AAS 17-790

APPROXIMATE-OPTIMAL FEEDBACK GUIDANCE FOR SOFT LUNAR LANDING USING GAUSSIAN PROCESS REGRESSION

Pradipto Ghosh*, James Woodburn†, and Cody Short ‡

A recently-developed optimal feedback synthesis method based on Gaussian Process Regression (GPR) is applied to soft lunar landing guidance. GPR is a form of supervised learning technique that is highly useful in constructing surrogate models of unknown functions from input-output training dataset. In this work, GPR has been utilized in capturing the functional relationship between optimal state and control using a pre-generated field of extremals as training data. At each guidance call, when control computation is desired for a newly-sensed state, a new Gaussian process model regressing state and control is created with only a subset of the offline-computed training data, those that are "temporally similar" to the current state. It is argued that this method of sequentially generating approximately-optimal controls from a new regression model at each step effectively relaxes the assumption that the underlying map is smooth over the domain of interest. Having designed the GPR-based optimal state-feedback algorithm, its usefulness is assessed by verifying that its application leads to near-optimal trajectories when the lander starts from perturbed initial conditions. A distinctive feature of this work is the realistic quantification of the initial state uncertainty in the form of a full position-velocity estimation error covariance matrix obtained from lunar orbit determination. By randomly sampling states within the extent of this uncertainty, it is shown through numerical experiments that the GPR-based guidance algorithm is highly effective in compensating for imperfectly-known initial conditions of the lander.

INTRODUCTION

As interest in lunar missions thrives unabated from space agencies around the world, the problem of designing improved autonomous guidance strategies for lunar soft landing, or the somewhat related problem of near-zero-velocity touchdown on asteroids and other atmosphere-free bodies, continues to receive close attention from the global research community. In early 2017, SpaceX announced its intention of sending a crewed Dragon 2 vehicle to a lunar orbit sometime in 2018 [1]. At the time of writing, China is planning to fly a robotic lunar sample return mission, the so-called Chang'e 5 mission, in late 2017 [2]. The Indian Space Research Organization's (ISRO) upcoming Chandrayaan-2, planned for launch in 2018, will include a lunar orbiter, lander, and rover for lunar surface exploration [3]. In a similar vein, the Japan Aerospace Exploration Agency (JAXA) is developing the Smart Lander for Investigating Moon (SLIM), a lander that is being designed to perform precise, pin-point landing on the lunar surface, powered by an image-based navigation system [4].

*Astrodynamics Engineer, Analytical Graphics Inc., 220 Valley Creek Blvd., Exton, PA, 19341-2380.
†Technical Director of Development, Analytical Graphics Inc., 220 Valley Creek Blvd., Exton, PA, 19341-2380.
‡Astrodynamics Software Engineer, Analytical Graphics Inc., 220 Valley Creek Blvd., Exton, PA, 19341-2380.

The landing phase for a probe is generally initiated from a low lunar parking orbit. Soft landing typically implies a vertical velocity of less than ~ 5 m/s and near-zero horizontal velocity on or close to the lunar surface. The vertical velocity requirement ensures that the structural stress on the landing mechanism remains under acceptable levels, whereas a small horizontal velocity reduces the risk of the craft tipping over on touchdown. In the absence of any appreciable drag forces to dissipate the craft's kinetic energy, it is clear that the desired terminal conditions can only be achieved by selecting an appropriate thrust program, a process called powered braking. An enormous volume of work analyzes various lunar soft landing guidance and control schemes, a consequence of investigations that began in the pre-Apollo era [5]. It is not the intent of the present paper to provide an exahustive survey of the relevent literature. For this purpose, reference may be made to some of the more recent papers, such as those those by McInnes [6], Ramanan and Lal [7], Chomel and Bishop [8], Lee [9], Zhou *et al.* [10], and Cecchetti *et al.* [11], and the references therein. The work in reference [6] proposes modulating of the instantaneous vehicle speed as a function of the velocity pitch angle along a gravity turn descent trajectory, but does not adopt a control theoretic approach. The velocity pitch angle is assumed to be independently regulated, and the trajectory is not optimal in any sense. Ramanan and Lal present analyses of various minimum-fuel, open-loop thrust programming strategies for vertical soft landing from different initial orbits, but they do not address the feedback guidance problem, which is the focus of this paper. Chomel and Bishop develop a reference trajectory generation algorithm as well as a guidance algorithm for the real-time tracking of this reference trajectory. The authors use Lyapunov stability analysis to demonstrate that the guidance law is able to track the reference states asymptotically. However, their formulation does not utilize the optimal control framework. In reference [9], analytical expressions are derived for a control-authority-minimizing guidance law based on a linear model of the lander dynamics. The research reported by Zhou et al. in reference [10] is qualitatively similar to that presented in this paper in that both involve synthesizing closed-loop optimal controls from open-loop ones, but there are two main differences between that work and the current one. First, the success of the feedback synthesis method followed in [10] is contingent upon the dynamical system model and the cost function having special structures, namely affine-nonlinear dynamics and a quadratic cost function. The applicability of the feedback synthesis method presented in this research, on the other hand, is more universal because it does not place such a restriction on the problem structure. Secondly, while Zhou *et al.* use spline interpolation to reconstruct a Riccati-like gain matrix for off-nominal states, the present research utilizes sequential Gaussian Process Regression (s-GPR), a variant of the framework developed in [12, 13] to approximate controls as a function of the current state. Lastly, Cecchetti, Pontani and Teofilatto apply a modified neighboring optimal guidance method to solve the lunar soft landing problem. Their modification consists in updating the so-called "sweep method" so as to avoid singularities in the elements of the associated gain matrices.

In this paper, the lunar soft landing guidance problem is solved by adopting a particular implementation of dynamic programming, namely optimal feedback synthesis [14, Ch. 4]. Feedback synthesis is computationally much more tractable than attempting a numerical solution of the Hamilton Jacobi Bellman (HJB) equation for determining optimal feedback policies. In optimal feedback synthesis, a feedback chart is constructed from a family of open-loop extremals, thus ensuring optimality with respect to any state vector interior to the convex hull of the set of states in that family. In this paper, feedback synthesis is realized using sequential GPR, a powerful non-parametric regression technique, where the control corresponding to each guidance-call state is approximated by a Gaussian regression model valid only within a neighborhood of that state sample. An often-adopted approach to testing the effectiveness of guidance laws is to verify that they can accomodate reasonable uncertainty in the initial conditions of the system. While most of the papers cited earlier arbitrarily assume the size of an uncertainty volume surrounding the nominal initial state without justifying how this was arrived at, this paper, notably, uses the full position-velocity covariance of a lunar orbiter as the uncertainty region around the nominal de-orbit state. The covariance function is computed as part of the lunar orbit determination solution using an Extended Kalman Filter (EKF) operating on a high-fidelity force model and simulated observations from the NASA/JPL Deep Space Network (DSN).

PROBLEM DESCRIPTION

The point-mass lunar probe is initially assumed to be in an eccentric, polar, parking orbit with the following nominal Keplerian orbit parameters referred to a Moon-centric inertial coordinate system:

$$a = 1837.1\text{km}, \;\; e = 0.0272, \;\; i = 90^\circ, \;\; \Omega = 120^\circ, \;\; \omega = 90^\circ, \;\; f = -90^\circ \tag{1}$$

where a, e, i, Ω, ω and f are the semi-major axis, eccentricity, inclination, right ascension of the ascending node, argument of periapsis, and true anomaly, respectively. The objective is to design a state-feedback control law that brings the lander "close" to the lunar surface in minimum time with (ideally) zero vertical and horizontal velocities. No constraint is placed on the landing site location; it is allowed to be anywhere in the initial orbit plane. For the purposes of optimal control design, the following two-body, planar, spherical-gravity dynamical equations of motion of the lander powered by a constant-specific-impulse (CSI) engine are considered:

$$\dot{r} = u \tag{2}$$

$$\dot{u} = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T}{m_0 - |\dot{m}|t}\sin\beta \tag{3}$$

$$\dot{v} = -\frac{uv}{r} + \frac{T}{m_0 - |\dot{m}|t}\cos\beta \tag{4}$$

with initial conditions:

$$r_0 = \frac{a(1-e^2)}{1+e\cos f_0} \tag{5}$$

$$u_0 = \sqrt{\frac{\mu}{p}}e\sin f_0 \tag{6}$$

$$v_0 = \sqrt{\frac{\mu}{p}}(1+e\cos f_0) \tag{7}$$

and final conditions:

$$r_f - R_M = 10\text{ km} \tag{8}$$

$$u_f = 0 \tag{9}$$

$$v_f = 0 \tag{10}$$

minimizing the Mayer cost function:

$$\mathcal{J} = t_f \tag{11}$$

In the above differential-algebraic system of equations Eqs. (2 - 11), r denotes the radial distance of the vehicle center of mass from the moon center, u the radial (vertical) component of the velocity,

v the tangential (horizontal) component of the velocity, T the engine thrust, β the thrust pointing angle measured counter-clockwise from the local horizontal, f_0 the true anonaly at braking initiation and $R_M = 1737.1$km. the mean lunar radius. A true soft landing usually requires that the craft be vertical at touchdown. The above model, however, does not account for the vehicle's yaw kinematics, nor does it capture the operation of a Reaction Control System (RCS) for orienting the thrust pointing angle, which will, in general be different from $90°$ when the terminal manifold is reached. As such, the final altitude condition is selected to allow for a second, attitude re-orienting phase to be initiated if one is deemed necessary. The following values are used for the (constant) engine thrust T, initial mass m_0 and the engine specific impulse I_{sp} [7]:

$$T = 440 \text{ N}, \; m_0 = 300 \text{ kg.}, \; I_{\text{sp}} = 310 \text{ seconds}$$

Figure (1) shows the solution to the *open-loop* trajectory optimization problem described by Eqs. (2 - 11) obtained using GPOPS-II, a commercially available optimal control software [15]. GPOPS-II uses Gauss pseudospectral collocation for discretizing the dynamcal equations and Interior Point OPTimizer (IPOPT) for solving the resulting numerical optimization problem. It is clear from Fig. (1) that the thrust pointing angle starts close to $180°$, and remains in the $[120°, 180°]$ range throughout, implying that the craft retro-thrusts over the entire duration to diminish altitude and neutralize the velocity components. It may be recalled that for the current problem geometry, $\beta = 0°$ is along-motion and $\beta = 180°$ is the anti-motion direction. As noted already, the control β numerically solved for is open-loop in the sense that it is a function of the initial states and current time, i.e $\beta = \beta(r_0, u_0, v_0, t)$. This means that if the system starts from an off-nomal state, this control will not be optimal with respect to that state and will not meet the algebraic constraints.

SOLUTION METHODOLOGY

As alluded to earlier, the focus of this paper is the design of a guidance strategy that will drive the system states from an initial uncertainty hypervolume, realized by a complete position-velocity ellipsoid obtained from lunar orbit determination (OD), to the desired terminal hypersurface while honoring other optimality criteria as closely as possible. It was further indicated that the computation of such a guidance strategy begins with generating a field of open-loop extremals uniformly covering this uncertainty region so that any random, nature-selected initial state interior to this region can be autonomously acted upon by the s-GPR guidance algorithm/feedback controller. Lunar orbit determination was carried out in this research using the Orbit Determination Toolkit (ODTK) software developed by Analytical Graphics Inc. [16]. The envisioned workflow can be summarized in the following steps:

1. Select a de-orbit point on the parking orbit from general mission considerations.
2. Generate, offline, a field of extremals with initial conditions sampled from the interior of an $n\sigma$ ellipsoid around this nominal point. This step involves solving multiple instances of the trajectory optimization problem described by Eqs (2 - 11).
3. Synthesize real-time controls from the extremals generated in step 2 using s-GPR.

In the sequel, the concept of optimal feedback synthesis is briefly explained, followed by details of its implementation using s-GPR. Subsequently, an outline of the lunar orbit determination process relevant to the current work is presented.

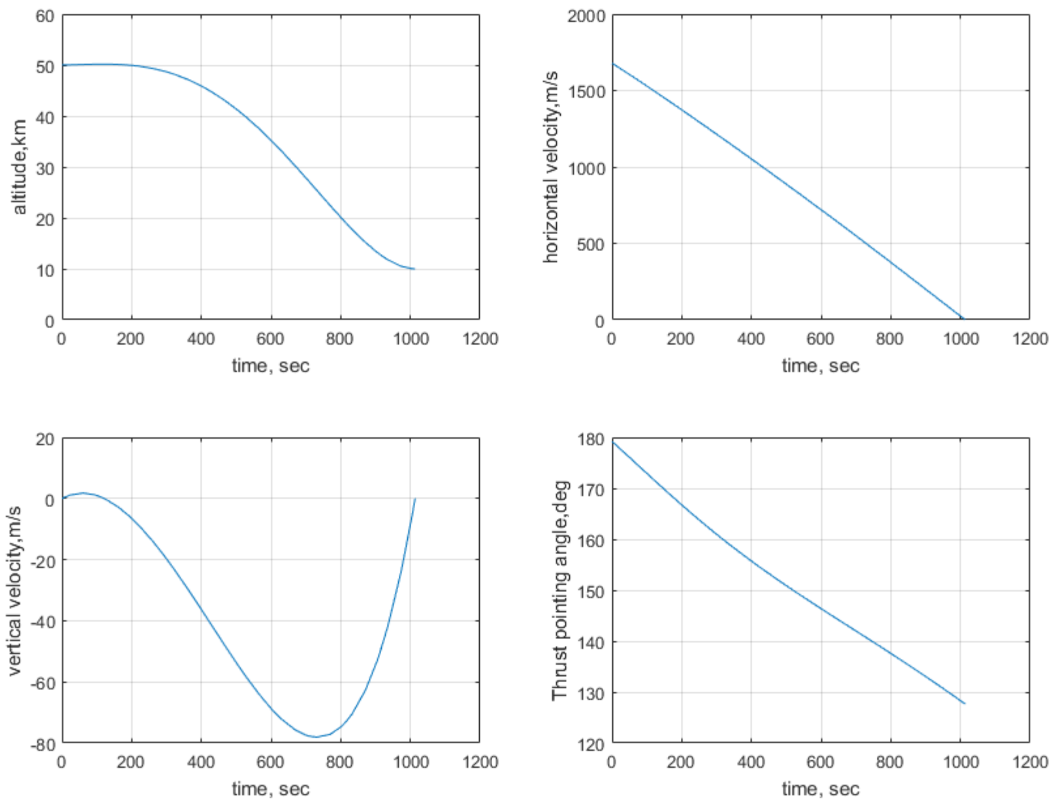

Figure 1. Minimum-time trajectories for the lander

Optimal Feedback Synthesis

Consider the optimal control problem $\mathcal{P}$ described by:

$$\mathcal{P}\begin{cases}\dot{\mathbf{x}}=\boldsymbol{F}(t,\mathbf{x}(t),\mathbf{u}(t)), \quad \mathbf{x}(0)=\mathbf{x0}, \quad t\geq 0\\ \mathbf{u}(t)=\boldsymbol{\gamma}(t,\mathbf{x}(t))\in\mathcal{U}, \quad \boldsymbol{\gamma}\in\Gamma\\ \mathcal{J}(\mathbf{u})=\phi(t_f,\mathbf{x}(t_f))+\int_0^{t_f}\mathcal{L}(t,\mathbf{x}(t),\mathbf{u}(t))dt\\ \boldsymbol{\Psi}(t_f,\mathbf{x}(t_f))=\mathbf{0}\end{cases}$$

where $\mathbf{x}\in\mathbb{R}^n$ is the state, $\mathbf{u}\in\mathbb{R}^m$ is the control, $\boldsymbol{F}:\mathbb{R}^+\times\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^n$ is a known smooth, Lipschitz vector function (the system function), $\boldsymbol{\gamma}:\mathbb{R}^+\times\mathbb{R}^n\to\mathbb{R}^m$ is the feedback strategy, $\mathcal{U}\subseteq\mathbb{R}^m$ and Γ is the class of all admissible feedback strategies. The control function or control program $\mathbf{u}$ must be chosen to minimize the Bolza-type cost-functional $\mathcal{J}(\cdot)$, and $\boldsymbol{\Psi}:\mathbb{R}^+\times\mathbb{R}^n\to\mathbb{R}^q, q\leq n$ represents a smooth terminal manifold in the state space. Adopting a feedback information pattern for $\mathcal{P}$, the optimal feedback control problem is to find a $\boldsymbol{\gamma}^*\in\Gamma$ such that:

$$J(\boldsymbol{\gamma}^*)\leq J(\boldsymbol{\gamma}) \ \ \forall\,\boldsymbol{\gamma}\in\Gamma \tag{12}$$

where $J(\boldsymbol{\gamma})\triangleq\mathcal{J}(\mathrm{u})$ with $\mathbf{u}(t)=\boldsymbol{\gamma}(t,\mathbf{x})$. The function $\mathbf{u}^*$ minimizing $\mathcal{J}(\cdot)$ is the optimal open-loop strategy while $\boldsymbol{\gamma}^*$ is referred to as the *feedback realization* of $\mathbf{u}^*$. Conversely, it also holds that the open-loop solution $\mathbf{u}^*$ of $\mathcal{P}$ is a representation of the feedback strategy $\boldsymbol{\gamma}^*$. This equivalence between open-loop and feedback strategies can be summarized with the following two statements:

1. Both $\mathbf{u}^*$ and $\boldsymbol{\gamma}^*$ generate the same unique, admissible, state trajectory.
2. They both have the same value(s) on this trajectory.

Given a set $V \triangleq \{\mathbf{x0} \in \prod_{i=1}^{n}[x0_i - a_i, x0_i + b_i] \subset \mathbb{R}^n\}$ of box-uncertain initial conditions, and an associated family $\mathcal{Q}(\mathbf{x0})$ of optimal programming problems parameterized by $\mathbf{x0} \in V$, the procedure for synthesizing optimal strategies for $\mathcal{Q}$ in feedback form now directly follows from the above discussion:

1. First construct the set U_{OL} (of finite cardinality, as opposed to $\mathcal{Q}$) that are strategies that depend only on the initial state $\mathbf{x}_0$ and the current time t. This can be achieved by solving for open-loop control programs and trajectories for a sample of initial conditions from within V. This process will also generate a corresponding set of open-loop states X_{OL} and a set of final times, or times-to-go from $\mathbf{x}_0$, T^0_{2go} . The collection of extremals $\{X_{OL}, U_{OL}\}$ will be referred to as the nominal ones, and can be obtained numerically using a direct or indirect formulation of the optimal trajectory problem.
2. Given an off-nominal state, and perhaps also the current time (for a non-autonomous system), *predict* the open-loop control that would be the optimal open-loop strategy for that state at that time. Such an open-loop control would also constitute, by the equivalence-of-strategies argument presented above, an equivalent feedback strategy $\widetilde{\boldsymbol{\gamma}}^*(t, \mathbf{x})$, with the feedback information pattern being enforced by the use of current state (and time) in the prediction scheme. Under the adopted assumption of the existence of a family of unique, admissible state-control pairs $\{\mathbf{x}(t), \mathbf{u}(t)\}$ corresponding to $\boldsymbol{\gamma}(t, \mathbf{x}(t))$ for every $\mathbf{x}_0 \in V$, $\boldsymbol{\gamma}(t, \mathbf{x}(t))$ is also called a *control synthesis*, and $\widetilde{\boldsymbol{\gamma}}^*(t, \mathbf{x})$ the approximate control synthesis for the set V of initial conditions.

In this paper, prediction is realized by the use of *universal kriging*, an instance of Gaussian Process Regression, itself a form of supervised learning that can be used for approximating non-random, deterministic input-output models arising from computer experiments [17]. It is assumed that all the states are directly available for measurement for a full-state feedback controller synthesis.

Feedback Synthesis through Sequential Gaussian Process Regression (s-GPR)

A Gaussian process is a generalization of the familiar Gaussian distribution. Just as a Gaussian distribution is fully specified by its mean *vector* and covariance *vector*, a Gaussian process is characterized by its mean *function* and covariance *function* [17]. A function $u(\mathbf{x})$ distributed as a Gaussian process with mean function $\mu(\mathbf{x})$ and covariance $Z(\mathbf{x})$ can be expressed as:

$$u(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), C(\mathbf{x})) \tag{13}$$

Formally, a Gaussian process is a set of random variables any finite collection of which have joint Gaussian probability distributions. Gaussian process regression can be utilized to address the following problem: suppose there is a training set $\mathcal{S}$ of p observations $\mathcal{S} = \{\{\mathbf{x}_i, u_i\},\ \mathbf{x}_i \in \mathbb{R}^n,\ u_i \in$

$\mathbb{R},\ i = 1, \ldots, p\}$, where $\mathbf{x}$ is an input vector to a black box model, perhaps a computer experiment, and u denotes a dependent variable or the output. Given this training set of inputs and outputs, what inference can be drawn about the functional relationship between $\mathbf{x}$ and u, or in other words, the conditional distribution of the outputs given the inputs? Thus, GPR can be exploited to *learn the mapping between* between $\mathbf{x}$ and u from the set $\mathcal{S}$. In this work, GPR is used the learn the state-optimal control mapping from a pre-generated field of extremals, the state samples X_{OL} being the inputs and control and time-to-go samples $\{U_{OL},\ T^0_{2go}\}$ being the outputs.

Let the Gaussian process representing the state-control mapping be expressed as the following additive composition of a regression polynomial $\mu(\mathbf{x})$ and a pure multivariate Gaussian process $Z(\mathbf{x}, \omega)$:

$$U(\mathbf{x}, \omega) = \mu(\mathbf{x}) + Z(\mathbf{x}, \omega) \tag{14}$$

where $\mathbf{x} \in \mathcal{D} \subset \mathbb{R}^n$ is the domain in which samples are observed, $\omega \in \Omega$ for some sample space Ω and

$$E[Z(\mathbf{x}, \cdot)] = 0,\ \ \mathrm{Cov}[Z(\mathbf{x}_i, \cdot), Z(\mathbf{x}_j, \cdot)] = C_{ij} = s^2 R_{ij},\ \ i, j = 1, \ldots, p \tag{15}$$

Here C_{ij} and R_{ij} are, respectively, the covariance and correlation between two of the observations and s^2 is the process variance. In other words, according to the model Eq. (14), the deterministic outputs $u(\mathbf{x}_i) = u_i$ (from the extremals) can be regarded as a particular *realization* $U(\mathbf{x}_i, \cdot)$ of the Gaussian random function $U(\cdot)$ with mean $\mu(\mathbf{x})$ and covariance matrix $[C_{ij}]$, cf. Eq. (13). The discrepancy between the observed data (the open-loop controls) and the mean function is the covariance $C_{ij} = C(x_i, x_j)$, typically a smooth function. In order for GPR to be effective, the actual form of the mean and covariance functions must be chosen *conditioned on the data*. This process of training the GPR model is, however, non-trivial, given that in most practical applications, no prior information is available regarding the process generating the data, the one whose metamodel is sought. A common way to deal with the issue is to use a *hierarchical prior*, whereby the mean and covariance functions are expressed in terms of the so-called *hyperparameters* that are subsequently computed given the data, for instance, through Maximum Likelihood Estimation (MLE) [17]. For example, if it is a reasonable assumption that the map to be approximated is a second order polynomial, then, for $n = 3$, the mean function can be expressed in the form:

$$\begin{aligned}\mu(\mathbf{x}) = \beta_{000} &+ \beta_{100}x_1 + \beta_{010}x_2 + \beta_{001}x_3 + \beta_{110}x_1x_2 \\ &+ \beta_{101}x_1x_3 + \beta_{011}x_2x_3 + \beta_{200}x_1^2 + \beta_{020}x_2^2 + \beta_{002}x_3^2\end{aligned} \tag{16}$$

If, in addition, a linear covariance function of the type:

$$C_{ij}(\theta, \mathbf{x}_i, \mathbf{x}_j) = s^2 \max\{0,\ 1 - \theta\,|\mathbf{x}_i - \mathbf{x}_j|\} \tag{17}$$

is assumed, then MLE estimation must determine the following hyperparameter vector:

$$\mathcal{V} = [\beta_{000}\ \beta_{100}\ \beta_{010}\ \beta_{001}\ \beta_{110}\ \beta_{101}\ \beta_{011}\ \beta_{200}\ \beta_{020}\ \beta_{002}\ \theta]^T \tag{18}$$

In general, assuming that the regression polynomial μ is expressible as a linear combination of regression basis functions , i.e $\mu(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{f}(\mathbf{x})$, it can be shown that the Minimum Variance Unbiased Estimate (MVUE) of the control at a test point $\mathbf{x}_{\text{test}}$ reduces to the following expression [12]:

$$\widetilde{u}^*(\mathbf{x}_{\text{test}}) = \widetilde{\gamma}^*(\mathbf{x}_{\text{test}}) = \hat{\boldsymbol{R}}_0(\mathbf{x}_{\text{test}})^T \boldsymbol{\Phi} + \mathbf{f}_0(\mathbf{x}_{\text{test}})^T \hat{\boldsymbol{\beta}}_{GLS} \tag{19}$$

where

i) $\hat{\boldsymbol{R}}_0$ is the training-test set covariances computed based on the ML estimated free parameters of the covariance matrix $[C_{ij}]$.

ii) $\mathbf{\Phi} \in \mathbb{R}^p$ is a matrix computed from the training control samples u_i, the regression basis function evaluations over the design matrix $\{\mathbf{x}_i\}$ and control test set covariances, cf. reference [12] for details.

iii) $\mathbf{f}_0 := [f_1(\mathbf{x}_{\text{test}})\ f_2(\mathbf{x}_{\text{test}}) \ldots f_k(\mathbf{x}_{\text{test}})]^T$, f_i being the regression basis functions.

iv) $\hat{\boldsymbol{\beta}}_{GLS}$ is the vector of the generalized least squares estimate of the regression coefficients $\boldsymbol{\beta}$, also computed based on the ML estimated free parameters of the covariance matrix.

Optimal control synthesis was accomplished through the principles outlined above, and in particular, by using Eq. (19). All GPR-related computations were carried out using DACE, a Matlab-based Kriging toolbox [18].

In *sequential* Gaussian Process Regression, control is synthesized each time a guidance call is made, or equivalently, at each numerical integration step, by constructing a *local* GPR model with extremal samples drawn from a temporal neighborhood of the current state $\mathbf{x}_{\text{curr}}$. Thus, if control at the current time t_{curr} corresponding to the current state $\mathbf{x}_{\text{curr}}$ is desired, a subset $\mathcal{W} \subseteq \mathcal{S}$ of the original training set is mined from $\mathcal{S}$ with the criteria that the times associated with the design submatrix be in a neighborhood $\mathcal{N}_\epsilon$ of t_{curr}. Figure (2) schematically illustrates the concept for a scalar system at two successive instants of control approximation. This manner of sequentially constructing local GPR models as numerical integration proceeds is the principal difference in control synthesis implementation between this work and previously reported, related work of references [12, 13]. There, a *global* GPR model is formed only once from the entire extremal training set $\mathcal{S}$ at the simulation outset, which is then queried each time a guidance call is made. To appreciate the motivation for adopting s-GPR, it may be recalled that the role of the prior covariance function Eq. (15) in GPR is to control the smoothness of the model, the influence of nearby design points on the test location, and the differentiability of the mapping, by quantifying the correlation between two observations. Typically, not enough prior information is available about the "truth" map, such as $\beta(r, u, v)$, to be able to confidently specify a functional form of C_{ij}. Typical choices for this are exponential, Gaussian, spherical and linear covariance models, all of which are smooth functions. Adopting the global GPR approach is tantamount to presupposing that the underlying multivariate function being approximated is indeed smooth over the entire domain of interest. The s-GPR method, on the other hand, only demands local smoothness around each new state sample, which is less restrictive. In addition, greater flexibility in modeling the control hypersurface is afforded by the fact that each local GPR mean function, associated with its own set of regression coefficients $\boldsymbol{\beta}$ (see Eq. (16)), will in general be different from the next GPR mean function in the sequence. Thus, each local GPR structure can be molded to follow the local surface trend more faithfully as compared to global GPR, as the latter would effectively result in fitted coefficients forced to model an "averaged out", global surface. The trade-off is, however, the lag or the control computation time. While the global GPR approach results in near-instantaneous feedback control computation given the current state, this is not so for the s-GPR method. This is not surprising because a GPR model construction at each step involves solving a non-linear optimization problem to determine the covariance hyperparameters, in addition to manipulating matrices [12]. In spite of this, simulation result in the sequel indicates that control computation time is small enough to be suitable for consideration in real-time applications with reasonable guidance call intervals. Implementation-specific details of the s-GPR algorithm adopted for the lunar lander guidance problem are presented in the Results section.

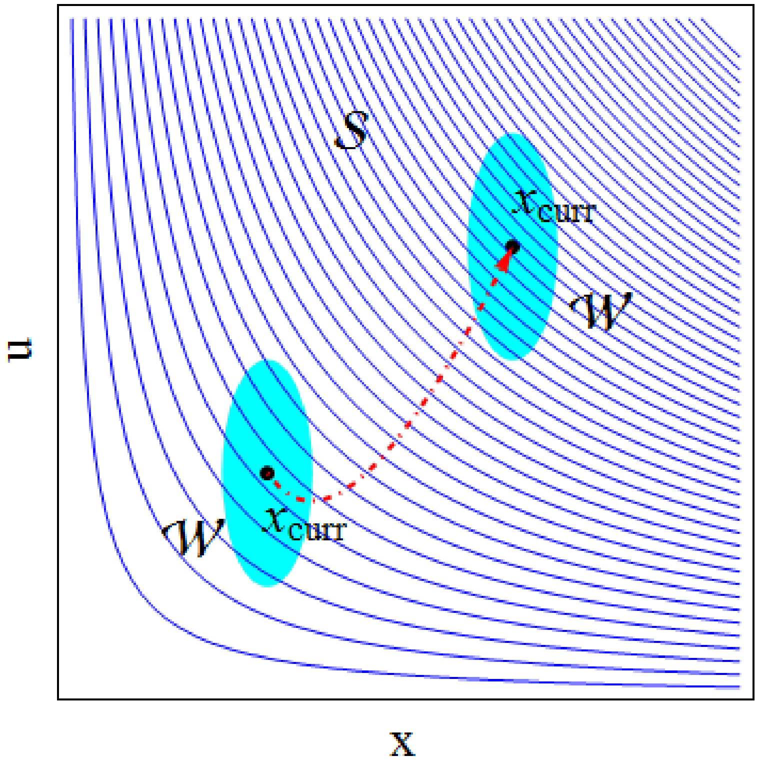

Figure 2. Sequential GPR in the Neighborhood of a State

Orbit State Estimation Error Covariance From Lunar Orbit Determination

Lunar orbit determination is accomplished in this work by applying a commercially available optimal sequential filter, in particular the EKF available in ODTK, to simulated DSN observations of the nominal lunar orbit described by Eq. (1). Figure (3) graphically illustrates the concept, capturing an instant at which the lander is visible from two of the three DSN stations. An EKF can be described as a continuously running recursive machine consisting of an alternating series of time updates and measurement updates. Time updates transition the state and estimation error covariance to the following measurement instant, while measurement updates are performed to incorporate the latest measurement information, producing an improved state estimate, with an attendant reduction in the σ's of the state error covariance matrix. The state estimation error covariance matrix is recursively updated using the following matrix difference equation as part of EKF time update:

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}(t_k, t_{k+1})\mathbf{P}_{k|k}\mathbf{\Phi}^T(t_k, t_{k+1}) + \mathbf{Q}(t_k, t_{k+1}) \tag{20}$$

where $\mathbf{P}_{k+1|k}$ is the state estimation error covariance at time t_{k+1} with measurement information through time t_k, $\mathbf{\Phi}(t_k, t_{k+1})$ is the linear state error transition function and $\mathbf{Q}(t_k, t_{k+1})$ is the additive process noise matrix that accounts for dynamical modeling uncertainty in the interval $[t_k, t_{k+1}]$. The role of $\mathbf{Q}$ quantifying dynamical modeling uncertainty in Eq. (20) is crucial in that it ensures that the covariance is realistic at the time of measurement processing. If the dynamical process noise is either too small or too large, filter divergence may occur [19]. Traditionally, the greatest challenge to the successful application of an EKF to lunar OD has been the large process noise arising out of inaccurate mapping of the lunar gravity field, especially of that on the far side of the Moon.

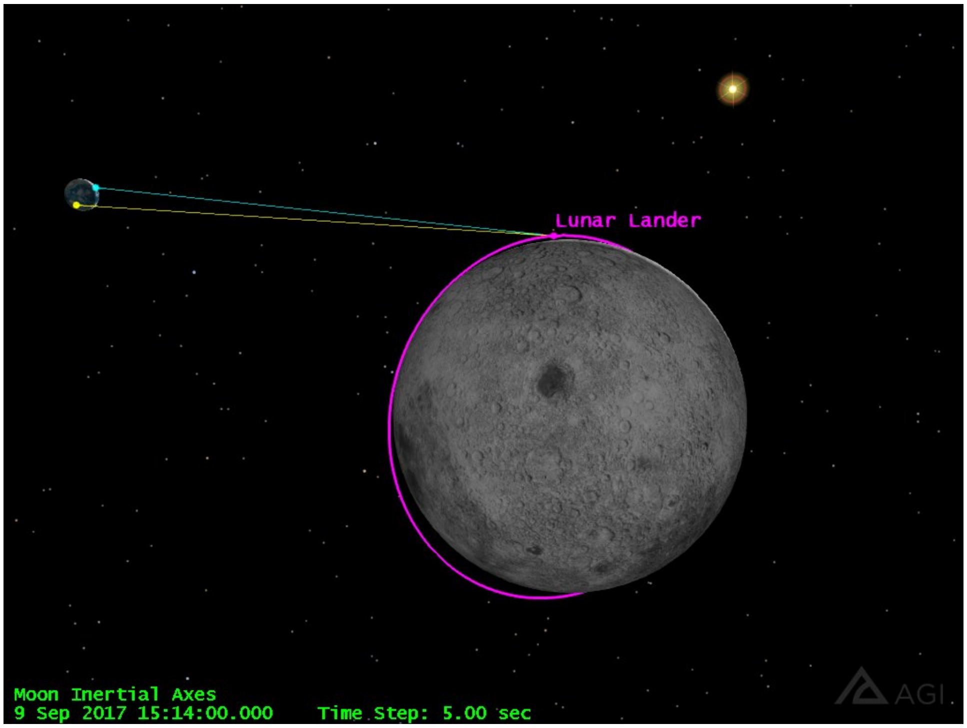

Figure 3. Lunar lander orbit visibility from DSN stations

These large uncertainties were reflective of the lack of far-side tracking on missions such as Lunar Prospector (LP). Large uncertainty in lunar gravity field has been greatly reduced with data from the recent GRAIL mission. GRAIL data have been incorporated into ODTK, but this work uses a process noise model based LP150Q gravity solution. The motive for this is to provide increased uncertainty in the estimated orbit, with an eye to "stress test"-ing the proposed guidance algorithm. For more details on lunar OD and the associated challenges, cf. [19, 20] and the references therein.

To generate a realistic position-velocity covariance time history for the lunar lander, visibility time intervals of the lander orbit from each of the DSN ground stations 27 (Goldstone, CA antenna), 34 (Canberra, Australian antenna), and 54 (Madrid, Spain antenna) were computed for a month from Aug 25 2017 00:00:00 to Sep 25 2017 00:00:00 using STK Access [21]. During the tracking interval, the Moon completed slightly more than one complete revolution about the Earth, leading the ground stations to experience all possible tracking geometries of the lander orbit, including instances when the polar lunar orbit is viewed edge-on and those when it is nearly perpendicularly viewed. The former viewing geomertry will result in relatively large cross-track position uncertaintly, whereas the latter geometry results in large in-track position uncertainty. The access intervals so generated were converted to a tracking schedule and tracking data where each ground station experienced exactly one lander pass per day. This modest observation volume is expected to result in evenly spaced coverage of the orbit while keeping cost (the number of tracking passes) at a reasonable level. Table (1) shows the OD force modeling options for the lunar lander, while Table (2) enumerates the measurement summary for each tracker by observation type. Typical numerical values were chosen for measurement statistics for each measurement type, namely, DSN total count phase and two-way sequential range and are reported in tables (3) and (4) respectively. Figure (4) shows the measurement residuals normalized by the measurement error root variance, a quantity

Table 1. Force Models for the lander

Lunar Gravity	Field: 50×50 Solid Tides: On, $k_2 = 0.03$
Third Body Gravity	Sun, Earth
Solar Pressure	Satellite Model: Spherical, $C_r = 4.61538$, A = 1.3 m^2 Shadow Model: Dual Cone

Table 2. Measurement Types By Tracker

Tracker Name	Measurement Type	Number of Measurements
DSS27	DSN Sequential Range DSN Total Count Phase	930 930
DSS34	DSN Sequential Range DSN Total Count Phase	930 930
DSS54	DSN Sequential Range DSN Total Count Phase	930 930

called residual ratios. This is a unitless metric typically used to indicate valid measurements when its numerical value is with ± 3. It is clear that data points primarily lie within the desirable range, which is theoretically predicted if the samples follow a Gaussian distribution. Position and velocity uncertainties from the filter at the 0.99 probability level are shown in Figs. (5) and (6), respectively.

GENERATING OPEN-LOOP EXTREMALS FROM OD COVARIANCE

Having associated each state on the lander parking orbit with an error covariance, the next step is to select a de-orbit point, the covariance extent of which affords a sampling region for generating a field of extremals, which in turn will act as a training set for the s-GPR controller model. However, before choosing a de-orbit point, it is instructive to examine the relation between Clohessy-Wiltshire-Hill frame coordinates and classical orbit element differences. Sampling a 6×6 position-velocity covariance ellipsoid around a mean state $\{\mathbf{r}_\mu, \mathbf{v}_\mu\}$ will generate states $\{\mathbf{r}_\mu + \delta\mathbf{r}_i, \mathbf{v}_\mu + \delta\mathbf{v}_i, i = 1 \ldots n_{\text{samples}}\}$ where $\{\mathbf{r}_\mu, \mathbf{v}_\mu \in \mathbb{R}^3\}$. With:

$$\delta\mathbf{r}_i := \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \quad \delta\mathbf{v}_i := \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} \tag{21}$$

Table 3. Station Total Count Phase Measurement Statistics

Station	Bias Model	Constant	Sigma	Half Life (min)	White Noise Sigma	Count Interval (sec)
DSS27 DSS34 DSS54	Gauss Markov	0	0.02	300	0.003	1

Table 4. Station Sequential Range Measurement Statistics

Station	Bias Model	Constant	Sigma (m)	Half Life (min)	White Noise Sigma (m)	Tropospheric Measurement Corrections Sigma
DSS27	Gauss Markov	0	0.5	300	1	0.05
DSS34	Gauss Markov	0	1.5	300	1	0.05
DSS54	Gauss Markov	0	1.5	300	1	0.05

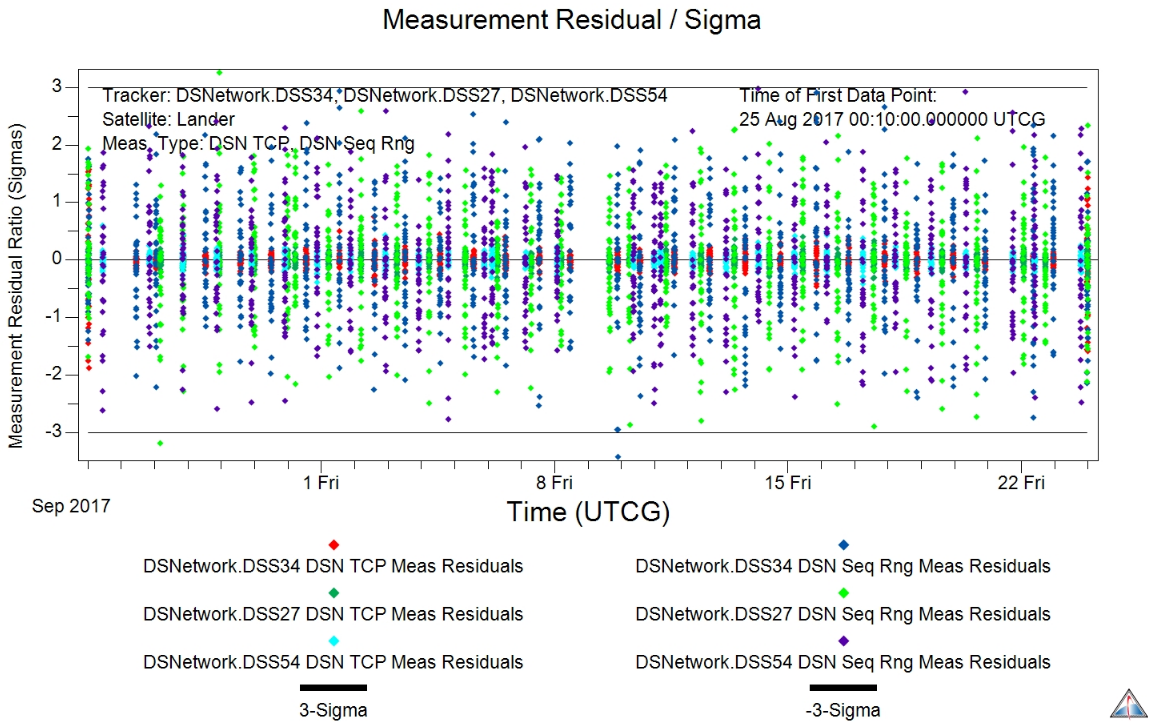

Figure 4. Total Count Phase and Sequential Range Normalized Residuals

where x, y, z and $\dot{x}, \dot{y}, \dot{z}$ denote the radial (R), in-track (I) and cross-track (C) components of the position and velocity, it can be shown that the following relation holds between $[\delta\mathbf{r}\ \delta\mathbf{v}]^T$ and $\delta\mathbf{e}^T := [\delta a\ \delta\theta\ \delta i\ \delta q_1\ \delta q_2\ \delta\Omega]^T$ [22]:

$$x = \delta r \tag{22}$$

$$y = r(\delta\theta + \cos i\ \delta\Omega) \tag{23}$$

$$z = r(\sin\theta\ \delta i - \cos\theta\ \sin i\ \delta\Omega) \tag{24}$$

$$\dot{x} = -\frac{u_0}{2a}\delta a + (\frac{1}{r} - \frac{1}{p})h\delta\theta + (u_0 a q_1 + h\sin\theta)\frac{\delta q_1}{p} + (u_0 a q_2 - h\cos\theta)\frac{\delta q_2}{p} \tag{25}$$

$$\dot{y} = -\frac{3v_0}{2a}\delta a - u_0\delta\theta + (3v_0 a q_1 + 2h\cos\theta)\frac{\delta q_1}{p} + (3v_0 a q_2 + 2h\sin\theta)\frac{\delta q_2}{p} + u_0\cos i\delta\Omega \tag{26}$$

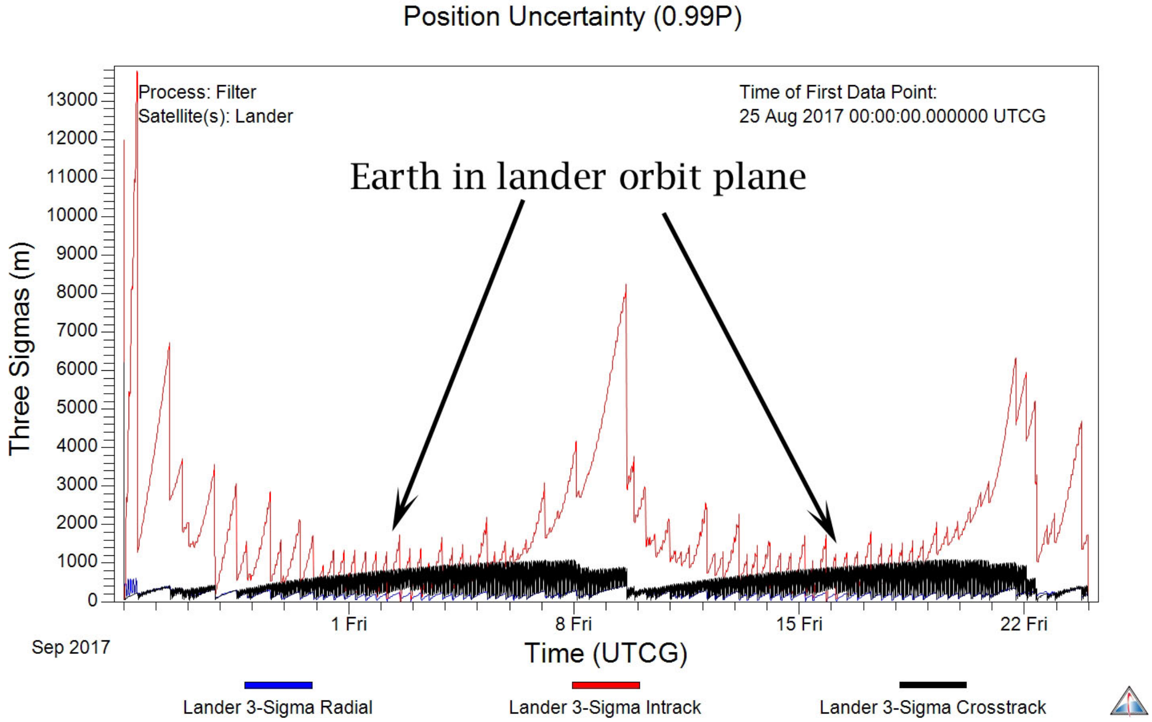

Figure 5. 3-σ position uncertainty from EKF

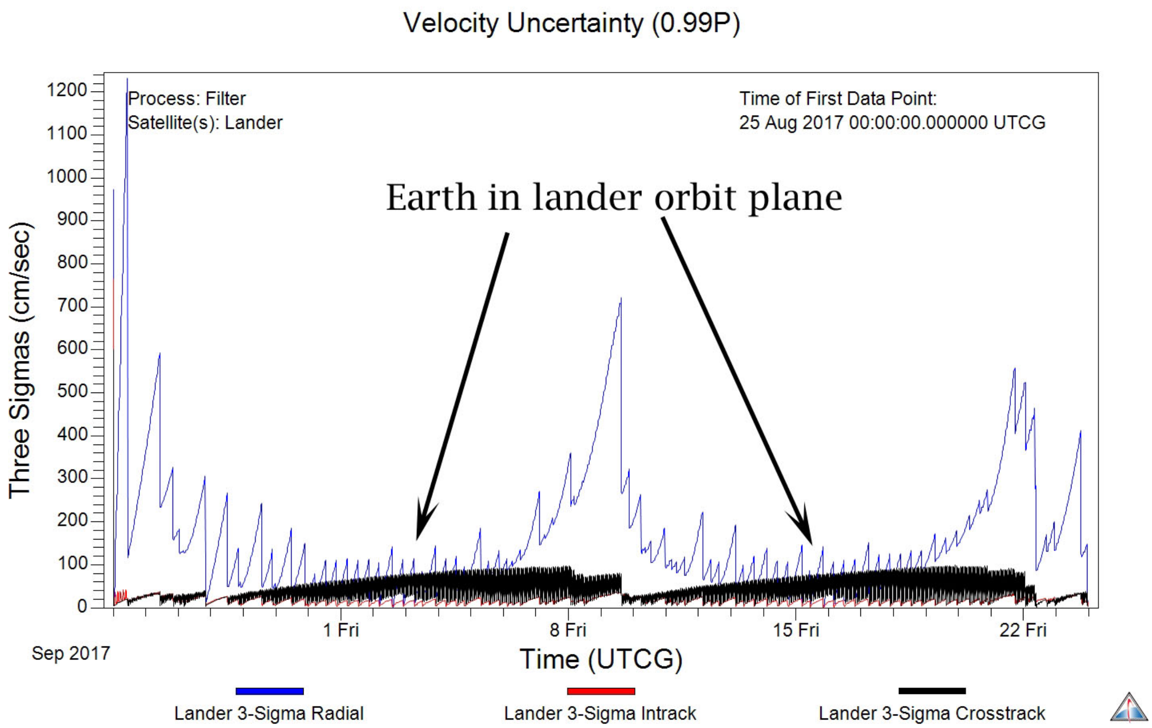

Figure 6. 3-σ velocity uncertainty from EKF

$$\dot{z} = (v_0 \cos\theta + u_0 \sin\theta)\delta i + (v_0 \sin\theta - u_0 \cos\theta)\sin i\delta\Omega \tag{27}$$

In the above relations, θ is the true latitude angle, $q_1 = e\cos\omega,\ q_2 = e\sin\omega$, h is the orbit angular momentum of the mean Keplerian orbit, and u_0 and v_0 are the radial and tangential velocity components of the lander nominal state. From Eqs. (23), (24), and (27), it is clear that any perturbation in in-track position, cross-track position and cross-track velocity is caused purely by the lander starting in an orbit of different inclination and orientation. Since the control objective in this paper is not pin-point landing, such perturbation is neither relevant nor modeled by the lander dynamics. Once the lander has started in an orbit of slightly different orientation, the guidance objective would still be to bring the lander to an altitude of 10 km. above the surface with negligible speed *in the starting orbit's plane*. Thus, in conclusion, only radial velocity, radial position and in-track velocity perturbations are sampled from the 6×6 state estimation error covariance matrix to generate $\{X_{OL}, U_{OL}\}$.

The de-orbit initiation epoch is selected as $t_{do} =$ 09 Sep 2017 15:14:00.0 UTCG, an instant at which radial velocity (and in-track position) uncertainty are large owing to nearly perpendicular viewing of the lander orbit by DSS27 and DSS34. Figure (7) visually depicts the 3σ position uncertainty of the lander in the vehicle-centered RIC frame at t_{do}. Specifically, the region enclosed by the position estimation error ellipsoid shown in the figure can be represented by the following inequality:

$$[x\ \ y\ \ z]P_{\mathbf{r}}^{-1}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq 3^2 \tag{28}$$

where $P_{\mathbf{r}}$ represents the 3×3 position submatrix of the position-velocity covariance matrix, cf. Eq. (20). This situation corresponds to the uncertainty peak approximately in the middle of the analysis interval in Figs. (5) and (6). From ODTK computations, the nominal (mean) radial position, radial and tangential/in-track velocity components at the de-orbit epoch are easily obtained to be:

$$r_0(t_{do}) = 1.787 \times 10^3 \text{ km.},\ \ u_0(t_{do}) = -30.2037 \text{ m/s},\ \ v_0(t_{do}) = 1.6783 \text{ km/s} \tag{29}$$

At the same epoch, the following 3σ uncertainties are extracted from the position-velocity covariance matrix associated with the above state:

$$\delta r_{3\sigma}(t_{do}) = 387.4192 \text{ m.},\ \ \delta u_{3\sigma}(t_{do}) = 841.9125 \text{ cm/s},\ \ \delta v_{3\sigma}(t_{do}) = 40.2120 \text{ cm/s} \tag{30}$$

In order to synthesize a near-optimal feedback controller based on s-GPR, a set of 25 initial conditions are randomly sampled from within a hypervolume represented by the following Cartesian product:

$$\begin{aligned} V \triangleq\ & [r(t_{do}) - \delta r_{6\sigma}(t_{do})\quad r(t_{do}) + \delta r_{6\sigma}(t_{do})] \\ & \times [u(t_{do}) - \delta u_{6\sigma}(t_{do})\ \ u(t_{do}) + \delta u_{6\sigma}(t_{do})] \times [v(t_{do}) - \delta v_{6\sigma}(t_{do})\ \ v(t_{do}) + \delta v_{6\sigma}(t_{do})]. \end{aligned} \tag{31}$$

Note that 6σ rather than 3σ uncertainty is used to generate controller training data to account for greater uncertainty during feedback synthesis, allowing for a more pessimistic controller design. Random sampling is accomplished with *Latin Hypercube Sampling* (LHS), a stratified sampling method commonly used to generate space-filling designs for computer experiments [23]. Figure (8) shows a family of solutions to the trajectory optimization problem posed by Eqs. (2 - 11) corresponding to the 25 LHS-drawn initial conditions from OD-generated uncertainty. All solutions are obtained using GPOPS-II.

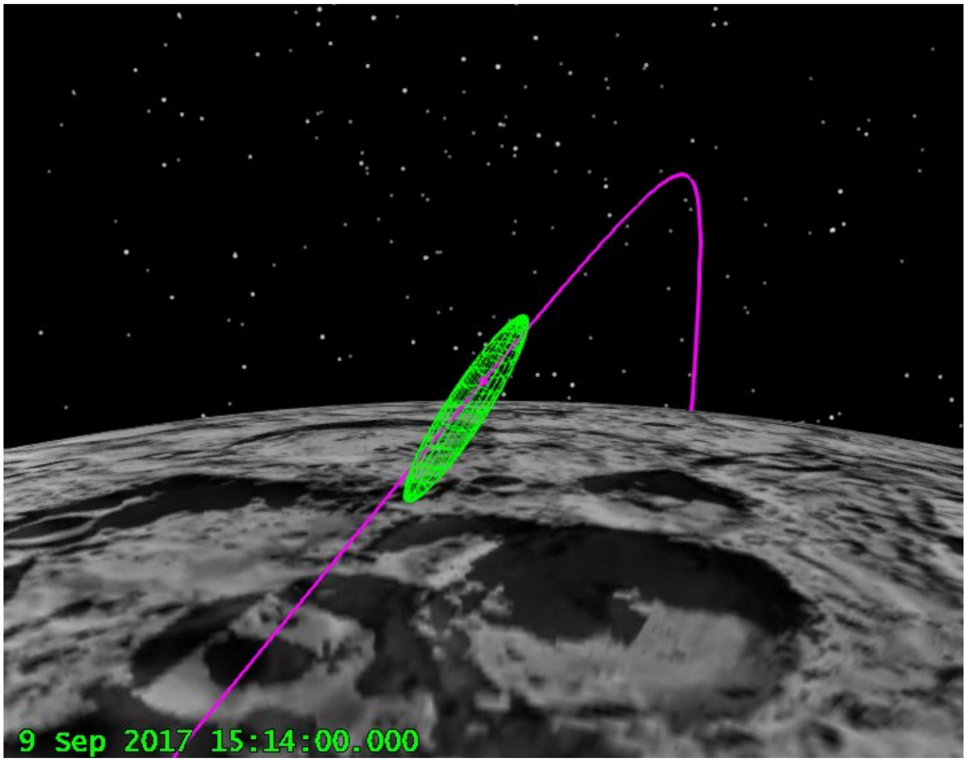

Figure 7. 3-σ error ellipsoid of the lander at the de-orbit instance

RESULTS

The aim of this section is to examine the effectiveness of the s-GPR guidance algorithm. To this end, the intention is to verify that the feedback control $\beta(r, u, v)$ computed at each guidance call and simulated as a numerical integration step, is able to guide the system autonomously from a random initial condition (that was not a member of the controller synthesis training set but still sampled from the OD-generated uncertainty), to the desired terminal manifold with a reasonable degree of accuracy while closely meeting the optimality criteria. Clearly, in the light of the earlier discussion on Optimal Feedback Synthesis, this verification process should involve noting the agreement between the feedback state-control pair and the corresponding open-loop quantities; a close agreement between the open and closed-loop solutions is indicative of the "goodness" of the guidance algorithm. The following steps are taken to verify that the s-GPR guidance strategy is suitable for the problem at hand:

1. Randomly generate a sample of n_{test} initial condition vectors $\mathbf{x}_{\text{test}_i} = [r_{0i}\ u_{0i}\ v_{0i}]^T$ within V (cf. Eq. (31)). Call this test set $\mathcal{T} \triangleq \{\mathbf{x}_{\text{test}_i}\}_{i=1}^{n_{\text{test}}}$. LHS is once again used for this purpose to ensure that the entire uncertainty extent is faithfully represented in the test samples. In this paper, $n_{\text{test}} = 10$.

2. For any given member $\mathbf{x}_{\text{test}_i} \in \mathcal{T}$, predict, using a GPR implementation (the DACE Kriging Toolbox in the present case as mentioned earlier), the time-to-go $t_{2go_i}(r_{0i}\ u_{0i}\ v_{0i})$ for that initial state. This is the simulation interval or the approximate optimal final time for $\mathbf{x}_{\text{test}_i}$. The GPR model construction data for this step consists of the input-output pairs $\{\mathbf{x}_{0_j},\ t_{2go_j}\}$,

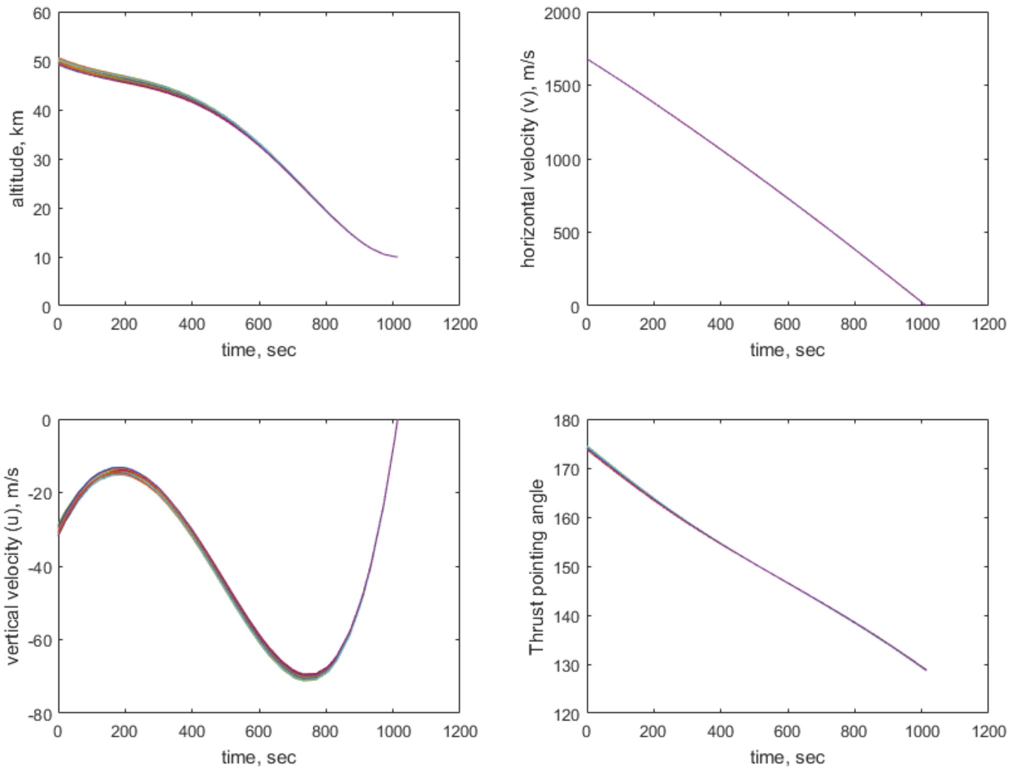

Figure 8. Optimal Feedback Synthesis Extremals

$j = 1, \ldots, 25$ from the original field of extremals set $\{X_{OL}, T^0_{2go}\}$. By trial and error, a first-order polynomial model and a exponential correlation model are found to result in the smallest prediction error in t_{2go_i} during the leave-one-out cross-validation stage using the pre-generated field of extremals.

3. To perform numerical integration with the feedback controller in the loop, at a generic current step t_{curr}, a GPR model is first constructed with input-output sets $\{x_{OL},\, u_{OL}\} \subseteq \{X_{OL},\, U_{OL}\}$ where $\{x_{OL},\, u_{OL}\}$ are open-loop states and controls at time instants $t_{OL} \in \mathcal{N}_\epsilon(t_{\text{curr}})$, a neighborhood of t_{curr}. The method of selecting $\mathcal{N}_\epsilon$ is, by no means unique and may be problem dependent. In this work, $\mathcal{N}_\epsilon \coloneqq \{t \in T_{OL} : t_{\text{curr}} - \Delta t \leq t \leq t_{\text{curr}} + \Delta t\}$, where Δt is 3 times the maximum time step in the open-loop family of extremals. Having generated a GPR model at a given integration step, the control $\widetilde{\gamma}^* = \beta(\mathbf{x}_{\text{curr}})$ can be queried from it by supplying the current state $\mathbf{x}_{\text{curr}}$. This control is then used in the lander dynamics, and the dynamical equations are propagated by *sequentially* computing a feedback control approximation at each integration step. Again, following trial and error and leave-one-out cross validation, a first-order polynomial regression model with a linear correlation function kernel, both available in DACE, yields the best agreement between open and closed-loop solutions.

Table 5 reports the mean terminal constraint violations, while Figs. (9) and (10) compare open-loop and the s-GPR-based feedback solutions for two representative members of the 10 test cases.

Table 5. Mean Terminal Constraint Violation Over Trial Cases

State	Mean Constraint Violation
Radial distance (r)	6.73 m
Radial velocity (u)	0.092 m/s
Tangential velocity (v)	0.0861 m/s

The agreement between the state and control functions for both cases are seen to be very good, so much so that differences are not perceptible at the scale of the graph. Comparing Eq. (30) and table 5 it can be seen that the mean terminal constraint violation in each state is several orders of magnitude smaller than the corresponding initial uncertainty. Another aspect relevant to the performance of guidance algorithms is the computation time. It may be recalled that generating a control at a single integration step requires querying the original field of extremals and selecting a subset of it before constructing a GPR model *at that step*. While an exact estimate of the time required for this process is non-trivial to obtain for a program executing on a high-level-language program environment such as Matlab, the average control computation time as each state "measurement" arrived was found to be 0.1511 seconds counting all the integration steps over 10 test cases. The simulation platform was Matlab 2016a on a Windows 10 Core(TM) i7-4900 Desktop clocked at 3.60 GHz with 32GB RAM [24]. Counting the overhead involved in the simulation, the actual control computation time should be smaller, although even this value leaves sufficient buffer for hardware-related overhead for a 1-second guidance call interval in a real-time application.

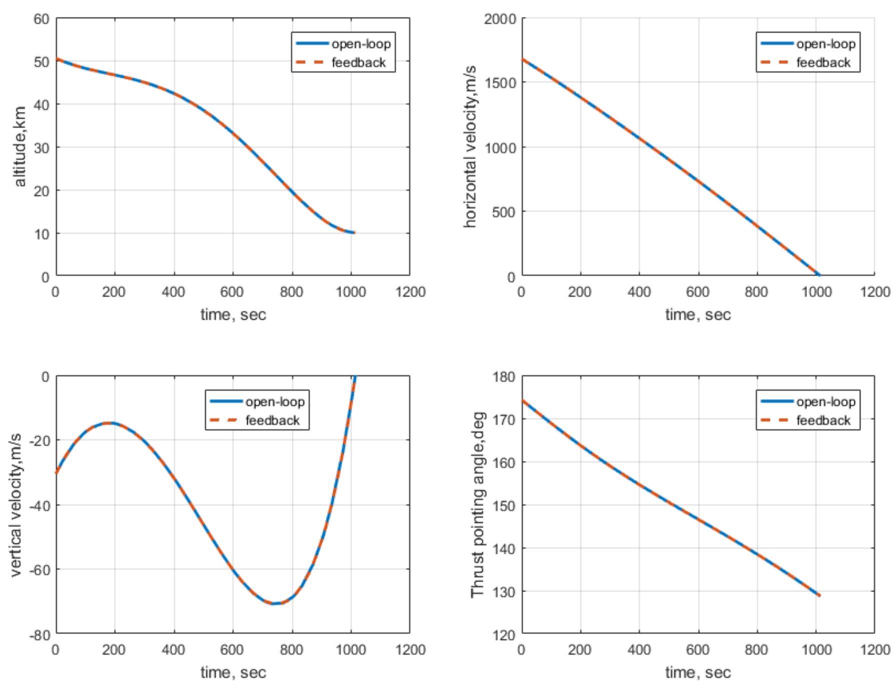

Figure 9. Open-loop and s-GPR feedback trajectories compared, case 1

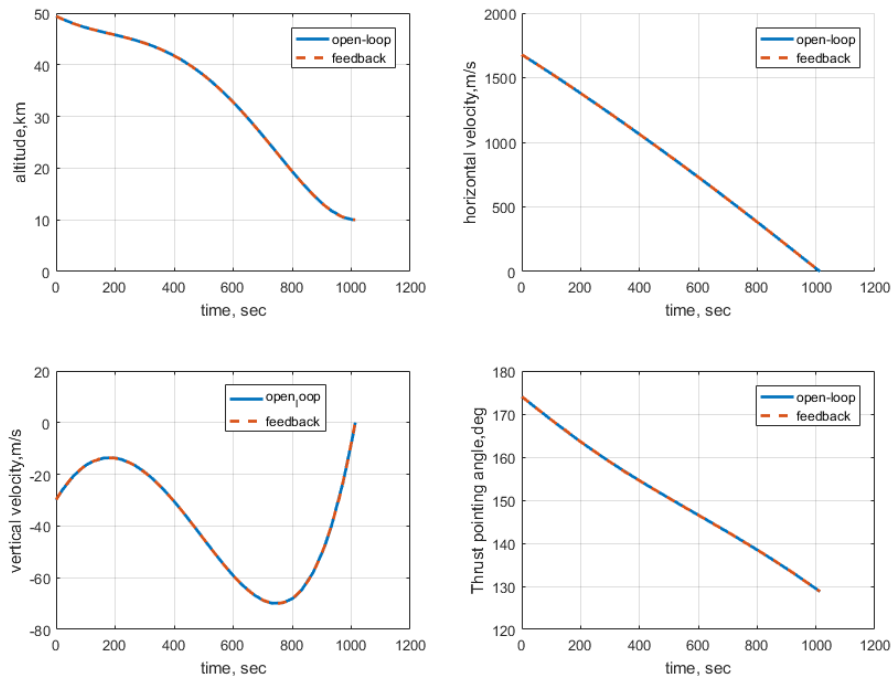

Figure 10. Open-loop and s-GPR feedback trajectories compared, case 2

CONCLUSION

A near-optimal, explicit, guidance law based on Gaussian Process prediction was applied to an instance of the lunar soft landing problem. At each new state sample, an optimal feedback control for the current state was synthesized by creating a GPR model with state-control information collected only from a neighborhood of that state. It was pointed out that this mode of local, sequential, control approximation along the trajectory does not necessarily require the assumption that the state-control mapping being learned by the GPR be smooth over the entire domain of approximation. The effectiveness of the proposed optimal feedback algorithm was evaluated by verifying that it could satisfactorily compensate initial state uncertainties of the lander, originally in a low lunar parking orbit. A key feature of this work is the mathematical characterization and numerical computation of the lander initial state uncertainty. A realistic measure of the orbit state uncertainty in the form of a full position-velocity covariance was computed from lunar orbit determination. Simulated observations from the NASA/JPL Deep Space Network was used for this purpose.

REFERENCES

[1] *How SpaceX's 2018 Moon Flight Will Work*, March 1, 2017 (last accessed August 2, 2017). `https://www.space.com/35876-how-spacex-moon-flight-will-work.html`.

[2] *Chang'e 5*, (last accessed August 2, 2017). `https://en.wikipedia.org/wiki/Chang%27e_5`.

[3] J. N. Goswami and M. Annadurai, "Chandrayaan-2 Mission," 42^{nd} *Lunar and Planetary Science Conference*, The Woodlands, Texas, USA, 2011.

[4] Sakai S., Sawai *et al.*, "Small Lunar-lander "SLIM" for the Pinpoint Landing Technology Demonstration," 11^{th} *Low Cost Planetary Missions Conference*, Berlin, Germany, 2015.

[5] A. R. Klumpp, "Apollo Lunar Descent Guidance," *Automatica*, Vol. 10, 1974, pp. 133–146.

[6] C. R. McInnes, "Gravity-Turn Descent from Low Circular Orbit Conditions," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, 2003, pp. 183–185.

[7] R. V. Ramanan and M. Lal, "Analysis of Optimal Strategies for Soft Landing on the Moon from Lunar Parking Orbits," *Journal of Earth System Science*, Vol. 114, No. 6, 2005, pp. 807–813.

[8] C. T. Chomel and R. H. Bishop, "Analytical Lunar Descent Guidance Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 915–926.

[9] A. Y. Lee, "Fuel-efficient Descent and Landing Guidance Logic for a Safe Lunar Touchdown," *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, 2011.

[10] J. Zhou, K. L. Teo, D. Zhou, and G. Zhao, "Nonlinear Optimal Feedback Control for Lunar Module Soft Landing," *Journal of Global Optimization*, Vol. 52, No. 2, 2012, pp. 211–227.

[11] G. Cecchetti, M. Pontani, and P. Teofilatto, "Neighboring Optimal Guidance for Soft Lunar Landing," *Second IAA Conference on Dynamics and Control of Space Systems 2014*, Rome, Italy, 2014.

[12] P. Ghosh and B. A. Conway, “Near-Optimal Feedback Strategies Synthesized Using a Spatial Statistical Approach,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 905–919.

[13] P. Ghosh and B. A. Conway, “Spatial Statistical Point Prediction Guidance for Heating-rate-limited Aeroassisted Orbital Transfer,” *Acta Astronautica*, Vol. 111, 2015, pp. 257–269.

[14] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. Taylor and Francis, 1975.

[15] M. A. Patterson and A. V. Rao, “GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming,” *ACM Trans. Math. Softw.*, Vol. 41, Oct. 2014, pp. 1:1–1:37.

[16] Analytical Graphics Inc., *Orbit Determination Tool Kit version 6.4.5*, 2016.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: The MIT Press, 2006.

[18] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard, “DACE: A Matlab Kriging Toolbox version 2.0,” tech. rep., IMM Technical University Of Denmark, 2002.

[19] S. Slojkowski, J. Lowe, and J. Woodburn, “Orbit Determination for the Lunar Reconnaissance Orbiter Using an Extended Kalman Filter,” 25^{th} *International Symposium on Space Flight Dynamics*, Munich, Germany, October 2015.

[20] J. Woodburn and J. H. Seago, “Orbit Determination of Satellites in Lunar Orbit Using an Optimal Sequential Filter,” *AAS/AIAA Space Flight Mechanics Meeting*, Galveston, Texas, January 2008.

[21] Analytical Graphics Inc., *Systems Tool Kit version 11.2*, 2017.

[22] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems, Second Edition* . American Institute of Aeronautics and Astronautics, 2009. pp. 606-611.

[23] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. Springer, 2003.

[24] The MathWorks Inc., *MATLAB version 9.1.0 (R2016b)*, 2016.