Robust Aerocapture Guidance With a Method for Atmospheric Adaptation

Kevin Tracy
The Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
ktracy@cmu.edu

***Abstract*—Missions to Mars orbit must convert their high-energy hyperbolic orbits into desired elliptical orbits around the planet. Traditionally, this capture maneuver is performed with chemical propulsion, and requires a large amount of fuel. Aerocapture is an alternative method that dramatically reduces the fuel needed for Mars capture by utilizing drag from the Martian atmosphere to remove the energy from the spacecraft's orbit. This work presents a combined estimation and control strategy for aerocapture that is able to directly reason about the uncertainty in the atmosphere and plan a single control policy that minimizes a performance metric over multiple possible dynamics models. A square-root extended Kalman Filter is used to estimate atmospheric density parameters, and a convex predictor-corrector guidance strategy is used to solve for a common control policy that applies to multiple dynamics models sampled from the uncertainty of the estimated atmosphere.**

I. Introduction

Interplanetary spacecraft can arrive at the destination planet well in excess of the escape velocity. Traditionally, spacecraft use propulsion to reduce the energy of the spacecraft to that of a target orbit. Alternatively, the spacecraft can pass the destination planet close enough to the surface to use atmospheric drag to remove the necessary energy, a technique called aerocapture. By leveraging the atmosphere of the destination planet, aerocapture guidance is capable of targeting the desired orbit with significantly less fuel than propulsive-only options, resulting in a larger payload mass. These mass savings result in significantly larger and more sophisticated spacecraft payloads to explore new planets, which is critical given there is usually only one mission at a time. For example, using aerocapture instead of propulsive capture results in a factor of 3.8 times as much payload to orbit around Saturn's moon, Titan [1]. Aerocapture has not been performed yet due to the uncertainty in the atmospheric density, and the danger the spacecraft posed by this uncertainty. This work seeks to develop a method of aerocapture guidance that is able to reason about this atmospheric uncertainty as part of the planning process.

Early work on aerocapture guidance came in the form of missions to Mars. In [2], a guidance algorithm that uses the bank angle of the spacecraft as a means of controlling the atmospheric interaction was used to convert interplanetary trajectories from earth to Mars into elliptical orbits around Mars. The state-of-the art guidance algorithm for planetary aerocapture is the Fully Numerical Predictor-corrector Aerocapture Guidance algorithm, or FNPAG [3], [4]. Using a predictor-corrector framework similar to the entry vehicle versions FNPEG [5], and a convex entry vehicle variant CPEG [6], these guidance algorithms run forward simulations, and calculate corrections to the nominal control input based on the result of the predicted trajectories. In the case of FNPAG, it is shown that when there are terminal constraints specifying the target orbit after the atmospheric interaction, the optimal control policy is that of a bang-bang controller. This approach is then modified for use with alternative objective and constraint configurations [3].

Unfortunately, the Martian atmospheric density is only known to around a factor of 2 [7]. A recent paper has addressed this uncertainty by modifying FNPAG with the introduction of a cost term that penalizes state uncertainty [8]. This work proposes an aerocapture guidance algorithm based on convex optimization with the ability to reason about atmospheric uncertainty, the biggest risk involved in aerocapture. By estimating the density of the atmosphere and sampling over this distribution, a predictor-corrector guidance algorithm can solve for control commands that minimize an average cost function over the potential dynamics models. The aerocapture objective will be addressed by augmenting the state with the specific orbital energy of the spacecraft, and penalizing the average terminal error from a desired orbital energy that corresponds to the target orbit. The atmospheric density estimator will output an atmospheric profile, and an associated covariance. Atmospheric profiles will be sampled from this covariance using an Unscented Transform [9], [10]. This will spawn multiple dynamics models, and a convex predictor-corrector guidance framework will be used to minimize an average cost function for all the dynamics models, given one common control plan.

II. Vehicle Dynamics

Traditionally, the dynamics of an entry vehicle are described using spherical coordinates in something known as the "Vinh Model" [2], [11]–[16]. While highly nonlinear and prone to numerical scaling issues, this model has become the standard for simulation, guidance, and control of entry vehicles. Alternatively, the dynamics of the entry vehicle can be described in a planet-fixed Cartesian coordinate system. This approach has become standard within the powered-descent guidance

community for soft landing trajectory optimization [17]–[19]. The derive this model, first the assumptions must me made that the planet frame P spins relative to an inertial frame N with a constant angular velocity $\omega \in \mathbb{R}^3$. From here, the velocity of the entry vehicle with respect to the planet frame P is expressed as follows:

$$^{P}v = {}^{N}v - \omega \times r, \tag{1}$$

And this expression can be differentiated for the acceleration of the entry vehicle relative to the planet frame

$$^{P}a = {}^{N}a - 2(\omega \times {}^{P}v) - \omega \times (\omega \times r). \tag{2}$$

Now, the accelerations on the entry vehicle can be computed as $a_g \in \mathbb{R}^3$ for the acceleration due to gravity, $a_D \in \mathbb{R}^3$ for the drag acceleration, and $a_L \in \mathbb{R}^3$ for the Lift acceleration. These forces can be Incorporated into the Cartesian dynamics model, resulting in the following equations of motion:

$$\begin{bmatrix} v \\ a \end{bmatrix} = \begin{bmatrix} 0 & I \\ -[\omega\times]^2 & -2[\omega\times] \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ a_g + a_D + a_L \end{bmatrix}, \tag{3}$$

where $r \in \mathbb{R}^3$ is the position of the entry vehicle in the planet frame, $v \in \mathbb{R}^3$ is the velocity with respect to the the planet frame, and $a \in \mathbb{R}^3$ is the acceleration with respect to the planet frame. All three of these vectors are also expressed in coordinates of the planet frame. The cross product matrix, or skew matrix $[\omega\times]$ is defined as the following,

$$[\omega\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{4}$$

The gravitational force is modeled using a point mass representation of the planet, and is expressed as a function of the standard gravitational parameter of the planet, $\mu \in \mathbb{R}$:

$$a_g = -\frac{\mu}{\|r\|^3} r. \tag{5}$$

The drag acceleration is modeled as a function of the coefficient of drag $C_d \in \mathbb{R}$, the cross sectional area of the entry vehicle $A \in R$, and the atmospheric density $\rho \in \mathbb{R}$:

$$a_D = -\frac{1}{2m}\rho A C_d \|v\| v, \tag{6}$$

The atmospheric density is nominally described as an exponential model, and will be detailed further in the estimation section of this paper [16].

To define the lift force, the E frame will defined such that e_1 and e_2 form a basis for the plane orthogonal to the direction of the velocity. These two basis vectors can be described concretely as the following

$$\hat{e}_1 = \frac{r \times v}{\|r \times v\|}, \tag{7}$$

$$\hat{e}_2 = \frac{v \times \hat{e}_1}{\|v \times \hat{e}_1\|}. \tag{8}$$

This frame is also described graphically in figure 1. The magnitude of the lift acceleration is described in a similar way

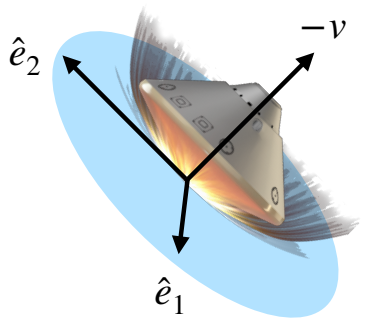

Fig. 1: Graphical description of the E frame. The e_1 and e_2 vectors are orthogonal to the velocity vector, and the lift vector for the entry vehicle can be described using these two basis vectors.

to the drag acceleration, but with a coefficient of lift $C_L \in \mathbb{R}$ instead of drag,

$$\|L\| = \frac{1}{2m} C_L \rho(r) A \|v\|^2. \tag{9}$$

From here, the acceleration due to lift is a function of the bank angle $\sigma \in \mathbb{R}$, where the entry vehicle can rotate the lift vector about it's velocity vector. This acceleration is defined by using the E frame basis vectors:

$$a_L = \|L\|(\sin(\sigma)\hat{e}_1 + \cos(\sigma)\hat{e}_2). \tag{10}$$

III. Atmospheric Estimation

The atmospheric density is the largest source of uncertainty in the dynamics models for entry vehicles. Similar to the Earth's atmosphere, the atmosphere on Mars has a density profile that changes based on time of year, weather, solar weather, and unlike earth, even depends highly on dust storms on the planet. Since there is no method of real time observation of this atmosphere, most of the data used to recreate the atmosphere model comes from data collected by previous entry vehicle missions. In order to increase the fidelity of the guidance and control during entry, the vehicle must adapt to the uncertain atmospheric density profile. To accomplish this, a square-root Extended Kalman Filter (EKF) will be used to estimate a correction factor to be applied to a nominal exponential atmosphere.

A. Square Root Extended Kalman Filter

A Square Root Extended Kalman Filter (SREKF) is a variant of the EKF where all of the covariance matrices are stored and manipulated in the form of their Cholesky factorizations. In the case of infinite precision arithmetic, it is equivalent to the EKF, but with floating point precision is able to achieve the same numerical precision as the EKF with half as many bits.

Of the available interpretations of a matrix square root, this paper will look at those with an upper triangular structure as follows:

$$M = U^T U, \tag{11}$$

Where we will define the square root of the matrix as $\sqrt{M} = U$. For positive definite matrices, this upper triangular matrix square root can be found using the Cholesky factorization. As discussed in [20], a QR decomposition on stacked matrix square roots results in the square root of the addition of the origin matrices. For ease of notation, this operation of taking the QR decomposition and returning the upper triangular R will be expressed as:

$$\sqrt{A+B} = qr_r(\sqrt{A}, \sqrt{B}). \tag{12}$$

The following covariances can be factored: the state covariance $\Sigma = F^T F$, innovation covariance $S = G^T G$, process noise covariance $Q = \Gamma_Q^T \Gamma_Q$, and sensor noise covariance $R = \Gamma_R^T \Gamma_R$. From here, the SREKF can then be derived by factoring each step of EKF into these two factors, and taking the qr_r of the factor to result in an upper triangular matrix square root. The resulting algorithm is detailed in algorithm 1

Algorithm 1 Square Root Extended Kalman Filter

function srekf($\mu_{t|t}, F_{t|t}, u_t, y_{t+1}, \Gamma_Q, \Gamma_R$)
  // prediction
  $A = \partial f / \partial x|_{\mu_{t|t}, u_t}$
  $\mu_{t+1|t} = f(\mu_{t|t}, u_t)$
  $F_{t+1|t} = qr_r(F_{t|t} A^T, \Gamma_Q)$
  // innovation
  $C = \partial g / \partial x|_{\mu_{t+1|t}, u_t}$
  $z = y_{t+1} - g(\mu_{t+1|t})$
  $G = qr_r(F_{t+1|t} C^T, \Gamma_R)$
  $L = [G^{-1}(G^{-T} C) F_{t+1|t}^T F_{t+1|t}]^T$
  // update
  $\mu_{t+1|t+1} = \mu_{t+1|t} + Lz$
  $F_{t+1|t+1} = qr_r(F_{t+1|t}(I - LC)^T, \Gamma_R L^T)$
  return($\mu_{t+1|t+1}, F_{t+1|t+1}$)

B. Atmospheric Estimation

Traditionally, the density of a planetary atmosphere like that on Earth and Mars can be described by an exponential function of the following form [16], [21], [22]:

$$\rho(a) = \rho_0 \exp(-\frac{a}{H}), \tag{13}$$

where $\rho_0 \in \mathbb{R}$ is the density at the surface of the planet, $a \in \mathbb{R}$ is the altitude, and $H \in \mathbb{R}$ is the scale height. It is not abnormal for the empirical density of during entry to deviate from this nominal exponential atmosphere by a factor of 2 or more. As a result, the most numerically robust method for estimating density information online is not to estimate the density of the atmosphere, but instead estimate a correction factor to be applied to this exponential model. The following corrected atmospheric density can be described using a correction term $k_\rho \in \mathbb{R}$:

$$\rho(a) = (1 + k_\rho) \cdot \rho_0 \exp(-\frac{a}{H}). \tag{14}$$

By augmenting the state of the entry vehicle to include this k_ρ term, the SREKF is able to converge on an estimate for this correction factor. The performance of the filter is shown in figure 2. The filter has trouble estimating the true value of k_ρ at the higher altitudes, since the atmospheric density is so small that it falls below the noise floor. However, when the entry vehicle descends to lower altitudes where there is enough density that the drag and lift vectors produce an discernible change in the motion of the entry vehicle, the filter is able to quickly converge on the true k_ρ, with a very tight variance around that estimate.

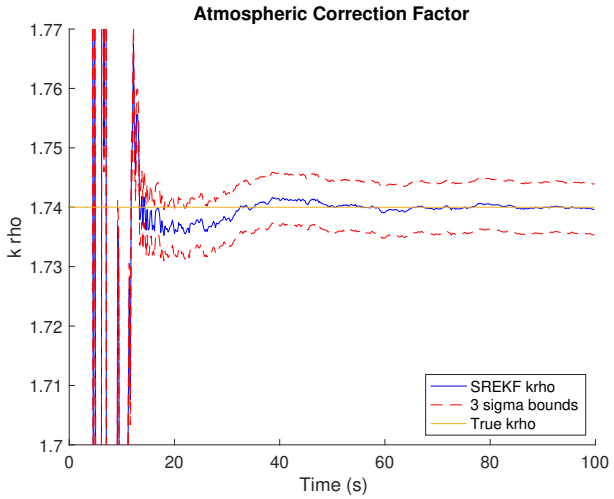

Fig. 2: SREKF estimating the k_ρ during an aerocapture maneuver. The filter has trouble estimating the true value of k_ρ until it descends into an altitude with a significant enough atmospheric density such that the drag and lift accelerations are measureable.

IV. Control

The guidance and control aspect of the aerocapture maneuver takes into account information from the estimator about the atmospheric density profile, and samples that distribution to create multiple candidate dynamics models. A convex predictor-corrector algorithm is then be used to solve for a common control policy that minimizes the average cost function for the candidate dynamics models. One approach is to take the variance of the k_ρ parameter in the estimator, and sample three atmospheres from this parameter, one that uses the estimate k_ρ, $k_\rho + \sigma_{k_\rho}$, and $k_\rho - \sigma_{k_\rho}$. These dynamics models will be referred to as f_1, f_2, and f_3 respectively. Since we expect that the atmospheric uncertainty decreases as the altitude decrease, instead of holding this correction factor constant throughout the whole altitude profile, this correction factor will decay down to a small number ϕ 0 at the surface of the planet. This altitude varying correction factor looks like the following:

$$k_\rho(a) = k_{\rho,t} \cdot \exp(-\frac{1}{a_t}(a - a_t) \ln \frac{\phi}{k_{\rho,t}}), \tag{15}$$

where $a_t \in \mathbb{R}$ is the current altitude where the estimate is from, and $k_{\rho,t}$ is the current estimate. This adjusted density

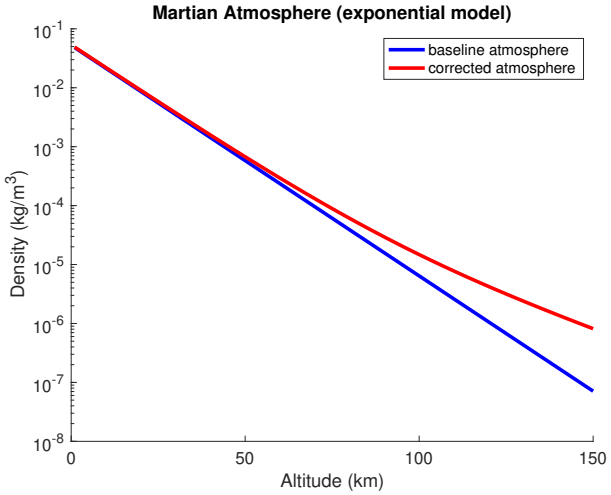

Fig. 3: Density of the Martian atmosphere as a function of altitude. Shown is a baseline model, as well as an example modified density model.

profile is shown next to the nominal exponential density profile in figure 3 In order to build an appropriate cost function for aerocapture manuevers, the specific orbital energy of the entry vehicle will be appended to the state.

$$\epsilon = \frac{v^2}{2} - \frac{\mu}{r} \tag{16}$$

The derivative of this specific orbital energy with respect to time is the following:

$$\dot{\epsilon} = \langle \dot{v} + \omega \times v, v + \omega \times r \rangle - \frac{\mu \cdot \langle v, r \rangle}{r^3}. \tag{17}$$

This allows us to specify the target orbit by specifying the terminal specific orbital energy. From here, three dynamics models are then created by sampling the atmospheric correction factor k_ρ from the estimator, and the stacked trajectory optimization problem is solved, where one common policy applies to all three dynamics models, and the cost function is the average terminal specific orbital energy error, and a regularization on the controls.

$$\begin{aligned} \underset{X_1, X_2, X_3, U}{\text{minimize}} \quad & \sum_{i=1}^{3} (\epsilon_{N,i} - \epsilon_f)^2 + \gamma \|U\|^2 \\ \text{subject to} \quad & x_{i,0} = x_0 \qquad \forall i, \\ & f_1(x_{1,k}, u_k) = x_{1,k+1} \quad \forall k, \\ & f_2(x_{2,k}, u_k) = x_{2,k+1} \quad \forall k, \\ & f_3(x_{3,k}, u_k) = x_{3,k+1} \quad \forall k, \\ & \underline{u} \leq u_k \leq \bar{u} \qquad \forall k \end{aligned} \tag{18}$$

This optimization problem is non-convex due to the nonlinear dynamics constraints, but can be solved by Non-Linear Programming (NLP) solver like IPOPT [23] and SNOPT [24]. Alternatively, a convex predictor-corrector algorithm as described in [6] can be used to solve the non-convex control problem by predicting a trajectory with forward simulation, then solving a convex correction problem linearized about the prediction. This algorithm is detailed in algorithm 2. Where the convex

Algorithm 2 Convex Predictor-corrector Algorithm

```
input x_0, U                                  ▷ nominal control plan
while ||δU|| > tolerance do
    X̄, Ū = simulate(x_0, U)                   ▷ predict trajectory
    A, B = linearize(X̄, Ū)                    ▷ linearize about prediction
    δX, δU = cvx(X̄, Ū, A, B)                  ▷ solve for correction
    U += δU                                   ▷ correct control plan
end while
return U                                      ▷ return updated control plan
```

correction problem solves for the update to the control policy by linearizing the three dynamics models about their predicted trajectories, and minimizing the same cost function as with the NLP formulation. This convex optimization problem is described in 19.

$$\begin{aligned} \underset{\delta X_1, \delta X_2, \delta X_3, \delta U}{\text{minimize}} \quad & \sum_{i=1}^{3} (\delta\epsilon_{N,i} - \tilde{\epsilon}_{f,i})^2 + \gamma \|U + \delta U\|^2 \\ \text{subject to} \quad & \delta x_{i,0} = 0 \qquad \forall i, \\ & A_{1,k}\delta x_{1,k} + B_{1,k}\delta u_k = \delta x_{1,k+1} \quad \forall k, \\ & A_{2,k}\delta x_{2,k} + B_{2,k}\delta u_k = \delta x_{2,k+1} \quad \forall k, \\ & A_{3,k}\delta x_{3,k} + B_{3,k}\delta u_k = \delta x_{3,k+1} \quad \forall k, \\ & \delta\underline{u} \leq \delta u_k \leq \delta\bar{u} \qquad \forall k \end{aligned} \tag{19}$$

where $\tilde{\epsilon}_{f,i} = \epsilon_f - \epsilon_{N,i}$. The convex predictor-corrector combination was run on a sample aerocapture case, where the three dynamics models were created using a $\pm 10\%$ distribution on the atmospheric correction factor $k\rho$. Given the same initial conditions for all three models, the bank angle control policy that minimizes the terminal specific orbital energy error was solved for by solving the inner loop convex optimization problems with a custom primal-dual interior point method [25], [26]. The resulting altitude trajectories for the three models is shown in figure 4, where there is a slight difference in the altitude profiles. The specific orbital energy histories are shown in figure 5, where despite the differences in the three trajectories, they were all able to finish with the target energy. The common bank angle control history for these three models is shown in figure 6, and this control plan was applied to all three of the dynamics models.

V. Discussion

This paper presented a method for controlling an entry vehicle during an aerocapture maneuver that paired an atmospheric estimation strategy with a robust trajectory optimization algorithm to achieve a target orbit in the presence of atmospheric uncertainty. The atmospheric estimation was done by using a SREKF to estimate a correction factor modifying a baseline exponential atmosphere to fit the measurements. This correction factor was then sampled for the $\pm\sigma$ case and a robust trajectory optimization problem was formed where a common control policy would apply to all three dynamics

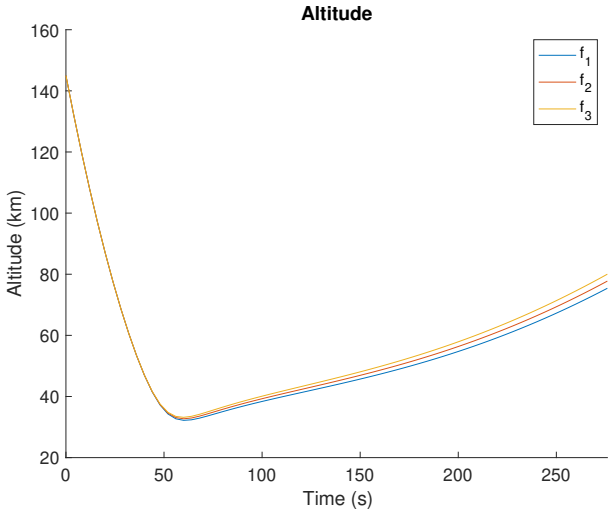

Fig. 4: Altitude histories for the three trajectories given the common bank angle control policy. Despite slight deviations, all three trajectories are within 5 km of each other.

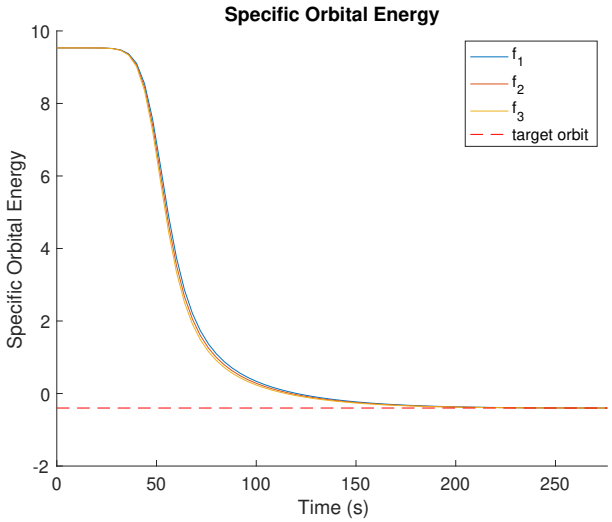

Fig. 5: Specific Orbital Energy for the three trajectories given the common bank angle control policy. Despite differences in the dynamics models, the desired specific orbital energy is achieved at the end of all three trajectories.

models, and the cost function was an average cost amongst the three trajectories. This trajectory optimization problem was solved using a convex predictor-corrector method, where the nominal control policy is used to simulate the trajectories, and then a convex correction problem is solved by linearizing the dynamics about the predicted trajectory.

My main takeaway from this project was how difficult the control problem was to solve. Due to the ill-conditioning present in the units scaling for these problems (you slow down very quickly), the QP solver I was using OSQP [27] was taking 20,000 iterations to solve. In order to solve it more reliably, I wrote my own primal-dual interior point method with centring-correcting steps [25], [26], which had far better convergence

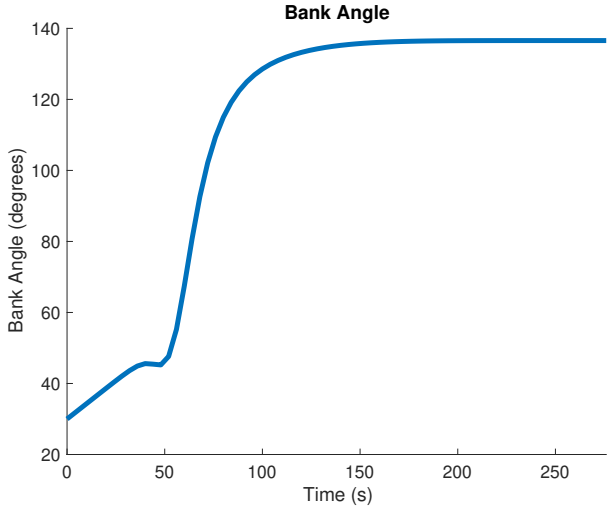

Fig. 6: The optimal bank angle control policy that is executed for all three trajectories.

behavior. After I did this, the problem converged after 5-10 iterations every-time the predictor-corrector was called. This is mostly due to the affine-invariance of Newton's method, and how first-order methods like ADDM (used in OSQP) are very prone to poor scaling and often need preconditioning for reliable convergence. Another thing I learned was the difficulty in estimating the atmospheric properties. My first thought was to estimate the scale height H (inside the exponential), but this turned out to be extremely numerically sensitive because tiny changes in H result in orders of magnitude in the density at higher altitudes. As a result, it became clear that I need something to estimate that was more physical, and the atmospheric correction factor k_ρ ended up working well.

Future work includes testing this combined estimation and control strategy on a range of initial conditions, and testing it on more realistic atmospheric conditions. This is possible with a tool known as MARSGRAM [7], but since it's written in Fortran 90, it is challenging to build and use. I imagine that through this testing, there will have to be some changes to the cost function in order to ensure robustness, and I will likely have to augment the current setup with more tuning parameters.

References

[1] J. L. Hall, M. A. Noca, and R. W. Bailey, "Cost-Benefit Analysis of the Aerocapture Mission Set," *Journal of Spacecraft and Rockets*, May 2012.

[2] N. Vinh, W. Johnson, and J. Longuski, "Mars aerocapture using bank modulation," in *Astrodynamics Specialist Conference*, Denver,CO,U.S.A.: American Institute of Aeronautics and Astronautics, Aug. 2000.

[3] P. Lu, C. J. Cerimele, M. A. Tigges, and D. A. Matz, "Optimal Aerocapture Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 553–565, Apr. 2015.

[4] D. A. Matz, P. Lu, G. F. Mendeck, and R. R. Sostaric, "Application of a Fully Numerical Guidance to Mars Aerocapture," in *AIAA Guidance, Navigation, and Control Conference*, Grapevine, Texas: American Institute of Aeronautics and Astronautics, Jan. 2017.

[5] C. W. Brunner and P. Lu, "Skip Entry Trajectory Planning and Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1210–1219, Sep. 2008.

[6] K. Tracy and Z. Manchester, "CPEG: A Convex Predictor-corrector Entry Guidance Algorithm," in *IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2022, p. 10.

[7] H. L. Justh and K. L. Lee, "Mars-GRAM 2010: Additions and Resulting Improvements," in *International Planetary Probe Workshop*, 17-21 Jun. 2013, United States, Jun. 2013.

[8] C. R. Heidrich, M. J. Holzinger, and R. D. Braun, "Optimal Information Filtering for Robust Aerocapture Trajectory Generation and Guidance," *Journal of Spacecraft and Rockets*, Oct. 2021.

[9] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153–158.

[10] R. Van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," vol. 6, IEEE, 2001, pp. 3461–3464.

[11] A. Busemann, N. Vinh, and R. Culp, "Hypersonic Flight Mechanics," Tech. Rep. NASA-CR-149170, Sep. 1976, p. 440.

[12] N. X. Vinh, A. Busemann, and R. D. Culp, "Hypersonic and planetary entry flight mechanics," *NASA STI/Recon Technical Report A*, vol. 81, p. 16 245, Jan. 1980.

[13] Z. Wang and M. J. Grant, "Near-Optimal Entry Guidance for Reference Trajectory Tracking via Convex Optimization," American Institute of Aeronautics and Astronautics, Jan. 2018.

[14] ——, "Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization," in *AIAA Scitech 2019 Forum*, San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 2019.

[15] P. Lu, "Entry Guidance: A Unified Method," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 713–728, May 2014.

[16] P. Gallais, *Atmospheric Re-Entry Vehicle Mechanics*. Berlin ; New York: Springer, 2007.

[17] L. Blackmore, B. Açikmeşe, and J. M. Carson, "Lossless convexification of control constraints for a class of nonlinear optimal control problems," in *2012 American Control Conference (ACC)*, Jun. 2012, pp. 5519–5525.

[18] B. Acikmese and S. R. Ploen, "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, Sep. 2007.

[19] B. Acikmese, J. M. Carson, and L. Blackmore, "Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, Nov. 2013.

[20] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: A Fast Solver for Constrained Trajectory Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019.

[21] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer New York, 2014.

[22] O. Montenbruck, E. Gill, and F. Lutze, "Satellite Orbits: Models, Methods, and Applications," *Applied Mechanics Reviews*, vol. 55, no. 2, B27, 2002.

[23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[24] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[25] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," in *Optimization Engineering*, 2012, pp. 1–27.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*, Second. Springer, 2006.

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