028e+04 -7.286643e+03|
 0:0:00| chol  1  1
 2|0.886|1.000|4.4e+00|2.5e-01|3.5e+05| 1.038611e+04 -4.445138e+04|
 0:0:00| chol  1  1
 3|0.877|1.000|5.4e-01|7.5e-02|6.3e+04| 2.349045e+03 -3.169547e+04|
 0:0:00| chol  1  1
 4|0.750|0.794|1.3e-01|2.1e-02|2.1e+04| 1.484675e+03 -1.495913e+04|
 0:0:00| chol  1  1
 5|0.758|0.904|3.2e-02|2.7e-03|8.1e+03| 1.239459e+03 -6.014713e+03|
 0:0:01| chol  1  1
 6|0.788|0.972|6.9e-03|6.6e-03|2.9e+03| 1.148724e+03 -1.661217e+03|
 0:0:01| chol  1  1
 7|0.997|0.880|2.1e-05|2.2e-03|4.5e+02| 1.090408e+03  6.412857e+02|
 0:0:01| chol  1  2
 8|0.961|0.944|8.6e-07|1.3e-04|1.8e+02| 1.065187e+03  8.824841e+02|
 0:0:01| chol  2  2
 9|0.941|0.936|2.8e-07|8.6e-06|3.0e+01| 1.057253e+03  1.027106e+03|
 0:0:01| chol  1  1
10|1.000|0.741|1.0e-07|2.3e-06|1.2e+01| 1.056206e+03  1.044692e+03|
 0:0:01| chol  2  2
11|1.000|0.950|2.5e-08|1.4e-07|4.4e+00| 1.055632e+03  1.051257e+03|
 0:0:01| chol  1  1
12|0.951|0.872|1.2e-08|2.5e-08|1.0e+00| 1.055405e+03  1.054366e+03|
 0:0:01| chol  2  1
```

```
13|0.980|0.982|4.6e-09|3.5e-09|1.9e-02| 1.055345e+03  1.055326e+03|
 0:0:01| chol  1  1
14|0.989|0.989|9.2e-10|1.8e-08|2.7e-04| 1.055344e+03  1.055344e+03|
 0:0:01| chol  2  2
15|0.991|0.989|1.3e-11|1.3e-09|5.4e-06| 1.055344e+03  1.055344e+03|
 0:0:01|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 15
 primal objective value =  1.05534376e+03
 dual   objective value =  1.05534375e+03
 gap := trace(XZ)       = 5.42e-06
 relative gap           = 2.57e-09
 actual relative gap    = 2.12e-09
 rel. primal infeas (scaled problem)   = 1.31e-11
 rel. dual     "        "       "      = 1.26e-09
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 2.5e+02, 1.2e+02, 4.9e+00
 norm(A), norm(b), norm(C) = 1.8e+02, 1.1e+02, 1.5e+01
 Total CPU time (secs)  = 0.61
 CPU time per iteration = 0.04
 termination code       =  0
 DIMACS: 3.6e-11  0.0e+00  9.5e-09  0.0e+00  2.1e-09  2.6e-09
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +1055.34

Found minimal fuel consumption:
   1.0553e+03
```


Rocket trajectory
z coordinate
x coordinate
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
waypoint 1
waypoint 2
waypoint 3
waypoint 4
starting point
rocket trajecotry


Rocket engines forces
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
90
100
engine 1
engine 2



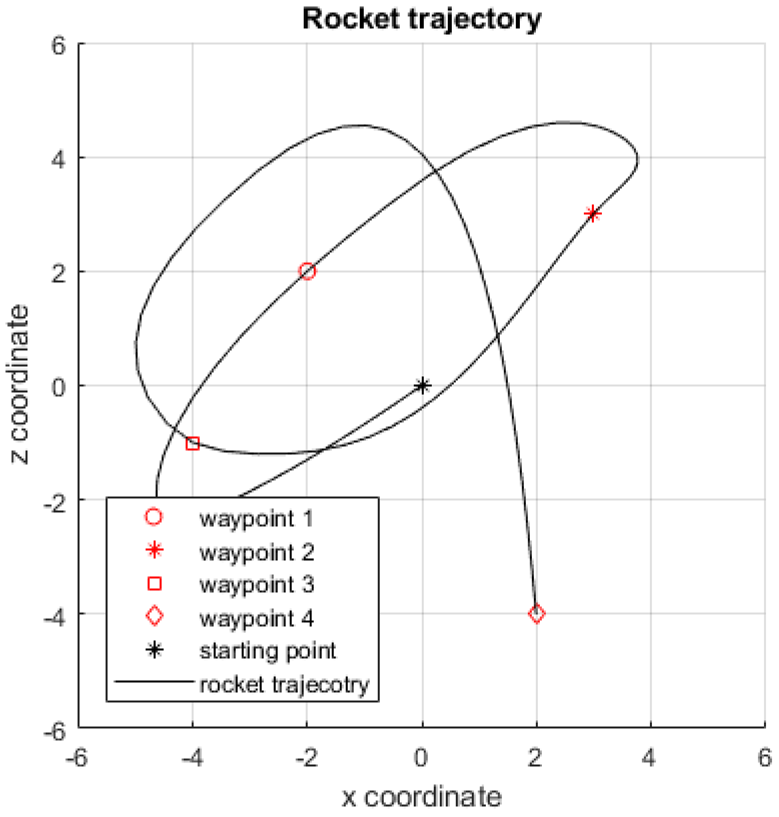
Part 2: Trust less than 20 for each engine

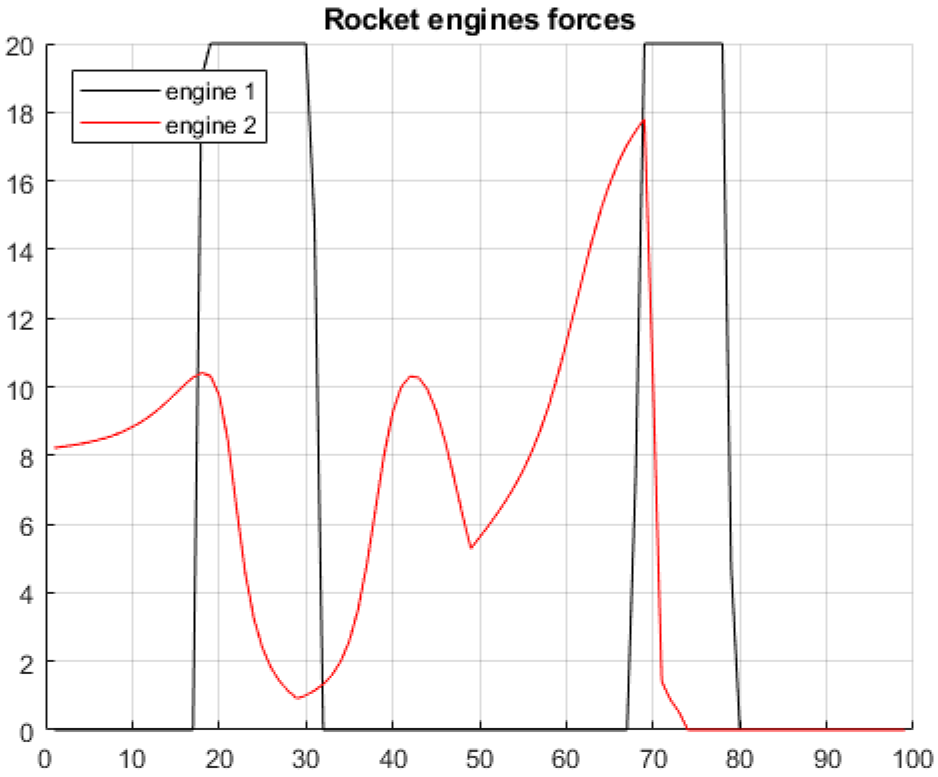
```
cvx_solver('sdpt3');
cvx_begin
variables engine1_power(K-1) engine2_power(K-1) force_vector_x(K) force_vector_y(K
minimize (sum(engine1_power)+sum(engine2_power))
subject to
    % where rocket starts
    points_x(1) == 0;
    points_y(1) == 0;
    % rocket wasn't moving on the start
    velocity_x(1) == 0;
    velocity_y(1) == 0;

    points_x(k1) == w1(1);
    points_y(k1) == w1(2);

    points_x(k2) == w2(1);
    points_y(k2) == w2(2);

    points_x(k3) == w3(1);
    points_y(k3) == w3(2);

    points_x(k4) == w4(1);
    points_y(k4) == w4(2);
    % engine power must be grather than 0!
    engine1_power >= 0; engine2_power >= 0;
    engine1_power < 20; engine2_power < 20;
    % bounds of the trajecory
    abs(points_x) <= pmax; abs(points_y) <= pmax;

    for k=1:K-1
     % Rocket model
     force_vector_x(k)== cos(theta1)*engine1_power(k) +
 cos(theta2)*engine2_power(k) + 0;
     force_vector_y(k)== sin(theta1)*engine1_power(k) +
 sin(theta2)*engine2_power(k) - m*g;

     velocity_x(k+1)==(1-alpha)*velocity_x(k) + (h/
m)*force_vector_x(k);
     velocity_y(k+1)==(1-alpha)*velocity_y(k) + (h/
m)*force_vector_y(k);

     points_x(k+1) == points_x(k) + h*velocity_x(k);
     points_y(k+1) == points_y(k) + h*velocity_y(k);
    end
cvx_end

total_fuel_consumption = (sum(engine1_power)+sum(engine2_power));
disp('Found minimal fuel consumption: ')
disp(total_fuel_consumption)


% draw the waypoints and rocket trajectory
figure(4);
hold on;
axis equal;
grid on;
title("Rocket trajectory");
xlim([-pmax-1 pmax+1]);
ylim([-pmax-1 pmax+1]);
xlabel("x coordinate");
ylabel("z coordinate");

point1 = plot(w1(1), w1(2) ,'ro');
point2 = plot(w2(1), w2(2) ,'r*');
point3 = plot(w3(1), w3(2) ,'rs');
point4 = plot(w4(1), w4(2) ,'rd');
point_start = plot(points_x(1), points_y(1), 'black*');
rocket_trajectory = plot(points_x, points_y, '-black');
legend([point1 point2 point3 point4 point_start
 rocket_trajectory],'waypoint 1','waypoint 2','waypoint 3','waypoint
 4', 'starting point', 'rocket trajecotry','Location','southwest');
hold off;

% draw plot of enigines trust over the time
figure(5);
hold on; grid on;
```

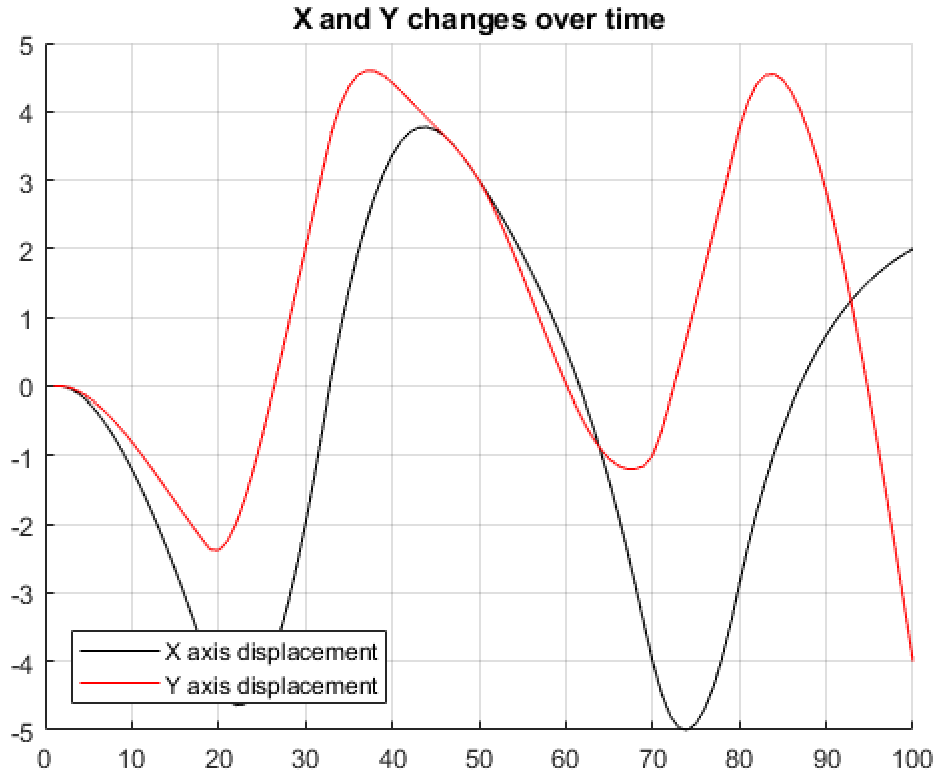
```
title('Rocket engines forces');
engine1 = plot(engine1_power, 'black-');
engine2 = plot(engine2_power, 'red-');
legend([engine1 engine2],'engine 1','engine
 2','Location','Northwest');
hold off

% Draw plot of  the X and Y position
figure(6);
hold on; grid on;
title('X and Y changes over time');
points_x = plot (points_x, 'black-');
points_y = plot(points_y, 'red-');
legend([points_x points_y],'X axis displacement','Y axis
 displacement','Location','Southwest');
hold off
```

```
Warning: The use of strict inequalities in CVX is strongly
 discouraged,
    because solvers treat them as non-strict inequalities. Please
    consider using "<=" instead.

Warning: The use of strict inequalities in CVX is strongly
 discouraged,
    because solvers treat them as non-strict inequalities. Please
    consider using "<=" instead.


Calling SDPT3 4.0: 998 variables, 606 equality constraints
------------------------------------------------------------

 num. of constraints = 606
 dim. of socp   var  = 400,   num. of socp blk  = 200
 dim. of linear var  = 596
 dim. of free   var  =  2 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-------------------------------------------------------------------
 0|0.000|0.000|9.3e+01|4.3e+01|1.3e+07| 1.786164e+05  0.000000e+00|
 0:0:00| spchol  1  1
 1|0.872|1.000|1.2e+01|5.0e-01|2.0e+06| 2.376211e+04 -1.281282e+05|
 0:0:00| spchol  1  1
 2|0.938|1.000|7.3e-01|1.5e-01|2.1e+05| 2.561084e+03 -1.072436e+05|
 0:0:00| spchol  1  1
 3|0.718|0.998|2.1e-01|1.5e-02|6.1e+04| 1.555192e+03 -4.199545e+04|
 0:0:00| spchol  1  1
 4|0.687|0.778|6.5e-02|4.5e-03|2.0e+04| 1.274367e+03 -1.547453e+04|
 0:0:00| spchol  1  1
 5|0.489|0.736|3.3e-02|1.4e-02|1.1e+04| 1.204352e+03 -7.994095e+03|
 0:0:00| spchol  1  1 
 6|0.756|0.527|8.1e-03|1.4e-02|6.9e+03| 1.138706e+03 -5.355988e+03|
 0:0:00| spchol  1  1 
 7|0.761|0.457|1.9e-03|9.0e-03|4.6e+03| 1.113686e+03 -3.392635e+03|
 0:0:00| spchol  1  1 
 8|1.000|0.957|2.8e-07|7.7e-04|5.9e+02| 1.092380e+03  5.070567e+02|
 0:0:00| spchol  2  2 
 9|1.000|0.950|8.5e-08|3.9e-05|2.2e+02| 1.068731e+03  8.494662e+02|
 0:0:00| spchol  2  2 
10|1.000|0.940|1.1e-07|2.3e-06|4.2e+01| 1.061178e+03  1.018998e+03|
 0:0:00| spchol  2  2 
11|0.718|0.941|5.1e-08|1.6e-07|2.0e+01| 1.059604e+03  1.039687e+03|
 0:0:00| spchol  2  2 
12|0.906|0.847|1.4e-08|3.7e-08|8.7e+00| 1.058483e+03  1.049811e+03|
 0:0:00| spchol  2  2 
13|0.171|0.434|9.6e-09|2.4e-08|8.0e+00| 1.058120e+03  1.050078e+03|
 0:0:00| spchol  2  2 
14|0.886|0.604|2.7e-09|1.2e-08|4.4e+00| 1.057683e+03  1.053313e+03|
 0:0:00| spchol  2  2 
15|1.000|0.952|4.5e-10|1.4e-09|2.2e+00| 1.057451e+03  1.055234e+03|
 0:0:00| spchol  1  1 
16|1.000|0.901|1.5e-11|3.4e-10|5.7e-01| 1.057237e+03  1.056665e+03|
 0:0:00| spchol  1  1 
17|0.927|0.983|8.1e-12|7.1e-11|1.4e-02| 1.057174e+03  1.057160e+03|
 0:0:00| spchol  1  1 
18|0.983|0.982|1.6e-11|2.1e-10|2.9e-04| 1.057170e+03  1.057169e+03|
 0:0:00| spchol  1  1 
19|0.989|0.989|8.0e-12|2.1e-11|6.1e-06| 1.057170e+03  1.057170e+03|
 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 19
 primal objective value =  1.05716958e+03
 dual   objective value =  1.05716957e+03
 gap := trace(XZ)       = 6.14e-06
 relative gap           = 2.90e-09
 actual relative gap    = 2.62e-09
 rel. primal infeas (scaled problem)   = 8.03e-12
 rel. dual     "        "       "      = 2.09e-11
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 2.7e+02, 1.2e+02, 5.2e+00
 norm(A), norm(b), norm(C) = 1.8e+02, 3.0e+02, 1.5e+01
 Total CPU time (secs)  = 0.25  
 CPU time per iteration = 0.01  
 termination code       =  0
 DIMACS: 6.0e-11  0.0e+00  1.6e-10  0.0e+00  2.6e-09  2.9e-09
-------------------------------------------------------------------
 
------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +1057.17
```

```
Found minimal fuel consumption:
   1.0572e+03
```




Rocket engines forces
engine 1
engine 2
20
18
16
14
12
10
8
6
4
2
0
0
10
20
30
40
50
60
70
80
90
100


X and Y changes over time
X axis displacement
Y axis displacement
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
10
20
30
40
50
60
70
80
90
100

Part 3: Trust less than 15

```
cvx_solver('sdpt3');
cvx_begin
variables engine1_power(K-1) engine2_power(K-1) force_vector_x(K) force_vector_y(K
minimize (sum(engine1_power)+sum(engine2_power))
subject to
    % where rocket starts
    points_x(1) == 0;
    points_y(1) == 0;
    % rocket wasn't moving on the start
    velocity_x(1) == 0;
    velocity_y(1) == 0;

    points_x(k1) == w1(1);
    points_y(k1) == w1(2);

    points_x(k2) == w2(1);
    points_y(k2) == w2(2);

    points_x(k3) == w3(1);
    points_y(k3) == w3(2);

    points_x(k4) == w4(1);
    points_y(k4) == w4(2);

    % engine power must be grather than 0!
    engine1_power >= 0; engine2_power >= 0;
    engine1_power < 15; engine2_power < 15;
    % bounds of the trajecory
    abs(points_x) <= pmax; abs(points_y) <= pmax;

    for k=1:K-1
     % Rocket model
     force_vector_x(k)== cos(theta1)*engine1_power(k) +
 cos(theta2)*engine2_power(k) + 0;
     force_vector_y(k)== sin(theta1)*engine1_power(k) +
 sin(theta2)*engine2_power(k) - m*g;

     velocity_x(k+1)==(1-alpha)*velocity_x(k) + (h/
m)*force_vector_x(k);
     velocity_y(k+1)==(1-alpha)*velocity_y(k) + (h/
m)*force_vector_y(k);

     points_x(k+1) == points_x(k) + h*velocity_x(k);
     points_y(k+1) == points_y(k) + h*velocity_y(k);
    end
cvx_end

total_fuel_consumption = (sum(engine1_power)+sum(engine2_power));
disp('Found minimal fuel consumption: ')
disp(total_fuel_consumption)
```

```
% In that case, the result should be false because we can't find it
 for
% this constrains (less than 15)

Warning: The use of strict inequalities in CVX is strongly
 discouraged,
    because solvers treat them as non-strict inequalities. Please
    consider using "<=" instead.

Warning: The use of strict inequalities in CVX is strongly
 discouraged,
    because solvers treat them as non-strict inequalities. Please
    consider using "<=" instead.


Calling SDPT3 4.0: 998 variables, 606 equality constraints
------------------------------------------------------------

 num. of constraints = 606
 dim. of socp   var  = 400,   num. of socp blk  = 200
 dim. of linear var  = 596
 dim. of free   var  =  2 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj
 cputime
-------------------------------------------------------------------
 0|0.000|0.000|1.2e+02|4.3e+01|1.3e+07| 1.786164e+05  0.000000e+00|
 0:0:00| spchol  1  1
 1|0.849|1.000|1.8e+01|5.0e-01|2.4e+06| 2.785466e+04 -1.058592e+05|
 0:0:00| spchol  1  1
 2|0.946|1.000|9.7e-01|1.5e-01|2.1e+05| 2.606972e+03 -9.452570e+04|
 0:0:00| spchol  1  1
 3|0.709|1.000|2.8e-01|1.5e-02|7.0e+04| 1.573383e+03 -4.636439e+04|
 0:0:00| spchol  1  1
 4|0.591|0.761|1.2e-01|4.7e-03|2.9e+04| 1.317431e+03 -2.107729e+04|
 0:0:00| spchol  1  1
 5|0.376|0.706|7.2e-02|1.7e-03|1.8e+04| 1.248321e+03 -1.290063e+04|
 0:0:00| spchol  1  1
 6|0.530|0.475|3.4e-02|1.5e-02|1.3e+04| 1.183609e+03 -9.722901e+03|
 0:0:00| spchol  1  1
 7|0.342|0.833|2.2e-02|9.4e-03|8.3e+03| 1.162200e+03 -4.797431e+03|
 0:0:00| spchol  2  2
 8|0.236|0.501|1.7e-02|9.2e-03|7.4e+03| 1.150760e+03 -3.088114e+03|
 0:0:00| spchol  2  2
 9|0.355|0.580|1.1e-02|7.3e-03|5.6e+03| 1.136231e+03  1.326401e+03|
 0:0:00| spchol  2  2
10|0.170|0.052|9.1e-03|9.1e-03|4.8e+03| 1.131223e+03  6.309853e+03|
 0:0:00| spchol  2  2
11|0.037|0.905|8.8e-03|2.7e-03|6.5e+04| 1.130272e+03  2.423033e+06|
 0:0:00| spchol  2  3
```

```
12|0.381|0.382|5.5e-03|3.4e-03|7.8e+08| 1.196691e+03  1.175159e+09|
 0:0:00| spchol  3  3
13|0.001|0.933|5.4e-03|1.3e-03|4.6e+09| 1.111708e+03  6.798526e+09|
 0:0:00| spchol  2  2
14|0.232|0.932|4.2e-03|1.2e-03|1.2e+10| 1.187193e+03  3.965130e+10|
 0:0:00| spchol  2  2
15|0.208|0.918|3.3e-03|1.9e-03|2.0e+10| 1.151780e+03  7.457735e+11|
 0:0:00| spchol  3  2
16|0.782|0.902|8.6e-04|1.6e+00|1.5e+15| 1.105180e+03  9.219602e+14|
 0:0:00| spchol  3  3
17|0.001|0.905|8.6e-04|3.8e+00|3.7e+15| 1.181947e+03  2.227928e+15|
 0:0:00| spchol  2  2
18|0.201|0.905|6.9e-04|1.2e+01|6.4e+15| 1.103024e+03  5.157799e+15|
 0:0:00| spchol  2  2
19|0.389|0.835|4.2e-04|4.6e+01|5.7e+15| 1.138442e+03  1.721931e+16|
 0:0:00| spchol  2  2
20|0.176|0.918|3.5e-04|9.2e+02|2.0e+16| 1.138697e+03  3.374393e+17|
 0:0:00|
  sqlp stop: primal problem is suspected of being infeasible
-------------------------------------------------------------------
 number of iterations   = 20
 residual of primal infeasibility
 certificate (y,Z)      = 4.13e-14
 reldist to infeas.    <= 6.31e-18
 Total CPU time (secs)  = 0.22
 CPU time per iteration = 0.01
 termination code       =  1
 DIMACS: 2.0e-03  0.0e+00  7.0e+03  0.0e+00  -1.0e+00  6.0e-02
-------------------------------------------------------------------

------------------------------------------------------------
Status: Infeasible
Optimal value (cvx_optval): +Inf

Found minimal fuel consumption:
   NaN
```

Published with MATLAB® R2017b