EE290N Final Project
Robust approach to force/torque generation problem

Pablo Anigstein

1 Introduction

Many spacecraft have an Orbit Control system which applies forces in order to maintain the orbital parameters within certain bounds. In general orbit control is used to increase altitude and adjust the orientation of the orbital plane. This is in most cases performed by means of thrusters which produce a force in a certain direction fixed to the spacecraft body. At the same time, the Attitude Control system is in charge of maintaining the desired orientation of the spacecraft. A commonly used approach is to use the same thrusters to control orbit and attitude. The resulting control scheme is as follows: at each control sampling time, a desired control force is computed by the orbit control system and a desired control torque is computed by the attitude control system. We are faced with the following problem: given desired force and torque to be applied to the spacecraft, compute the force to be produced by each thruster in some optimal way.

In this project, I propose an SOCP optimization problem to address this thrust computation problem. Then I consider uncertainty in the parameters and obtain a new problem to address robustness. I get the LFR representation of the uncertainty and obtain an SDP which approximates the robust SOCP. Finally I implemented both the original nonrobust and the approximation to the robust optimization problems in MATLAB using lmitools. I tested both on some simple examples, verifying the robust problem solution has the desired robustness property.

2 Problem Statement

We model the spacecraft as a rigid body. The thrusters configuration consists of a number of thrusters located at certain positions and orientations with respect to a frame fixed to the body with origin at the center of mass. The thrusters will in general exert both forces and torques to the body.

Define:

$$\begin{aligned} N &= \text{number of thrusters} \\ p_i &= \text{position of thruster } i \text{ (in body frame) } (p_i \in \mathbb{R}^3) \end{aligned}$$

$$
\begin{array}{rcl}
d_i & = & \text{direction of force produced by thruster } i \text{ (in body frame) } (d_i \in \mathbf{R}^3) \\
u_i & = & \text{magnitude of force of thruster } i \ (u \in \mathbf{R}^N) \\
f & = & \text{resulting total force on spacecraft (in body frame) } (f \in \mathbf{R}^3) \\
t & = & \text{resulting total torque on spacecraft (in body frame) } (t \in \mathbf{R}^3)
\end{array}
$$

We have N thrusters. Thruster i is located at a fixed position p_i and produces a force in a fixed direction d_i in body frame. The force produced by thruster i is then computed as $u_i d_i$, and the torque is $u_i p_i \times d_i$. Thus the total force and torque on the spacecraft are:

$$
\begin{aligned}
f &= \sum_{i=1}^{N} u_i d_i = Du \\
t &= \sum_{i=1}^{N} u_i p_i \times d_i = Bu
\end{aligned}
$$

where we have defined

$$
\begin{aligned}
D &= \begin{bmatrix} d_1 & \dots & d_N \end{bmatrix} \\
B &= \begin{bmatrix} p_1 \times d_1 & \dots & p_N \times d_N \end{bmatrix}
\end{aligned} \tag{1}
$$

We can control the magnitude of the force produced by thruster i through the variable u_i. The thrusters can only make force in the positive d_i direction, i.e. $u_i \geq 0$. Without loss of generality we will assume each thruster has the same maximum force capability $u_i \leq u_m$ (we could include different scale factors in the vectors d_i). So we have a linear constraint on u: $\mathbf{0} \preceq u \preceq u_m \mathbf{1}$, where $\mathbf{0} \in \mathbf{R}^N$ is the zero vector and $\mathbf{1} \in \mathbf{R}^N$ has all components equal to 1.

3 Thrusts Computation Problem

Thrusts computation problem is as follows: we are given the thruster configuration, i.e. the number of thrusters N, matrices D and B, and the maximum allowed thruster force u_m. We are also given a desired torque $t_d \in \mathbf{R}^3$ computed by the attitude control algorithm.

A possible formulation is to assume that a desired force f_d is given by the orbit control system. In this case, we can for example pose a fuel usage minimization problem. The fuel usage is proportional to $\mathbf{1}^T u$, so we could compute the optimal thrust u that minimizes the fuel usage while producing the desired force and torque as the following LP:

$$
\begin{array}{ll}
\text{minimize} & \mathbf{1}^T u \\
\text{subject to} & Du = f_d \\
& Bu = t_d \\
& \mathbf{0} \preceq u \preceq u_m \mathbf{1}
\end{array} \tag{2}
$$

In many cases the above is not a good formulation mainly due to two reasons:

1. It can be the case that due to the orbit maneuvers to be performed, the direction of the desired force in body frame is restricted to a certain set. As an extreme example, in many Earth-observation satellites orbit maneuvers are only performed to increase the orbit altitude. For such maneuvers, f_d is along the velocity of the satellite, which due to the satellite attitude is in a fixed direction in body frame. In such cases, it makes no sense to have a thruster configuration capable of producing force in all directions. The resulting thruster configuration designed may not have enough degrees of freedom to exactly satisfy the constraints $Du = f_d$ and $Bu = t_d$ simultaneously.

2. The nature of orbit maneuvers is usually impulsive: from the orbit control point of view we would like to apply an impulse at a given time. This means that we have a desired direction for the force, but the larger the magnitude in that direction the better, because a larger force means a shorter duration of the maneuver.

A formulation that includes these cases is as follows. Instead of a desired force, we are given a desired direction for the force $c \in \mathbb{R}^3$ ($\|c\| = 1$). The new objective is to minimize a weighted difference of the fuel usage and the resulting force projection in the given desired direction. We want to keep the undesired force (i.e. the projection of the force f on the plane normal to c) smaller than a given fraction of the desired force (i.e. the projection of f on c). We still require $Bu = t_d$. The resulting problem is the following SOCP:

$$\begin{array}{ll} \text{minimize} & \gamma \mathbf{1}^T u - c^T D u \\ \text{subject to} & \|(I - cc^T) D u\| \leq \alpha c^T D u \\ & Bu = t_d \\ & \mathbf{0} \preceq u \preceq u_m \mathbf{1} \end{array} \tag{3}$$

4 Uncertainty in the Model

We have considered up to now the ideal case where D and B are perfectly known. These matrices will have uncertainty due to several reasons. Recall from (1) that D and B are defined in terms of the positions p_i and orientations d_i. In practice, p_i and d_i are only known within certain errors:

- Recall that p_i is the position of thruster i with respect to a frame with origin in the center of mass of the spacecraft. The positions are known to a certain precision. Besides, the center of mass moves during the lifetime of the spacecraft: the mass distribution onboard the spacecraft varies due to the decrease of fuel (and in some cases other reasons).

- The orientation of the thrusters d_i is knwon to a certain precision. The fuel can be ejected shifted with respect to the axis of symmetry of the nozzle. Scale factor errors (i.e. we apply u_i but the resulting force magnitude can have a scale factor different from one, i.e. $\|d_i\| \neq 1$) are also to be included in d_i.

In what follows, I will consider the case when p_i are known exactly (it is simple to consider also uncertainty in p_i, I won't for brevity). I will make the assumption that each coordinate of each d_i have a bounded error independent from each other[1]:

$$d_i = d_i^0 + \delta_{di}$$

with $|\delta_{dij}| \leq \bar{\delta}_{ij}$ where $\bar{\delta}_{ij}$ are given uncertainty bounds. This is equivalent to assume that the uncertainty set for D (call it $\mathbf{D}$) is a matrix interval set. We can write $D = D^0 + \Delta_D$ where D^0 corresponds to the nominal parameters and $\Delta_D \in \mathbf{\Delta_D}$ where $\mathbf{\Delta_D} = \{\Delta_D \mid |\delta_{dij}| \leq \bar{\delta}_{ij}\ \}$ with $\bar{\delta}_{ij}$ given bounds. To obtain the resulting uncertainty set for B we can write $B = B^0 + \Delta_B$ where

$$\begin{aligned} B^0 &= \left[\begin{array}{ccc} p_1 \times d_1^0 & \ldots & p_N \times d_N^0 \end{array}\right] \\ \Delta_B &= \left[\begin{array}{ccc} p_1 \times \delta_{d1} & \ldots & p_N \times \delta_{dN} \end{array}\right] \end{aligned} \tag{4}$$

5 Robust Thrusts Computation

I will now consider the robust optimization problem arising when matrices D and B which define our configuration are uncertain. Assume that we are given uncertainty sets (as in the previous section) such that $D \in \mathbf{D}$ and $B \in \mathbf{B}$. Note that now it makes no sense to require that $Bu = t_d$ because B is uncertain. This constraint will thus be replaced by a robustness condition $\|Bu - t_d\| \leq \varepsilon \quad \forall B \in \mathbf{B}$, where ε is a given maximum acceptable error in the resulting torque. Then we can pose the robust version of (3) as:

$$\begin{array}{ll} \text{minimize} & \max_{D \in \mathbf{D}} (\gamma \mathbf{1}^T u - c^T D u) \\ \text{subject to} & \|(I - cc^T) D u\| \leq \alpha c^T D u \quad \forall D \in \mathbf{D} \\ & \|Bu - t_d\| \leq \varepsilon \quad \forall B \in \mathbf{B} \\ & \mathbf{0} \preceq u \preceq u_m \mathbf{1} \end{array} \tag{5}$$

which can be rewritten as:

$$\begin{array}{lll} \text{minimize} & v & \text{(over } u, v) \\ \text{subject to} & (\gamma \mathbf{1}^T u - c^T D u) \leq v \quad \forall D \in \mathbf{D} & \\ & \|(I - cc^T) D u\| \leq \alpha c^T D u \quad \forall D \in \mathbf{D} & \\ & \|Bu - t_d\| \leq \varepsilon \quad \forall B \in \mathbf{B} & \\ & \mathbf{0} \preceq u \preceq u_m \mathbf{1} & \end{array} \tag{6}$$

[1] It would be more accurate to consider independent scale factor error and angle error bounds in d_i. This would yield less conservative results, but would make the derivation of the robust approximation to our problem longer

As (6) is a robust SOCP, thus a robust SDP, we can try to get the LFR representation of the uncertainty in the constraints and then via a relaxation obtain a "tractable" SDP.

I will analyze each of the three robust constraints in (6) separately. The first constraint in (6) can be written as:

$$v - \gamma \mathbf{1}^T u + c^T D^0 u + c^T \Delta_D u \geq 0 \quad \forall \Delta_D \in \mathbf{\Delta_D} \tag{7}$$

Note that $c^T \Delta_D u = \sum_{i=1}^{3} \sum_{j=1}^{N} c_i u_j \delta_{dij}$. Thus, as $\delta_{dij} \leq \bar{\delta}_{ij}$, (7) is equivalent to

$$v - \gamma \mathbf{1}^T u + c^T D^0 u - \sum_{i=1}^{3} \sum_{j=1}^{N} |c_i u_j| \bar{\delta}_{ij} \geq 0$$

This in turn is equivalent to the existence of $S_1 \in \mathrm{R}^{3 \times N}$ such that:

$$\begin{array}{c} -c_i u_j \leq s_{1ij} \leq c_i u_j \quad \forall i = 1, \ldots, 3 \ , \ j = 1, \ldots, N \\ v - \gamma \mathbf{1}^T u + c^T D^0 u - \sum_{i=1}^{3} \sum_{j=1}^{N} \bar{\delta}_{ij} s_{1ij} \geq 0 \end{array} \tag{8}$$

The second constraint in (6) can be written as a robust LMI:

$$\mathbf{F_2}(u, \Delta_2) = \begin{bmatrix} \alpha c^T D u & \left[(I - cc^T) D u \right]^T \\ (I - cc^T) D u & (\alpha c^T D u) I \end{bmatrix} \succeq 0 \quad \forall D \in \mathbf{D} \tag{9}$$

I will now compute an LFR of $\mathbf{F_2}(u, \Delta_2)$. As $\mathbf{F_2}$ is affine in Δ_D we can write

$$\mathbf{F_2}(u, \Delta_2) = F_2(u) + L_2(u) \Delta_2 R_2 + (L_2(u) \Delta_2 R_2)^T$$

with

$$\begin{array}{rcl} F_2(u) & = & \begin{bmatrix} \alpha c^T D^0 u & \left[(I - cc^T) D^0 u \right]^T \\ (I - cc^T) D^0 u & (\alpha c^T D^0 u) I \end{bmatrix} \\ L_2(u) \Delta_2 R_2 & = & \begin{bmatrix} \frac{1}{2} \alpha c^T \Delta_D u & \mathbf{0} \\ (I - cc^T) \Delta_D u & \frac{1}{2} (\alpha c^T \Delta_D u) I \end{bmatrix} \end{array} \tag{10}$$

From the above we get:

$$\begin{array}{rcl} L_2(u) \Delta_2 R_2 & = & \sum_{i=1}^{3} \sum_{j=1}^{N} \delta_{dij} M_{ij}(u) \\ M_{ij}(u) & = & \begin{bmatrix} \frac{1}{2} \alpha u_j c_i & 0 \quad 0 \quad 0 \\ (e_{1i} - c_1 c_i) u_j & \\ (e_{2i} - c_2 c_i) u_j & \frac{1}{2} \alpha u_j c_i I \\ (e_{3i} - c_3 c_i) u_j & \end{bmatrix} \end{array} \tag{11}$$

where $e_{ij} = 1$ for $i = j$, $e_{ij} = 0$ for $i \neq j$. So we can take:

$$\begin{aligned} L_2(u) &= \begin{bmatrix} M_{11}(u) & \dots & M_{3N}(u) \end{bmatrix} && (12) \\ R_2 &= \begin{bmatrix} \bar{\delta}_{11} I_{4\times 4} \\ \vdots \\ \bar{\delta}_{3N} I_{4\times 4} \end{bmatrix} && (13) \\ \Delta_2 &= \mathbf{diag}\left(\frac{\delta_{d11}}{\bar{\delta}_{11}} I_{4\times 4}, \dots, \frac{\delta_{d3N}}{\bar{\delta}_{3N}} I_{4\times 4} \right) && (14) \end{aligned}$$

I have taken a normalized representation so that the uncertainty set is

$$\boldsymbol{\Delta_2} = \{ \mathbf{diag}\left(\delta_1 I_{4\times 4}, \dots, \delta_{3N} I_{4\times 4} \right) \mid |\delta_i| \leq 1 \quad \forall i = 1, \dots, 3N \}$$

$\mathbf{F_2}(u, \Delta_2)$ can be written as:

$$\begin{aligned} \mathbf{F_2}(u, \Delta_2) &= F_2(u) - L_2(u)(L_2(u))^T - R_2^T R_2 \\ &\quad + \begin{bmatrix} L_2(u) & (R_2)^T \end{bmatrix} \begin{bmatrix} I & \Delta_2 \\ \Delta_2^T & I \end{bmatrix} \begin{bmatrix} (L_2(u))^T \\ R_2 \end{bmatrix} \end{aligned}$$

and as $\|\Delta_2\| \leq 1$ we see that a sufficient condition for our second robustness constraint to hold is:

$$F_2(u) \succeq L_2(u)(L_2(u))^T + R_2^T R_2$$

The LFR of $\mathbf{F_2}$ is invariant to scalings that are block diagonal. So we get a sufficient robustness condition: $\exists S_2 \in \mathbb{R}^{12N \times 12N}$ symmetric and block diagonal with blocks of size 4×4,

$$\begin{bmatrix} F_2(u) - R_2^T S_2 R_2 & L_2(u) \\ (L_2(u))^T & S_2 \end{bmatrix} \succeq 0 \tag{15}$$

Now consider the third constraint in (6), which can be rewritten as a robust LMI:

$$\mathbf{F_3}(u, \Delta_3) = \begin{bmatrix} \varepsilon^2 & (Bu - t_d)^T \\ Bu - t_d & I \end{bmatrix} \succeq 0 \quad \forall B \in \mathbf{B} \tag{16}$$

We have $\mathbf{F_3}(u, \Delta_3) = F_3(u) + L_3(u)\Delta_3 R_3 + (L_3(u)\Delta_3 R_3)^T$ with

$$\begin{aligned} F_3(u) &= \begin{bmatrix} \varepsilon^2 & (B^0 u - t_d)^T \\ B^0 u - t_d & I \end{bmatrix} && (17) \\ L_3(u)\Delta_3 R_3 &= \begin{bmatrix} 0 & \mathbf{0} \\ \Delta_B u & \mathbf{0} \end{bmatrix} \end{aligned}$$

Using (4) we can obtain:

$$L_3(u) = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ u_1 \hat{p}_1 & \dots & u_N \hat{p}_N \end{bmatrix} \tag{18}$$

$$
R_3 = \begin{bmatrix} \bar{\delta}_{11} & 0 & 0 & 0 \\ \bar{\delta}_{21} & 0 & 0 & 0 \\ \vdots & & & \\ \bar{\delta}_{2N} & 0 & 0 & 0 \\ \bar{\delta}_{3N} & 0 & 0 & 0 \end{bmatrix} \tag{19}
$$

$$
\Delta_3 = \mathbf{diag}\left(\frac{\delta_{d11}}{\bar{\delta}_{11}}, \frac{\delta_{d21}}{\bar{\delta}_{21}}, \ldots, \frac{\delta_{d2N}}{\bar{\delta}_{2N}}, \frac{\delta_{d3N}}{\bar{\delta}_{3N}}\right) \tag{20}
$$

where $\hat{p}_j$ is the matrix corresponding to the cross product by p_j:

$$
\hat{p}_j = \begin{bmatrix} 0 & -(p_j)_z & (p_j)_y \\ (p_j)_z & 0 & -(p_j)_x \\ -(p_j)_y & (p_j)_x & 0 \end{bmatrix} \quad j = 1, \ldots, N
$$

So a sufficient robustness condition for our third constraint is: $\exists S_3 \in \mathbb{R}^{3N \times 3N}$ diagonal,

$$
\begin{bmatrix} F_3(u) - R_3^T S_3 R_3 & L_3(u) \\ (L_3(u))^T & S_3 \end{bmatrix} \succeq 0 \tag{21}
$$

The SDP approximation to our robust SOCP (6) is thus:

$$
\begin{array}{ll}
\text{minimize} & v \qquad\qquad\qquad\qquad (\text{over } u, v, S_1, S_2, S_3) \\
\text{subject to} & -c_i u_j \leq s_{1ij} \leq c_i u_j \quad \forall i = 1, \ldots, 3 \ , \ j = 1, \ldots, N \\
& v - \gamma \mathbf{1}^T u + c^T D^0 u - \sum_{i=1}^{3} \sum_{j=1}^{N} \bar{\delta}_{ij} s_{1ij} \geq 0 \\
& S_2 = S_2^T \\
& S_2 \text{ block diagonal (blocks of size } 4 \times 4), \ S_3 \text{ diagonal} \\
& \begin{bmatrix} F_2(u) - R_2^T S_2 R_2 & L_2(u) \\ (L_2(u))^T & S_2 \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} F_3(u) - R_3^T S_3 R_3 & L_3(u) \\ (L_3(u))^T & S_3 \end{bmatrix} \succeq 0 \\
& \mathbf{0} \preceq u \preceq u_m \mathbf{1}
\end{array} \tag{22}
$$

The number of variables for the new problem (after eliminating unnecessary ones) is: $N(u) + 1(v) + 3N(S_1) + 30N(S_2) + 3N(S_3) \sim 37N$. We have $8N + 1$ linear constraints, and 2 LMIs of sizes $12N + 4$ and $3N + 4$.

6 Example

I implemented both the original nonrobust problem (3) and the robust SDP approximation (22) using Matlab and lmitools (I enclose Matlab functions listings).

I also wrote three simple functions that take as input the nominal configuration (D^0, B^0, u_m), parameters $(\gamma, \alpha, \varepsilon)$, desired torque and force direction (t_d, c), the modeling error bounds $(\bar{\delta}_{ij})$ and the thrusts (u), and exactly compute:

- The worst case objective:

$$\max_{D \in \mathbf{D}} (\gamma \mathbf{1}^T u - c^T D u)$$

- The worst case difference between nondesired force and desired force (weighted by α):

$$\max_{D \in \mathbf{D}} (\|(I - cc^T) D u\| - \alpha c^T D u)$$

- The worst case error torque:

$$\max_{B \in \mathbf{B}} \|B u - t_d\|$$

Note that the first case is an LP, and in the other two cases we are maximizing a convex function over a polytope. For the sake of testing in simple examples with a small number of thrusters, I just compute the maximum evaluating in the polytope vertices (there are 2^{3N}).

As an example, I take the following simple configuration of four thrusters:

$$\begin{aligned}
\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix} &= \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \\
D^0 &= \begin{bmatrix} -0.1736 & -0.1736 & 0.1736 & 0.1736 \\ 0 & 0 & 0 & 0 \\ 0.9848 & 0.9848 & 0.9848 & 0.9848 \end{bmatrix} \\
\bar{\delta}_{ij} &= 0.01 \quad i = 1, \ldots, 3, \; j = 1, \ldots, 4 \\
u_m &= 1
\end{aligned}$$

and for the optimization I take $\gamma = 0$ and $\alpha = 0.1$ (i.e. maximum nondesired force is 10% of desired force). The desired force direction is $c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. I computed optimal points u of (3) and (22) for several values of desired torque t_d: $t_{dx} = t_{dz} = 0$ and t_{dy} ranging from 0 to 1.2. Note that it is reasonable to take a bound on the torque error ε in (22) to increase when $\|t_d\|$ grows. For each t_d I took $\varepsilon = 0.07 + 0.1\|t_d\|$.

I present the results in three graphs. Figure 1 shows the worst case error torque that results for the solution of (3) and (22) as a function of t_{dy}. Also plotted is the maximum acceptable error ε. In figure 2 I plotted the worst case objective obtained with the solution of each problem. In figure 3 we see the worst case of the difference $\|(I - cc^T) D u\| - \alpha c^T D u$.

We observe that the robust formulation works as expected. For low values of t_{dy} (up to approx. 0.2) the nonrobust solution gives a worst case error torque that is above the maximum allowed (ε). Note that the robust solution results in a worst case error torque that is always below ε (dotted line in figure 1), and it is not very far below (i.e. the approximation of the constraint is not very conservative). In figure 2 we see that for values of t_{dy} above approx. 0.2 (for which the nonrobust solution results in an acceptable torque error), we observe that the robust solution obtains a smaller (i.e. better) worst case objective. In figure 3 observe that for values of t_{dy} approx. between 1.05 and 1.2 the nonrobust solution results in an unacceptable worst case nondesired force to desired force ratio (i.e. the plotted curve is above zero) , while solution to our robust formulation still satisfies the constraint robustly. Also note that there is a small interval around $t_{dy} = 0.2$ where the nonrobust solution performs slightly better, i.e. it yields a robustly feasible point of (6) which obtains a slightly smaller worst case objective than the solution to our robust SDP approximation (22). This is to be expected, because (22) is a conservative approximation to (6). But we see that it is not too conservative.

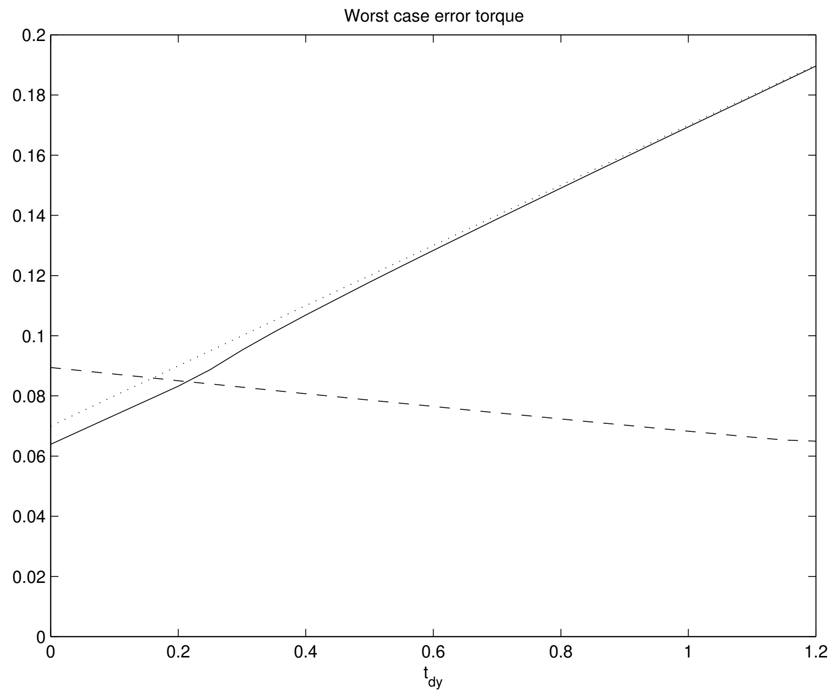

Figure 1: Solid: Robust SDP solution. Dashed: Nonrobust solution. Dotted: Maximum accepted error (ε,robust problem)

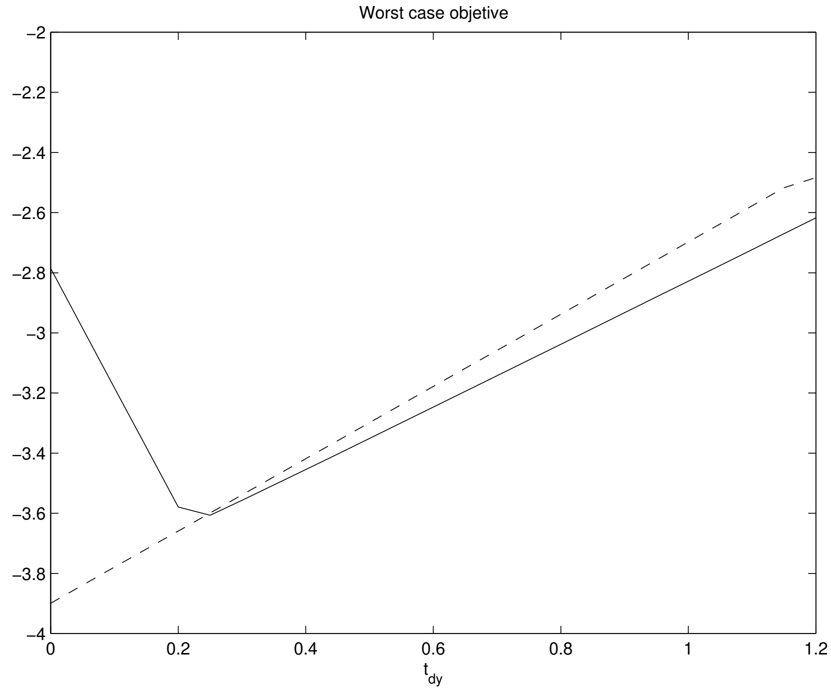

Figure 2: Solid: Robust SDP solution. Dashed: Nonrobust solution.

7 Conclusions

We have posed a thrust computation problem as an SOCP (3). We have considered uncertainty in the directions of the thrusters, posing a new, robust SOCP problem. We have used the LFR representation and a relaxation to obtain an SDP approximation (22) which yields a bound and a robust solution to our problem. We have compared our robust SDP solution to the solution yielded by our original nonrobust problem in a simple example.

As a final remark, note that in practice our problem would have to be solved in real time once every control sampling time, which would typically be of the order of one second. This imposes a limit to the applicability of our SDP problem (22) as is. Further work could be trying to obtain a "smaller sized" (though probably not so good) approximation to the robust problem (6).

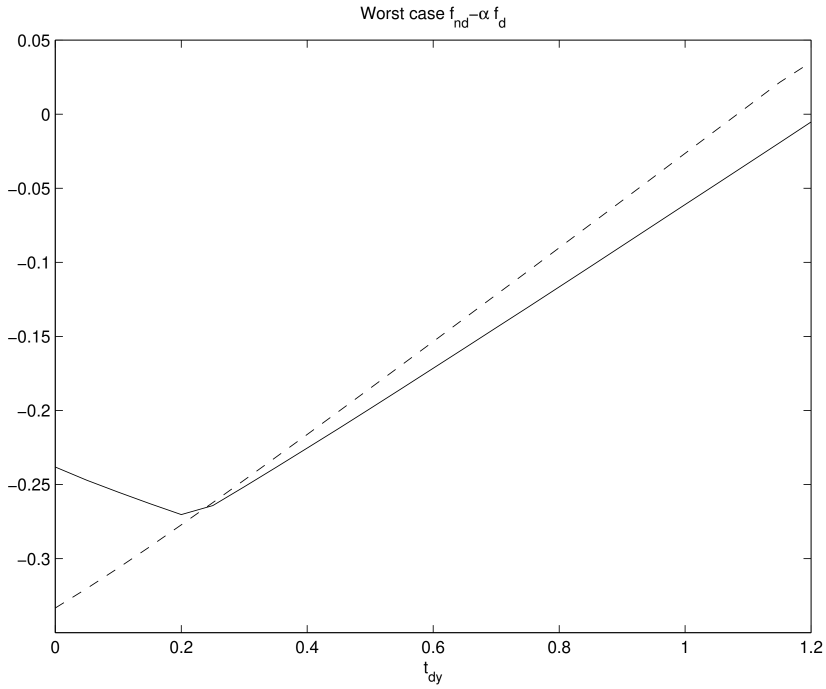

Figure 3: Solid: Robust SDP solution. Dashed: Nonrobust solution.