Robust Satellite Attitude Trajectory Optimization under Thruster Uncertainty

Keenan Albee and Alex Steighner

***Abstract*— Dynamic systems operating with uncertainty are everywhere when dealing with physical hardware. Modeling errors, hardware defects, and environmental disturbances are just a few factors contributing to deviation from deterministic system models. As such, these systems must incorporate models of uncertainty in order to operate effectively. Satellite systems are no exception–in some cases, satellites may have poor models of thruster performance that are better modeled using uncertainty than a deterministic model. A particular class of systems has an efficient solution for control of such systems, called the linear quadratic regulator (LQR), but this optimization is only in the sense of expected value of the resulting cost. One may desire, for instance, to provide guarantees on cost performance in the face of noise in the dynamics. This report implements such a scheme, using Bertsimas' et al.'s robust optimization solution to the problem. A desired use case, linear satellite dynamics with Gaussian process noise, is demonstrated using two approaches and compared against the commonly-implemented LQR solution.**

I. BACKGROUND

Satellites operate in a six degree of freedom environment to carry out missions on-orbit. Before launch, inertial and thruster properties are characterized to develop control laws that dictate how to move the satellite into certain desirable orientations. There are situations where the thruster properties change due to wear and tear on-orbit or other unforeseen circumstances. An example of this type of situation is for the on-orbit experiment SPHERES being flown by MIT's Space Systems Lab. After over a decade of use, thruster degradation has led to inaccuracies in the original model. Adding uncertainty to the control mode will aid in making the control system robust, and lead to better performance than optimizing for the incorrect deterministic model.

Optimization enters problems related to controlling dynamical systems when the desired system behavior can be framed as the minimization of some objective function. The aim of optimal control is to manipulate such a dynamical system described by a state vector $\mathbf{x}$ subject to constraints of the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, such that some cost (objective) function $J = \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t))\ dt$ is minimized.

Optimization of a discrete set of decision variables is frequently applied to control problems by discretizing continuous dynamics and optimizing over a set of the control variables $\mathcal{U}$ and the states $\mathcal{X}$. Common formulations include direct transcription, direct collocation, and direct shooting, each of which has different approaches to specifying the number of decision variables and how the original dynamics are enforced. Within the controls community, uncertainty is often addressed through guaranteeing that a given controller will succeed despite bounded uncertainty (e.g. H^∞), or developing certain results in stochastic control (e.g. stochastic LQR) that are provably optimal in the absence of constraints. [4]. For this particular class of systems with quadratic cost on state error and input, linear dynamics, and Gaussian white zero-mean process noise, optimal control gains, $\mathbf{K}$ for a desired operating point can be found by solving the Ricatti equation, provided in Section II. However, this solution provides no control over the probabilistic nature of the cost function–it is only optimal in the expected value sense.

However, applying the results of robust optimization (RO) directly to controls problems is a tool that is still relatively new to the community [1]. An RO approach has benefits including explicit incorporation of constraints, and guarantees on cost assuming an uncertainty set on probabilistic disturbances. To solve the problem of operating a satellite in an optimal way given uncertain dynamics, an approach must be taken to account for the uncertainty or to provide certain guarantees about performance given this uncertainty. RO is a promising way to formulate operating in an optimal way with uncertain dynamics [3] [2], while allowing for control over "probabilistic protection" of the cost function. As an additional benefit, these approaches can incorporate state and input constraints, unlike LQR solutions.

II. PROBLEM STATEMENT

A system with state $\mathbf{x} \in \mathbb{R}^n$ and input $\mathbf{u} \in \mathbb{R}^m$ can be modeled as uncertain by incorporating an uncertainty vector, $\mathbf{w} \in \mathbb{R}^w$, into its dynamics.

A quadratic cost function is considered for the system, shown below:

$$J = \int_0^t \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u} \tag{1}$$

This cost function applies a penalty for not being at the desired state and for applying control input. For the purposes of optimization, this equation will be discretized as a summation:

$$J = \sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i + \mathbf{u}_{i-1}^T \mathbf{R}_i \mathbf{u}_{i-1} \tag{2}$$

The state dynamics are a function of the current state and the control input:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{y}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{w} \tag{3}$$

The dynamics may also be discretized:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + f(\mathbf{x}_n, \mathbf{u}_n)dt \tag{4}$$

The satellite attitude dynamics used in this project are 3 degrees of freedom (DoF) since they are linear, a simplification from 6 DoF nonlinearity. We seek a feedback law the minimizes Equation 2, but that allows for control over the degree of conservativeness due to the uncertain parameter. The purpose of this project is to prove feasibility in implementing this type of control in a system such as SPHERES. Once feasibility is shown in the simplified state, further work can be pursued to expand to more representative system dynamics.

III. PROPOSED APPROACH

A. Robust Counterpart Formulation-SDP

The formulation here follows Bertsimas et al. The ultimate purpose of implementing a controller is to minimize the cost function.

$$\begin{aligned} J(\mathbf{x}_0, \mathbf{u}, \mathbf{w}) = &\sum_{k=1}^{N} (\mathbf{x}_k^T \mathbf{Q}_k \mathbf{x}_k + 2\mathbf{q}_k^T \mathbf{x}_k) + \\ &\sum_{k=0}^{N-1} (\mathbf{u}_k^T \mathbf{R}_k \mathbf{u}_k + 2\mathbf{r}_k^T \mathbf{u}_k) \end{aligned} \tag{5}$$

This can also be written as follows:

$$\begin{aligned} J(\mathbf{x}_0, \mathbf{u}, \mathbf{w}) = 2\mathbf{a}^T \mathbf{x}_0 \mathbf{A} \mathbf{x}_0 + 2\mathbf{b}^T \mathbf{u} + \mathbf{u}^T \mathbf{B} \mathbf{u} \\ + 2\mathbf{c}^T \mathbf{w} + \mathbf{w}^T \mathbf{C} \mathbf{w} + 2\mathbf{u}^T \mathbf{D} \mathbf{w} \end{aligned} \tag{6}$$

The robust counterpart will be formed for this receding horizon control (RHC) problem [1]. To minimize the cost function in equation 6, it is sufficient to optimize over all $\mathbf{y}$ in the following cost-to-go function.

$$\tilde{J}(\mathbf{x}_0, \mathbf{y}, \mathbf{w}) = \mathbf{y}^T \mathbf{y} + 2\mathbf{h}^T \mathbf{w} + 2\mathbf{y}^T \mathbf{F} \mathbf{w} + \mathbf{w}^T \mathbf{C} \mathbf{w} \tag{7}$$

In this formulation the below substitutions are made for $\mathbf{h}$ and $\mathbf{F}$.

$$\mathbf{h} = \mathbf{c} - \mathbf{D}^T \mathbf{B}^{-1} \mathbf{b} \tag{8}$$

$$\mathbf{F} = \mathbf{B}^{-1/2} \mathbf{D} \tag{9}$$

An uncertainty set is given by:

$$\mathcal{W}_\gamma = \{w : ||w||_2 \leq \gamma\} \tag{10}$$

The cost can now be minimized in its robust formulation using the SDP found below.

$$\begin{aligned} &\min z \\ &\text{subject to} \\ &\begin{bmatrix} \mathbf{I} & \mathbf{y} & \mathbf{F} \\ \mathbf{y}^T & z - \gamma^2 \lambda & -\mathbf{h}^T \\ \mathbf{F}^T & -\mathbf{h} \lambda \mathbf{I} - \mathbf{C} + \mathbf{F}^T \mathbf{F} & \end{bmatrix} \succeq 0 \\ &\lambda \geq 0 \end{aligned} \tag{11}$$

The decision variables in this formulation are $\mathbf{y}$, z, and λ, with $\mathbf{y}$ being the primary decision variable of interest. When the decision variable of $\mathbf{y}$ is output, it is used in the transformation below to find the optimal control inputs.

$$\mathbf{u} = \mathbf{B}^{-1/2} \mathbf{y} - \mathbf{B}^{-1} \mathbf{b} \tag{12}$$

This formulation is computationally intractable to solve online for a long time horizon. Notably, it has some additional advantages other than robustness of performance, namely, that it can handle state and input constraints. If an unconstrained solution is suitable, it is even possible to find a closed-loop control law.

B. Robust Counterpart Formulation-SOCP

Since the SDP solution is generally unable to handle long time horizons (and its closed-loop form requires dropping constraints), it is desirable to find an alternative, fast solution capable of handling these constraints. The robust counterpart will be formed for this receding horizon control law problem [1]. To minimize the cost function in Equation 5, it is sufficient to optimize over all $\mathbf{y}$ as in Equation 7.

In Equation 7, y is a newly introduced decision variable vector, replacing the original statement of the cost-to-go. The uncertainty set is captured in:

$$W_\Omega = \left\{ \sum_{j=1}^{Nn_w} u_j e^i \mid \parallel u \parallel_2 \leq \Omega \right\} \tag{13}$$

Following Bertsimas et. al, the SOCP formulation below generates a feasible solution to the problem.

$$\begin{aligned} &\min z \\ &s.t. \\ &\|\mathbf{y}\|_2^2 \leq z - \Omega \tilde{y} - \Omega^2 \|\mathbf{C}\|_2^2 \\ &|2(\mathbf{h} + \mathbf{F}^T \mathbf{y})^T \mathbf{e}^j| \leq t_j \\ &j = 1, ..., N \cdot n_w, \|\mathbf{t}\|_2 \leq \tilde{y} \end{aligned} \tag{14}$$

There is a probabilistic performance guarantee (i.e. a guarantee on the cost-to-go) associated with the solution represented in the equation below. Note that this is for *open-loop* performance–guarantees on receding horizon use are challenging.

$$P(J(\mathbf{x}_0, \mathbf{y}*_{SOCP}, \mathbf{w}) > z*_{SOCP}) \leq \sqrt{\frac{e}{2}} \Omega exp(\frac{-\Omega^2}{4}) \tag{15}$$

The essential tradeoff being made in this approach by specifiyng Ω is protection of the cost function versus expected-value optimality.

C. Linear Quadratic Regulator (LQR) Comparison

1) The Linear Quadratic Regulator: The traditional approach to the problem proposed is to implement a stochastic linear quadratic regulator (LQR). The optimal input for a system using LQR control is a gain matrix K produced by solving the Ricatti equation:

$$\begin{aligned} &\mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B} \mathbf{P} + \mathbf{Q} = -\dot{\mathbf{P}} \\ &\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P} \\ &\mathbf{u}_i = -\mathbf{K} \mathbf{x}_i \end{aligned} \tag{16}$$

2) LQR Zero-Noise Baseline: The LQR solution (i.e. Ricatti solution) is used as the baseline for comparison in this project. There are advantages of using an LQR approach, including low computational burden on the system since optimal gains can be computed offline. Further, the ease of implementation and wide use of LQR makes it a good approach for comparison with the robust approach, which should mirror LQR performance as γ goes to zero.

Below shows a 14 second collage of a simulation using LQR under ideal, no-noise conditions. As expected, the system moves directly toward the goal (the origin), balancing fuel usage with state error. This is the optimal unconstrained policy.

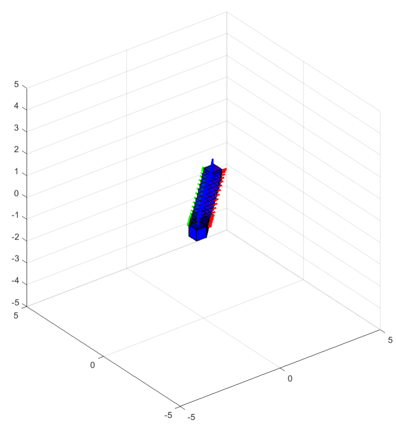

Fig. 1. Ideal motion of satellite using LQR.

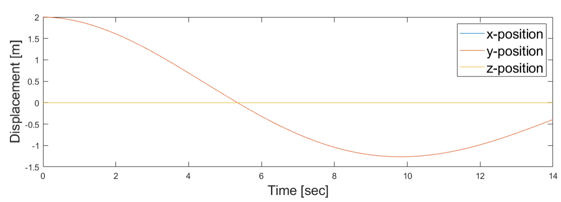

Fig. 2. Position error of satellite using LQR.

IV. Results

The SDP and SOCP approaches were implemented using the MOSEK Fusion MATLAB API, and results compared against baseline LQR for various levels of noise and γ. Our results serve as a verification of [1] for this system, demonstrating the process on a higher-dimensional state space ($n = 6$) with implications for a real satellite system.

A. LQR Baseline

LQR quickly becomes unable to converge toward the origin. Figure 8 LQR in response to a **w**=0.1 over 100 seconds. Overall the LQR did not handle noise well. The controller overshot its target, even after 140 s. Below is the first 14 seconds for lower **w**, which will be compared against the robust controller.

The system shows no indication of convergence. Implementing the robust control adds assurance of system stability.

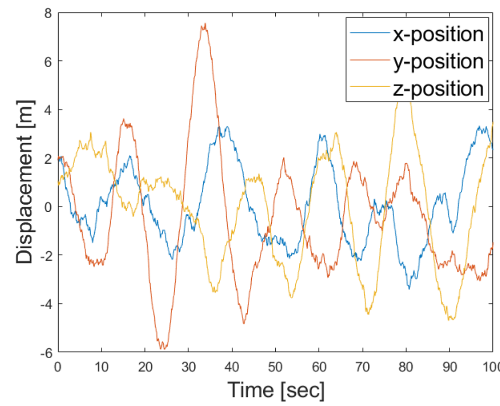

Fig. 3. LQR in response to **w**=0.1 noise.

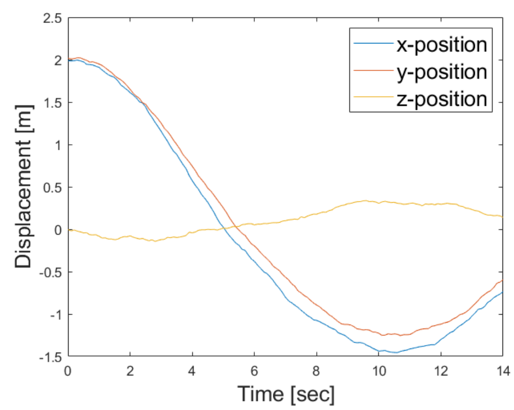

Fig. 4. Position error of satellite using LQR.

Becoming more robust to disturbances, by using a high γ to bound the uncertainty set, leads to higher total cost in the solution, but also a higher guarantee of stability in the presence of noise. It allows for a higher level of decision-making over the way the cost function probability distribution should be treated, rather than LQR's optimal expected-value formulation.

B. Robust Control Using SDP Formulation

The robust optimization problem formulated in Part A of Section III represents the optimal solution to this problem guaranteeing cost function performance over worst-case noise scenarios. The growing complexity of the problem drove the use of $N = 5$ step sizes into the future for a reasonable solution time (a few hundred milliseconds per solution). The solution of the SDP is a vector of future inputs, only the first m of which are applied to the system at a particular timestep (i.e. the receding horizon solution).

Figure 6 shows that as the noise increases the overall cost of the optimal solution increases, as expected. Adding noise to the problem greatly affects the stability of the system. Figure 7 shows the costs in a closed-loop solution, similar results arise, with the total cost increasing when γ increases. Overall, performance is slightly improved since reoptimization can occur based on the current state information.

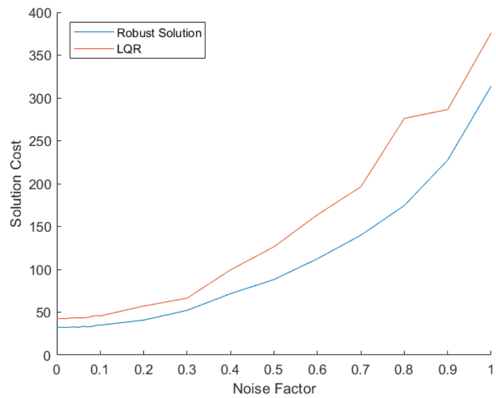

Fig. 5. Cost comparison of [sdp or socp?] robust formulation and LQR.

γ	J (**w** = 0.001)	J (**w** =0.01)	J (**w** = 0.1)
0	3.3209	3.3397	4.6047
0.001	3.3125	3.335	4.3842
0.01	3.325	3.3276	4.6623
0.1	3.3948	3.4186	4.5602
1.0	5.9402	5.9511	6.9493
LQR	4.0585	4.0706	5.999

Fig. 6. The results for the open-loop SDP formulation for varying levels of w and γ, compared against the average LQR baseline.

γ	J (**w** = 0.001)	J (**w** =0.01)	J (**w** = 0.1)
0.001	3.3261	3.2586	4.0336
0.01	3.3263	3.3373	4.9688
0.1	3.3817	3.4195	5.5989
1.0	3.9098	3.9493	9.8329
10	3.9124	4.1376	9.0717
LQR	4.0585	4.0706	5.999

Fig. 7. The results for the closed-loop SDP formulation for varying levels of w and γ, compared against the average LQR baseline.

C. Robust Control Using SOCP Formulation

The SOCP formulation can be solved closed-loop much faster than the SDP approach. Even though it is an approximation, the total cost was lower than open-loop SDP.

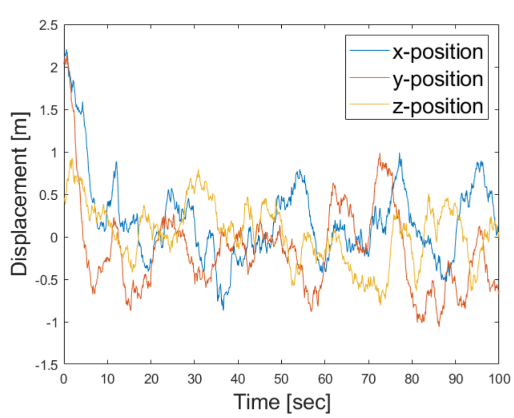

Fig. 8. SOCP system performance over $100s$.

The above figure shows the SOCP approach run over 100 seconds with a noise factor of **w**=0.1. Although the position does not settle at 0 at the end of this time frame, it does settle in to a range of +/- 0.5m, which is a better result that the LQR controller that has a range of closer to +/- 2 m after 100 seconds. Below shows a comparison of the cost of using the SOCP method when different magnitudes of noise and solution protection are introduced.

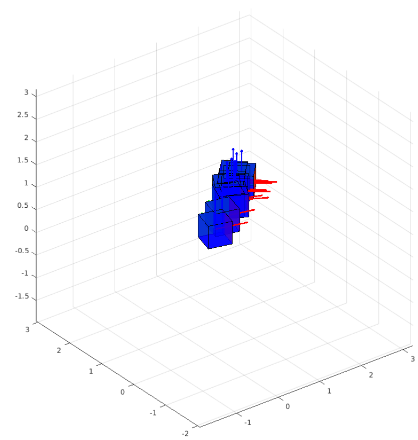

Fig. 9. A sample collage of a run of the SOCP approach, moving toward the origin in the presence of uncertain dynamics.

γ	J (**w** = 0.001)	J (**w** =0.01)	J (**w** = 0.1)
0.001	3.3281	3.3619	6.0897
0.01	3.3290	3.3597	6.1661
0.1	3.3540	3.3448	6.6999
1.0	3.7308	3.9813	8.8506
10	4.9254	4.9813	11.5006
LQR	4.0585	4.0706	5.999

Fig. 10. The results for the closed-loop SOCP formulation for varying levels of w and γ, compared against the average LQR baseline.

In general the cost of the SOCP solution was less than LQR in low noise, low γ scenarios. This cost becomes higher than LQR as noise gets higher, again due to the stability requirements being enforced by the SOCP solution. The table below shows the percentage difference in cost between the closed-loop SDP solution and the SOCP solutions.

γ	J (**w** = 0.001)	J (**w** =0.01)	J (**w** = 0.1)
0.001	0.06%	3.17%	50.97%
0.01	0.08%	0.67%	24.10%
0.1	0.82%	2.18%	19.66%
1.0	4.58%	0.81%	9.99%
10	25.89%	20.39%	26.77%

Fig. 11. Percent cost difference of SOCP relative to SDP.

The MATLAB code for this implementation can be found at: https://bit.ly/2WQAhwg.

V. CONCLUSION

A. Summary

This report details the modelling of a satellite with linear uncertain dynamics, moving to a desired state, x_g with a trajectory optimal with respect to a cost function J. After discretizing the problem, three solution approaches were considered. The robust counterpart was applied with the relevant constraints, following Bertsimas, and the system behavior was evaluated for various levels of uncertainty. Two robust approaches were considered: a slower SDP formulation, and a faster SOCP formulation. These robust approaches were compared relative to each other and against LQR in terms of cost. This project aims to demonstrate the applicability of robust optimization to applied optimal control, in a useful application of practical significance for a real problem in controlling satellites.

It is clear that the onboard modelling of the problem outperformed the offline solution provided by LQR regardless of the noise it was optimized for and the noise it received. Stability relative to LQR was improved. Matching Bertsimas, larger uncertainty sets decreased system performance relative to LQR, particularly for larger γ. The benefit, however, comes in terms of stability relative to the goal point. As expected, the SDP approach, while slower, leads to better cost performance than the SOCP.

A side benefit of on-board planning using the robust receding horizon approaches is the ease of which one can implement additional constraints into the problem. For example, obstacles can be put into constraints on the state and thruster limitations can be added as constraints on the inputs. This is impossible using LQR.

B. Future Work and Lessons Learned

This work has demonstrated application of robust SDP and SOCP approaches for the stochastic LQC problem on a $n = 6$ dimensional model, for potential use on a real system. The results reflect those seen in prior theoretical work, and could be promising for future robust control applications on satellite systems with a variety of uncertainties, including thruster degradation. Application on actual hardware is a logical next step. Finally, application of new techniques in adaptive robust optimization could be applied, as the time-horizon nature of this problem seems to lend itself to adaptive decision rules.

A takeaway from implementing the approach is the importance of being familiar with posing problems in an optimization-friendly way. There are a variety of tricks required when working with different optimization APIs to formulate a straightforward mathematically-stated problem into one that is compatible with the optimization backend. For example, writing an L2-norm requires using quadratic cone constraints and prepending the equation's right-hand side to the vector. Becoming familiar with conventions like this will help the reader in writing their own implementations.

REFERENCES

[1] Dimitris Bertsimas and David B. Brown. Constrained stochastic LQC: A tractable approach. *IEEE Transactions on Automatic Control*, 52(10):1826–1841, 2007.

[2] David B Brown and Constantine Caramanis. Theory and Applications of Robust Optimization Robust Optimization. 2012.

[3] Paul J. Goulart, Eric C. Kerrigan, and Daniel Ralph. Efficient robust optimization for robust control with constraints. *Mathematical Programming*, 114(1):115–147, 2008.

[4] Eric C. Kerrigan and Jan M. Maciejowski. On robust optimization and the optimal control of constrained linear systems with bounded state disturbances. *2003 European Control Conference (ECC)*, pages 1453–1458, 2018.