AAS 20-755

# MODEL-PREDICTIVE ATTITUDE CONTROL FOR FLEXIBLE SPACECRAFT DURING THRUSTER FIRINGS

**Kevin Tracy*, Zachary Manchester†**

We present a model-predictive control technique for attitude control of flexible spacecraft during thruster firings. Due to the flexibility of the spacecraft, control laws designed under the rigid-body assumption can perform poorly during thruster-induced deflections. A model-predictive controller, in contrast, can leverage a greater understanding of the flexible-body dynamics and actuator constraints, and can achieve significantly better pointing performance than traditional control laws. To demonstrate the effectiveness of this control strategy, we compare the MPC controller with an LQR feedback controller during a thruster firing in the presence of noise and model uncertainty. The MPC controller performs significantly better, enabling lighter and more flexible spacecraft designs.

## INTRODUCTION

Spacecraft engineers must often trade between mass and structural stiffness, with the stiffness of some appendages, like antennas, being critical to mission success. For appendages like probes and solar arrays, moderate flexibility can be afforded because payload performance isn't directly affected. In this case, the need for structural stiffness is driven by attitude control requirements. Control of rigid spacecraft is well-understood, and has been thoroughly studied over many decades.[1,2] The goal of this paper is to describe a method where flexible spacecraft can be controlled with similar attitude control performance to that of a rigid body, enabling lighter and less expensive spacecraft designs.

During nominal pointing operations, a spacecraft with flexible appendages can operate without much concern for the flexible modes. In the presence of a thruster firing, however, the flexible behavior is excited by both translational acceleration and torque resulting from thruster misalignments, plume impingements, and uncertainty about the center of mass. In some cases, the flexible dynamics of the spacecraft appendages can be destabilizing if the attitude control system is not designed to properly account for it. But, more often, it simply results in large pointing errors.

The approaches that have been explored to deal with the control of flexible spacecraft fall into three different categories: The first approach is to perform input shaping on the thruster response, avoiding excitation of certain modes in the structure. Banerjee[3] explored this approach as it related to actively damping out the flexible vibrations after an orbit adjustment. A second approach is to filter the sensor measurements with prior knowledge of the flexible behavior. Wie[4] employed this approach for the Intelsat V series of geostationary spacecraft during station-keeping maneuvers. Finally, the third approach is to design feedback controllers that exploit some understanding of the

*Department of Mechanical Engineering, Stanford University, 496 Lomita Mall, Stanford, CA 94305.
†Department of Aeronautics and Astronautics Engineering, Stanford University, 496 Lomita Mall, Stanford, CA 94305.

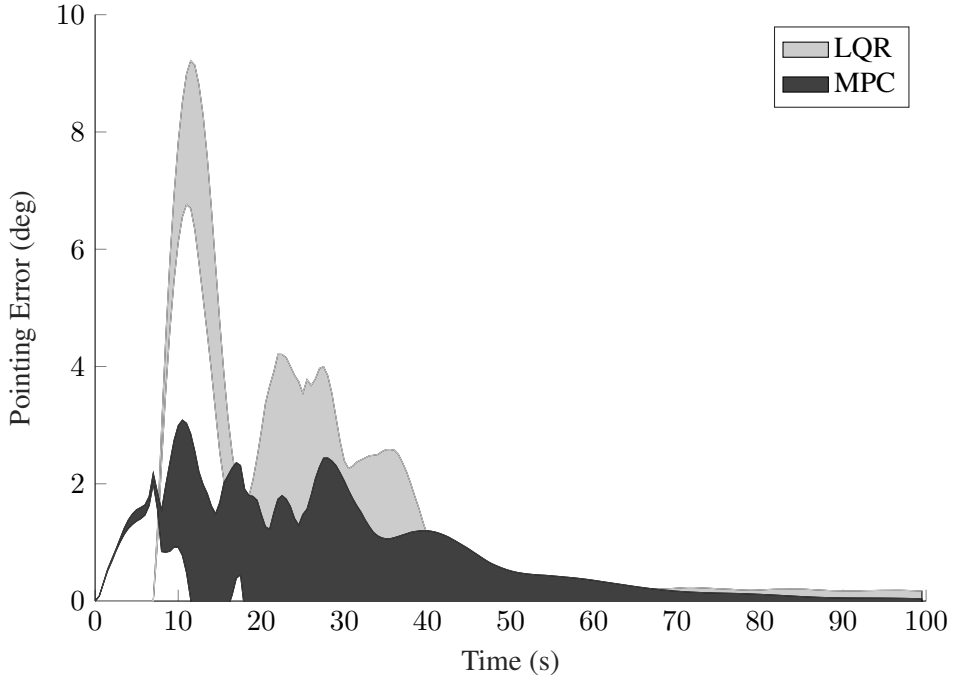


**Figure 1. Attitude pointing error distributions are shown for LQR and MPC flexible body spacecraft attitude control for 1000 Monte-Carlo trials. The ranges shown for each control strategy encompass 3 standard deviations above and below the respective mean performances. Perturbations in pointing are caused by a thruster firing at 7.5 seconds.**

flexible dynamics to decrease the possibility of destabilizing the spacecraft.[5] All three of these approaches have been employed with success, but can only handle limited flexibility, and involve highly non-trivial design and tuning for each spacecraft.

Model-Predictive Control (MPC) is a strong candidate for flexible body spacecraft control because it can reason about the flexible dynamics of the spacecraft, respect actuator constraints, and plan ahead for thruster firings. The cost of MPC comes from a computational perspective, where traditionally it was difficult to run MPC onboard a spacecraft with modest computational resources. Convex optimization solvers, specifically Quadratic Program (QP) solvers, have now progressed to the point where MPC problems can be solved in less than a millisecond on modest processors,

therefore placing a negligible burden on flight software load.

This paper presents a fast and efficient MPC formulation of the flexible-spacecraft attitude control problem that offers a significant improvement in both performance and robustness over traditional feedback controllers. The MPC framework that is described is generic to any flexible spacecraft, robust to model uncertainty, and is solved extremely quickly by a QP solver. The paper proceeds with sections on the full nonlinear dynamics of a flexible spacecraft, a simplified linearized dynamics model, an MPC formulation for the linearized system, and, finally, simulation results.

## DYNAMICS

Many different dynamics models have been employed to simulate flexible-body behavior of spacecraft. Many of them employ Kane's method,[6–8] as well as more recent advancements from Banerjee.[9] For this paper, a test spacecraft based on the JPL Thermoelectric Outer Planet Spacecraft (TOPS) concept is used as the flexible spacecraft.[10] The flexible dynamics are described using a truncated modal coordinate method.[11] This model leverages the mode shapes and frequencies from a finite-element analysis to describe momentum coupling matrices that relate torques and forces to modal excitation. The model is truncated to some order $j$, such that only the first $j$ modes are incorporated in the dynamics. The state for the dynamics simulation is as follows,

$$x = [p^T, \omega^T, \eta^T, \dot{\eta}^T, r^T]^T, \tag{1}$$

where $p$ is the 3-dimensional vector of modified Rodrigues parameters (MRP)[1,12] that represents the spacecraft's attitude, $\omega$ is the angular velocity of the central body expressed in the body frame, $\eta \in \mathbb{R}^j$ is a vector of modal-coordinate displacements, $\dot{\eta} \in \mathbb{R}^j$ is a vector of modal-coordinate velocities, and $r \in \mathbb{R}^m$ is the reaction wheel rotor speeds. The control input $u \in \mathbb{R}^m$, is the angular acceleration applied to the reaction wheels. The nonlinear dynamics expressed in the body-fixed frame are,

$$J\dot{\omega} + \omega \times [J\omega + B_{sc}r + G^T\dot{\eta}] + G^T\ddot{\eta} = \tau_d - B_{sc}u, \tag{2}$$

$$\ddot{\eta} + C\dot{\eta} + K\eta + \Phi f = -G\dot{\omega}, \tag{3}$$

$$\dot{r} = u, \tag{4}$$

where $J \in \mathbb{R}^{3\times3}$ is the spacecraft central body inertia matrix, $B_{sc} \in \mathbb{R}^{3\times m}$ is the reaction wheel Jacobian, $G \in \mathbb{R}^{j\times3}$ is the flexible angular momentum coupling matrix, $\Phi \in \mathbb{R}^{j\times3}$ is the flexible linear momentum coupling matrix, $\tau_d$ are the external disturbance torques, $C \in \mathbb{R}^{j\times j}$ is the modal damping matrix, $K \in \mathbb{R}^{j\times j}$ is the modal stiffness matrix, and $f$ is the translational acceleration. The kinematics of the MRP are as follows:[1]

$$\dot{p} = \frac{1 + p^T p}{4}\left(I_3 + 2\frac{[p\times]^2 + [p\times]}{(1 + p^T p)}\right)\omega. \tag{5}$$

An MRP has been chosen to parameterize the attitude instead of a quaternion because it will linearize well in the absence of the unit norm constraint, and the singularity present in the MRP will not be reached in stationary attitude control scenarios.[1] Together, these represent a system of $6+m+2j$ first-order ODE's, where j is the number of flexible modes expressed in the dynamics. These equations must be rearranged in order to represent the state derivatives explicitly. Equation (3) can be substituted into equation (2) resulting in the following expression for angular acceleration:

$$\dot{\omega} = [J - G^T G]^{-1}[\tau_d - B_{sc}u - \omega \times h + G^T(C\dot{\eta} + K\eta + \Phi f) + GT(\tau_d + G^T\Phi f) + \Phi f], \tag{6}$$

where $h = [J\omega + B_{sc}r + G^T\dot{\eta}]$ is the total angular momentum of the spacecraft. Equation (6) can now be plugged back into equation (3) giving the expression for the modal coordinate acceleration vector,

$$\ddot{\eta} = -\big[(C + GTG^TC)\dot{\eta} + (K + GTG^TK)\eta + GT\tau_d - GTB_{sc}u + GTG^T\Phi f + \Phi f\big], \quad (7)$$

where $T = [J - G^TG]^{-1}$. Equations (4) - (7) will be used to simulate the true nonlinear dynamics of the flexible spacecraft.

**Linearized Dynamics**

The dynamics can be linearized about the desired spacecraft pointing state with zero angular velocity and no modal displacement. The spacecraft reaction wheels will no longer be an active state in the dynamics, but the nominal rotor speed, $r_0$, at time of linearization, will be. The resulting state vector is,

$$x = \begin{bmatrix} p^T, & \omega^T, & \eta^T, & \dot{\eta}^T \end{bmatrix}^T, \quad (8)$$

which has dimension $6 + 2j$ where $j$ is the number of modes represented in the model. The system will now be linearized by taking a first order Taylor series about the steady state point:

$$x_{ss} = 0_{6+2j}, \quad (9)$$

$$u_{ss} = 0_3. \quad (10)$$

To perform the linearization, equations (4) - (7) are differentiated with respect to the state and control variables at the steady state values, resulting in the following Jacobian with respect to state,

$$A = \left.\frac{\partial \dot{x}}{\partial x}\right|_{x_{ss},u_{ss}} = \begin{bmatrix} 0_{3\times3} & \frac{1}{4}\mathrm{I}_3 & 0_{3\times j} & 0_{3\times j} \\ 0_{3\times3} & [(B_{sc}r_0)\times] & TG^TK & TG^TC \\ 0_{j\times3} & 0_{j\times3} & 0_{j\times j} & \mathrm{I}_j \\ 0_{j\times3} & 0_{j\times3} & [-K - GTG^TK] & [-C - GTG^TC]. \end{bmatrix}, \quad (11)$$

as well as the following input Jacobian:

$$B = \left.\frac{\partial \dot{x}}{\partial u}\right|_{x_{ss},u_{ss}} = \begin{bmatrix} 0_{3\times3} \\ -TB_{sc} \\ 0_{j\times3} \\ GTB_{sc} \end{bmatrix}. \quad (12)$$

The linearized dynamics also include the following affine term to account for thruster-induced forcing:

$$d(\tau, f) = \dot{x}_{ss}(\tau, f) = \begin{bmatrix} 0_{3\times1} \\ T\tau_d + TG^T\Phi f \\ 0_{j\times1} \\ -GT\tau_d - GTG^T\Phi f - \Phi f \end{bmatrix}. \quad (13)$$

The final linearized equations of motion can be written in state space form as follows:

$$\begin{bmatrix} \dot{p} \\ \dot{\omega} \\ \dot{\eta} \\ \ddot{\eta} \end{bmatrix} = A \begin{bmatrix} p \\ \omega \\ \eta \\ \dot{\eta} \end{bmatrix} + Bu + d(\tau, f). \quad (14)$$

Note that there are three important differences between the linearized model and the true nonlinear model: The first is the removal of the rotor velocities from the state, the second is the removal of the gyroscopic coupling term, and the third is the approximation of the attitude kinematics with $\dot{p} = (1/4)\,\omega$. All three of these approximations will remain valid while the spacecraft is close to the nominal attitude and the angular velocities remain small.

### Discretization of Linearized Dynamics

In order to discretize the continuous system (14), the matrix exponential will be used. For a generic homogeneous linear ODE of the form $\dot{x} = Ax$, the solution for $x$ after a time $\delta t$, can be expressed using the matrix exponential and the initial condition:[13,14]

$$\dot{x} = Ax, \tag{15}$$

$$x(t_0 + \delta t) = \exp(A \cdot \delta t)x(t_0). \tag{16}$$

For a forced affine ODE, where the control input and affine forcing term are assumed constant over a time step, the state can simply be augmented with these terms,

$$\begin{bmatrix} \dot{x} \\ \dot{u} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} Ax + Bu + d \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} A & B & I_n \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \\ d \end{bmatrix}, \tag{17}$$

and this system can be discretized with a sample time of $\delta t$ in the same way as (16)

$$\begin{bmatrix} x_{t+1} \\ u_{t+1} \\ d_{t+1} \end{bmatrix} = \exp\left( \begin{bmatrix} A & B & I_n \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \delta t \right) \begin{bmatrix} x_t \\ u_t \\ d_t \end{bmatrix}. \tag{18}$$

Finally, we obtain in the following difference equation,

$$x_{t+1} = A_d x_t + B_d u_t + D_d d_t, \tag{19}$$

where the transition matrices come from the matrix exponential,

$$\begin{bmatrix} A_d & B_d & D_d \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} = \exp\left( \begin{bmatrix} A & B & I_n \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \delta t \right). \tag{20}$$

## MODEL-PREDICTIVE CONTROL

A Model-Predictive Control (MPC) approach is ideal for spacecraft with increased flexibility because it can plan for thruster firings, account for constraints on the state or control variables, and reason about the flexible body dynamics. This is done by developing feed-forward control plans for a finite horizon, executing the first control in the series, and re-solving the problem with the new measured state as the initial condition. This attitude control problem is formulated as an optimization problem where a cost function is minimized subject to dynamics, actuator, and state, constraints. More specifically, by formulating this problem as a convex optimization problem, it can be solved with speed and reliability.

**Formulation as a Quadratic Program**

This MPC problem will be formulated as a convex QP, a specific type of convex optimization problem with many fast and robust solvers available. The convexity of the MPC problem guarantees that any locally optimal solution is globally optimal,[15] without any need for an initial guess. For larger problems and those with poor scaling, numerical issues can occur, but this specific problem will be posed such that these issues are avoided.

The first step to converting this problem to a QP is formulating quadratic stage and terminal cost functions,

$$\ell_k(x_k, u_k) = \frac{1}{2}(x_k - x_g)^T Q(x_k - x_g) + \frac{1}{2}u_k^T R u_k, \tag{21}$$

$$\ell_N(x_N) = \frac{1}{2}(x_N - x_g)^T Q_N(x_N - x_g), \tag{22}$$

where $x_g$ is the goal state. The system dynamics are captured in a set of linear equality constraints using equation (19). Lastly, there are box constraints on the control and state variables. Together, this can be formulated as a convex QP:[15]

$$\begin{aligned} \min_{x_{1:N},\, u_{0:N-1}} \quad & \frac{1}{2}x_N^T Q_N x_N - x_g^T Q_N x_N + \sum_{k=0}^{N}\left(\frac{1}{2}x_k^T Q x_k - x_g^T Q x_k + \frac{1}{2}u_k^T R u_k\right) \\ \text{s.t.} \quad & x_{k+1} = A_d x_k + B_d u_k + D_d d, \qquad k = 0, \ldots, N-1, \\ & u_{lower} \le u_k \le u_{upper}, \qquad k = 0, \ldots, N-1, \\ & x_{lower} \le x_k \le x_{upper}, \qquad k = 0, \ldots, N. \end{aligned} \tag{23}$$

Problem (23) can be reformulated as a standard-form QP for use with the Operator Splitting Quadratic Program (OSQP) solver.[16] OSQP is a fast, ADMM-based,[17] QP solver that has extensive warm-starting capabilities, making it particularly well suited for MPC. Direct transcription[18] will be used, with both the state and control inputs concatenated into a single vector of decision variables for an $N$-step horizon problem,

$$\mathbf{x} = \begin{bmatrix} x_0^T, & x_1^T, & x_2^T, & \ldots, & x_N{}^T, & u_0^T, & u_1^T, & u_2^T, & \ldots, & u_{N-1}^T \end{bmatrix}^T. \tag{24}$$

Problem (23) can now be cast as a QP in standard form,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2}\mathbf{x}^T P \mathbf{x} + q^T \mathbf{x} \\ \text{s.t.} \quad & F\mathbf{x} = w, \\ & lb \le \mathbf{x} \le ub, \end{aligned} \tag{25}$$

where the problem matrices $P$, $q$, $F$, $w$, $lb$, and $ub$ are defined below. First, the quadratic cost term P is constructed such that $Q$ applies to $x_{0:N-1}$, $Q_N$ applies to $x_N$, and $R$ applies to $u_{0:N-1}$,

$$P = \text{blkdiag}\big(I_N \otimes Q,\ Q_N,\ I_N \otimes R\big), \tag{26}$$

where $\otimes$ is the Kronecker product.[13] Similarly, $q$ can be written compactly as,

$$q = \begin{bmatrix} 1_N \otimes -Qx_g \\ -Q_N x_g \\ 0_{(N \cdot nu) \times 1} \end{bmatrix}, \tag{27}$$

with $1_N$ denoting a column vector of $N$ ones. Next the dynamics, state, and control constraints must be represented in (25). The dynamics constraints will be embedded in the equality constraint $F\mathbf{x} = w$, with the following for $F$:

$$F = \left[ \left( I_{N+1} \otimes -I_{nx} + \begin{bmatrix} 0_{1\times N} & 0_{1\times 1} \\ I_N & 0_{N\times 1} \end{bmatrix} \otimes A_d \right) \quad \left( \begin{bmatrix} 0_{1\times N} \\ I_N \end{bmatrix} \otimes B_d \right) \right]. \tag{28}$$

The corresponding vector, $w$, includes the initial condition of $x_0$, as well as the affine contribution to the dynamics at each time step:

$$w = \begin{bmatrix} -x_0 \\ -D_d d(\tau_0, f_0) \\ -D_d d(\tau_1, f_1) \\ \vdots \\ -D_d d(\tau_{N-1}, f_{N-1}) \end{bmatrix}, \tag{29}$$

Where the function $d(\tau, f)$ is expressed in equation (13), and the affine transition matrix $D_d$ is defined in equation (20). For the box constraints on $\mathbf{x}$, the lower and upper bounds will equal to the state and control constraints respectively:

$$lb = \begin{bmatrix} 1_{N+1} \otimes x_{lower} \\ 1_N \otimes u_{lower} \end{bmatrix}, \qquad ub = \begin{bmatrix} 1_{N+1} \otimes x_{upper} \\ 1_N \otimes u_{upper} \end{bmatrix}. \tag{30}$$

Lastly, it is important to note that the problem matrices $P$ and $F$ are both sparse, consisting mostly of zeros. By properly leveraging sparse matrix libraries, OSQP can exploit this sparsity, resulting in much faster solve times.

## EXAMPLES

The MPC approach was compared to an LQR feedback control law for a sample thruster firing. In all tests, the thruster was fired at the 7.5 second mark, exciting all of the modal coordinates through both translational acceleration and disturbance torques. The pointing error achieved by both controllers is plotted in Figure 2. The MPC solution was able to anticipate the thruster firing and converge on an optimal trajectory that minimized pointing error. The LQR controller, in contrast, was only able to react to the attitude errors after they accumulated, and had significant trouble maintaining pointing during this period.

### LQR Controller Cost Function

For the LQR feedback controller, the same linear dynamics model described in equation (19) was used. This discrete model was then used in conjunction with a quadratic cost function to determine the optimal feedback gain matrix. Since the goal state for this system is $x_g = 0_{(6+2j)\times 1}$, the cost function for this problem was the following:

$$\ell(x, u) = \sum_{0}^{\infty} (x_k^T Q x_k + u_k^T R u_k), \tag{31}$$

where Q and R are diagonal matrices that determine the weights on each state and control,

$$Q = \text{blkdiag}(100 \cdot I_3,\ 30 \cdot I_3,\ I_3,\ I_3), \qquad R = 5 \cdot I_3. \tag{32}$$

These gains were tuned to minimize the spacecraft pointing error, while also being careful to avoid actuator saturation. Clamping was used to ensure that the commanded reaction wheel acceleration did not exceed the torque limits of $\pm\, 0.01\, N \cdot m$.

**MPC Controller Cost Function**

Similar to the LQR controller, weights were tuned for the MPC cost function expressed in (23). Since the constraints on the reaction wheels are explicitly handled in the MPC framework, the cost weighting on control usage is significantly less important in ensuring the controller functions well. The cost function chosen for this example is described by the following $Q$ and $R$ weighting matrices:

$$Q = \text{blkdiag}(20 \cdot I_3,\ I_3,\ I_3,\ I_3), \qquad\qquad R = 10 \cdot I_3. \tag{33}$$

It is worth re-iterating that the differences in cost functions between the LQR and MPC controllers is a result of needing to tune (32) to avoid actuator limits, while these are explicitly handled by constraints in the MPC controller.

**Simulation**

A spacecraft with 3 flexible modes was used for this experiment. The flexible characteristics were chosen to best exemplify a rigid spacecraft bus with flexible appendages. For this example, a spacecraft was used with the following inertia dyadic[19]

$$J = \begin{bmatrix} 1.0200 & 0.0946 & 0.1381 \\ 0.0946 & 1.9979 & -0.0975 \\ 0.1381 & -0.0975 & 2.9821 \end{bmatrix} kg \cdot m^2, \tag{34}$$

as well as the following angular and linear momentum coupling matrices,

$$\Phi = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & .2 & -.8 \end{bmatrix} kg \cdot m/s^2, \qquad G = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -.7 & .1 & .1 \end{bmatrix} kg \cdot m/s^2. \tag{35}$$

Together, these describe the mode shapes, and their impact on the spacecraft dynamics. These modes oscillate with respect to the following damping and stiffness matrices,

$$C = \begin{bmatrix} .0006 & 0 & 0 \\ 0 & .0025 & 0 \\ 0 & 0 & .0016 \end{bmatrix} 1/s, \qquad K = \begin{bmatrix} .0987 & 0 & 0 \\ 0 & 1.5791 & 0 \\ 0 & 0 & 0.6169 \end{bmatrix} 1/s^2. \tag{36}$$

These matrices correspond to natural frequencies of $\sigma = .05$, .2 and .125 Hz, with damping ratios of $\zeta = .001$. This completes the description of the flexible appendages up to the 3rd mode. The last parameter for the full spacecraft description is the reaction wheel Jacobian, which for this example, was chosen to be the identity matrix:

$$B_{sc} = I_3, \tag{37}$$

The reaction wheel Jacobian, $B_{sc}$, describes how the reaction wheels contribute to the angular momentum of the spacecraft. The MPC problem was solved for a sample rate of 2 hz, and a horizon of 100 steps. The QP was solved at every time step, and only the first control input of the 100-step horizon is executed before re-solving. Using OSQP,[16] each QP took less than 1 ms to solve*. This is due to the general speed of OSQP, it's ability to exploit problem sparsity, as well as the ability to warm start the solver with the solution from the previous time step.

Controlling the flexible spacecraft with MPC during this period was compared against a generic Linear Quadratic Regulator (LQR) feedback control law that was a formulated from the same linear system used in the MPC approach. These two methods were used to close the loop on the true nonlinear dynamics, and the pointing error is plotted in figure 2.

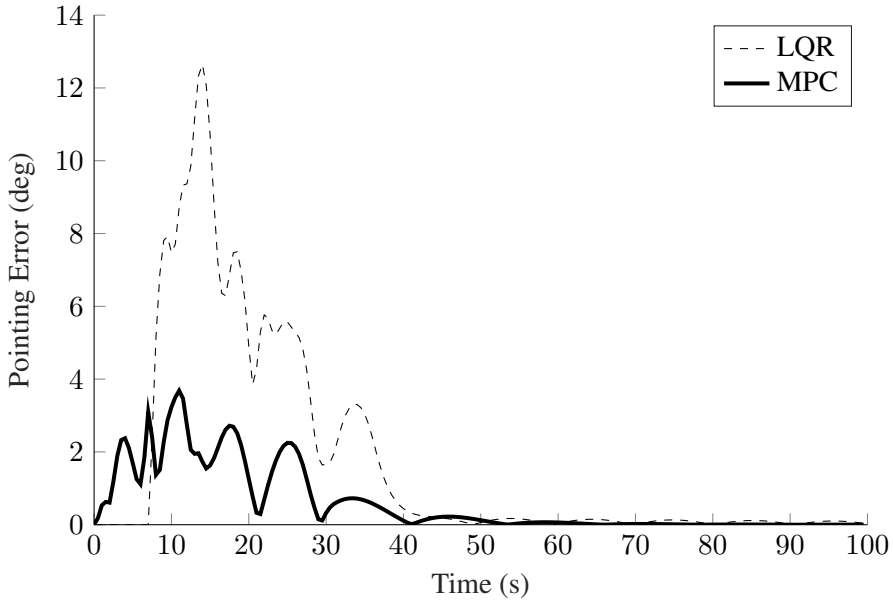


**Figure 2. Pointing errors for LQR and MPC attitude control stationkeeping methods for a flexible spacecraft. Perturbations are due to a chemical thruster firing at 7.5 seconds.**

*solve times reflect testing with a quad-core Intel Core i7-4870HQ @ 2.5 GHz

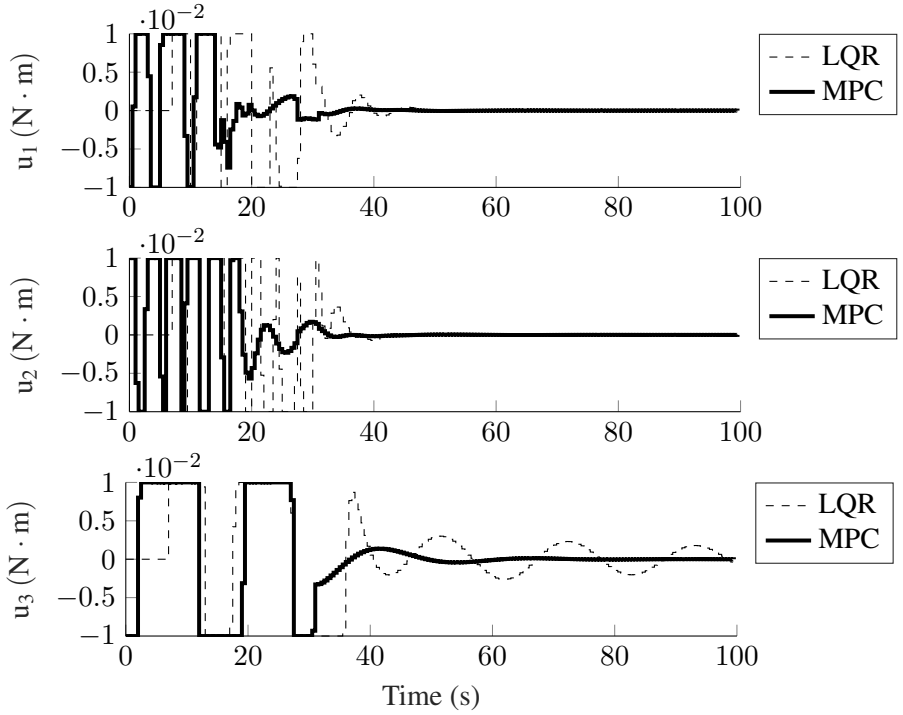


**Figure 3. Reaction wheel control torques for LQR and MPC for flexible attitude stationkeeping.**

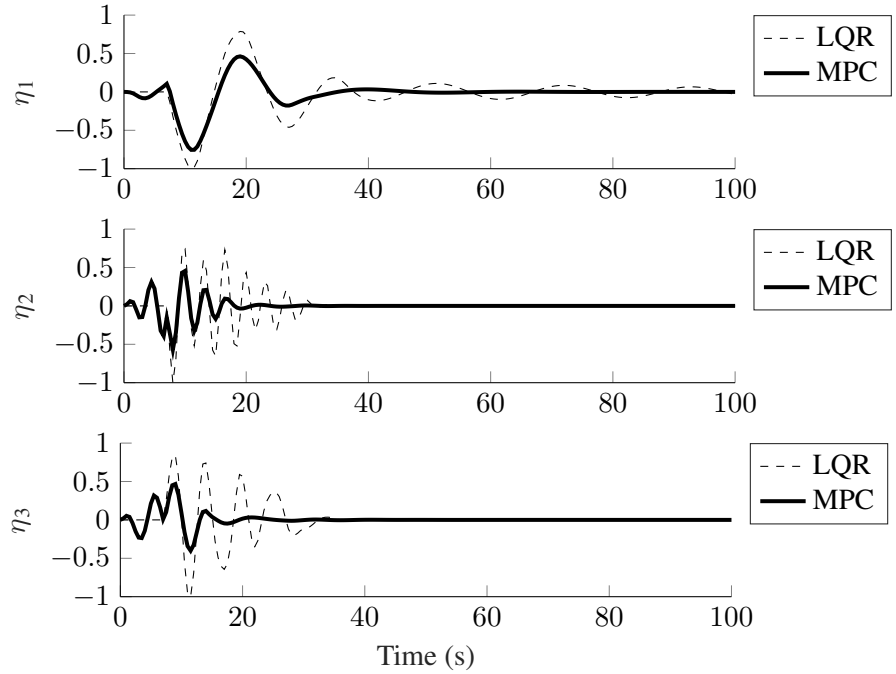


**Figure 4. Normalized modal coordinate displacements for the first three modes of the flexible spacecraft during LQR and MPC attitude stationkeeping.**

The MPC solution significantly outperforms LQR in pointing error, control usage, and modal coordinate excitement. This results in higher pointing accuracy and better payload performance, with the added benefit of less power draw from the reaction wheels. Another advantage of the MPC solution is that the controls respond to the thruster firing for 40 seconds, and after that point, the control period is effectively finished. In the LQR case, the feedback controller has trouble dealing with lowest mode, and had to perform control for the full 100 seconds. This is worse from a power consumption perspective, as well as potentially introducing more jitter into the spacecraft payload.

**Robustness Analysis**

The previous section demonstrated the overwhelming advantage that MPC has when compared to LQR for a stationkeeping control policy in the presence of a thruster firing. In that example, the system dynamics were known with absolute certainty. In this section, a series of Monte-Carlo simulations are run with uncertainty applied to various aspects of the dynamics model. This tests the robustness that MPC and LQR have to model parameters, more specifically, robustness to poor characterization of the flexible behavior. This is especially relevant for flexible spacecraft because representative flexible testing on earth is challenging. One thousand trials were run, with normally distributed multiplicative errors applied to model parameters at the start of each trial. The standard deviations for each of these errors are captured in table 1. The 3-$\sigma$ bounds on the pointing error

| **Parameter** | **Standard Deviation** |
|---|---|
| Damping Ratio | 10% |
| Natural Frequencies | 10% |
| Angular Coupling Matrix Rotation | 5 deg |
| Angular Coupling Matrix Scaling | 5% |
| Linear Coupling Matrix Rotation | 5 deg |
| Linear Coupling Matrix Scaling | 5% |

**Table 1. Multiplicative Gaussian white noise applied to various parameters during Monte-Carlo runs.**

performance for these trials are captured in figure 1. This plot shows variation in both the LQR and MPC performances based on the variations applied to the model parameters. The LQR pointing errors reach a higher maximum pointing error, as well as accrue a larger average pointing error than MPC. To look at the differences in performance, the root-mean-square error was calculated for each Monte Carlo run and plotted for both the LQR and MPC cases in figure 5.

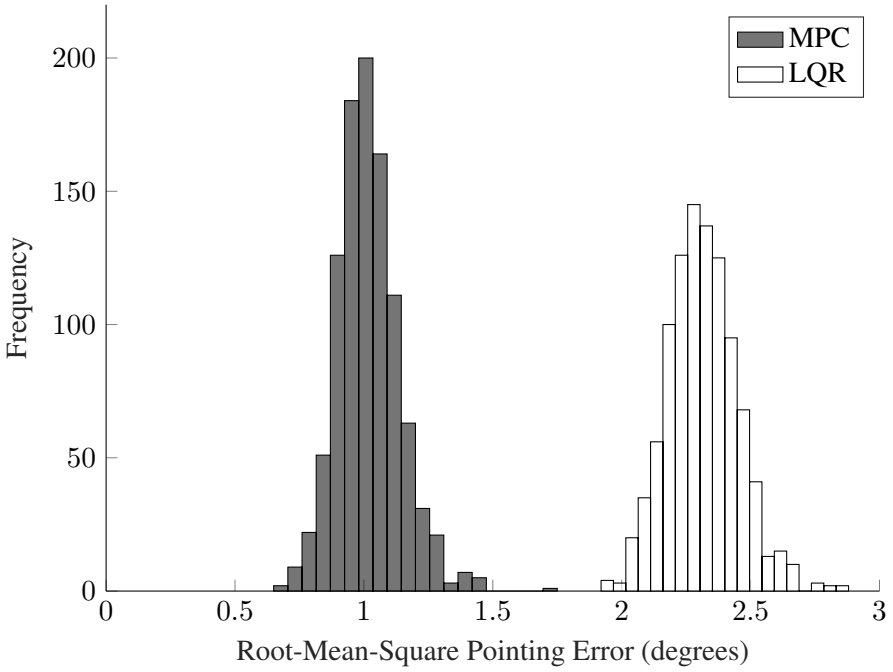


**Figure 5. Root-mean-square attitude pointing error for 1000 Monte-Carlo runs of MPC and LQR for flexible spacecraft control.**

The root-mean-square pointing error is an important metric for determining control performance, but it does not tell the full story. Many attitude control requirements specify a maximum allowable pointing error, where the pointing accuracy of the spacecraft must be constrained to be within a certain threshold. Because of this, the maximum experienced pointing error for the spacecraft was recorded for each run, and plotted in figure 6. In both figures 5 and 6, it is clear that the attitude pointing performance of MPC is significantly better than that of LQR. In 1000 trials, there was not a single example where MPC was outperformed by LQR in either of these two categories. This is a testament to the effectiveness of MPC, even when the flexible behavior of the spacecraft is poorly characterized.

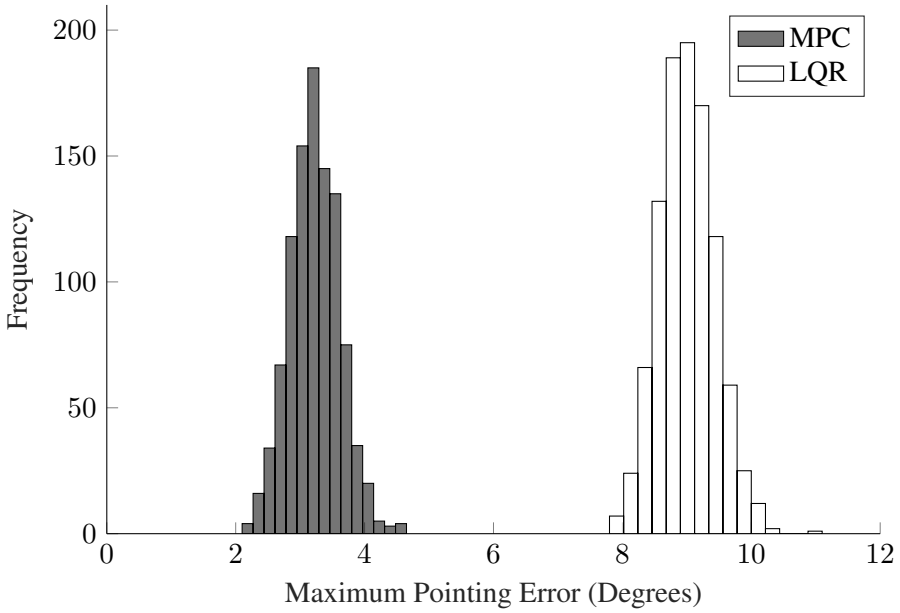


**Figure 6. Maximum attitude pointing error for 1000 Monte-Carlo runs of MPC and LQR for flexible spacecraft control.**

## CONCLUSIONS

A method for controlling flexible spacecraft during thruster firings using MPC was detailed and compared to traditional feedback control methods. MPC was shown to significantly outperform LQR feedback control in both pointing accuracy and robustness. The MPC problem was formulated in such a way that it can be readily solved by a standard QP solver with sub-millisecond solve times by taking advantage of problem structure and warm starting. The QP solver chosen for this experiment, OSQP,[16] is open-source and readily available for embedded systems, making it a strong candidate for onboard use in flight software. The ability to robustly control highly flexible structures opens up possibilities for less rigid spacecraft designs that are lighter and cheaper to produce, with only a small penalty for increased computational load. All code for the simulations presented in this paper is available online at `github.com/RoboticExplorationLab/FlexibleSpacecraftMPC`.

## REFERENCES

[1] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer New York, 2014, 10.1007/978-1-4939-0802-8.

[2] J. R. Wertz, ed., *Spacecraft Attitude Determination and Control*, Vol. 73 of *Astrophysics and Space Science Library*. Dordrecht: Springer Netherlands, 1978, 10.1007/978-94-009-9907-7.

[3] A. Banerjee and B. Diedrich, "Spacecraft Vibration Reduction Following Thruster Firing for Orbit Adjustment," *Journal of Guidance, Control, and Dynamics*, Vol. 26, July 2003, pp. 659–662, 10.2514/2.5097.

[4] B. Wie and C. T. Plescia, "Attitude Stabilization of Flexible Spacecraft during Stationkeeping Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 7, July 1984, pp. 430–436, 10.2514/3.19874.

[5] C. Pukdeboon and A. Jitpattanakul, "Anti-Unwinding Attitude Control with Fixed-Time Convergence for a Flexible Spacecraft," *International Journal of Aerospace Engineering*, Vol. 2017, 2017, pp. 1–13, 10.1155/2017/5018323.

[6] T. R. Kane, P. W. Likins, and D. A. Levinson, *Spacecraft dynamics*. McGraw-Hill, Inc., 1983.

[7] T. R. Kane and D. A. Levinson, "Formulation of Equations of Motion for Complex Spacecraft," *Journal of Guidance and Control*, Vol. 3, Mar. 1980, pp. 99–112, 10.2514/3.55956.

[8] E. Stoneking, "Implementation of Kane's Method for a Spacecraft Composed of Multiple Rigid Bodies," *AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, MA, American Institute of Aeronautics and Astronautics, Aug. 2013, 10.2514/6.2013-4649.

[9] A. K. Banerjee, ed., *Flexible Multibody Dynamics: Efficient Formulations and Applications*. Oxford, UK: John Wiley & Sons, Ltd, Apr. 2016, 10.1002/9781119015635.

[10] P. W. Likins and G. E. Fleischer, "Results of Flexible Spacecraft Attitude Control Studies Utilizing Hybrid Coordinates," *Journal of Spacecraft and Rockets*, Vol. 8, Mar. 1971, pp. 264–273, 10.2514/3.30258.

[11] P. Likins, *Dynamics and Control of Flexible Space Vehicles*. No. v. 105592 in Dynamics and Control of Flexible Space Vehicles, Jet Propulsion Laboratory, California Institute of Technology, 1969.

[12] H. Schaub and J. Junkins, *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston, VA: AIAA, second ed., 2009.

[13] G. Strang, *Linear Algebra and Its Applications*. fourth ed.

[14] M. Hochbruck and A. Ostermann, "Exponential Integrators," *Acta Numerica*, Vol. 19, May 2010, pp. 209–286, 10.1017/S0962492910000048.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An Operator Splitting Solver for Quadratic Programs," p. 40.

[17] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, Vol. 3, No. 1, 2010, pp. 1–122, 10.1561/2200000016.

[18] Z. Manchester and S. Kuindersma, "DIRTREL: Robust Nonlinear Direct Transcription with Ellipsoidal Disturbances and LQR Feedback," p. 9.

[19] P. Mitiguy, *Advanced Dynamics & Motion Simulation*. Motion Genesis, Aug. 2018.