DIFFERENTIAL DYNAMIC PROGRAMMING APPROACH FOR ROBUST-OPTIMAL LOW-THRUST TRAJECTORY DESIGN CONSIDERING UNCERTAINTY

Naoya Ozaki(1), Stefano Campagnola(2), Chit Hong Yam(2), and Ryu Funase(1)
(1)*The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan, +81-3-5841-6631*
ozaki@space.t.u-tokyo.ac.jp
(2)*Japan Aerospace Exploration Agency, Institute of Space and Astronautical Science, 3-1-1, Yoshinodai, Chuo-ku, Sagamihara, Kanagawa, 252-5210, Japan, +81-50-336-26364*

Abstract: *This paper proposes a robust-optimal trajectory design method for uncertain system to minimize the expected value of objective function. The basic idea is solving Stochastic Differential Dynamic Programming (SDDP), which solve optimal control problem to minimize the expected value of cost-to-go function, with Unscented Transform, which is used to estimate the expected value. Most recent studies have focused on the trajectory optimization assuming that the spacecraft can control their trajectory exactly as planed; however, the assumption is violated in the realistic operations where uncertain events, such as missed-thrust and navigation error, perturb the predetermined trajectory. Conventionally, experienced specialists empirically determine "margin" on optimal low-thrust trajectory by duty cycle or forced coast period. A proposed SDDP autonomously provides "margin" in optimization for future feedback as well. Numerical result by V-infinity leveraging problem shows that SDDP has better performance, in other words, expected value of objective function considering uncertainty is better, than conventional DDP when the system has uncertainty, since the result of SDDP has "margin" in optimal control for future feedback.*

Keywords: *Differential Dynamic Programming, Low-Thrust Trajectory Optimization, Stochastic Programming, Robust-Optimal Control*

1. Introduction

1.1. Background

In recent years, low-thrust propulsion systems have been increasingly used in the interplanetary missions, because these systems have high specific impulse and can achieve large delta-V to travel far from the Earth. Due to this background, various low-thrust trajectory design methods have been developed [1][2]. These methods assume that the spacecraft is perfectly guided to the predesigned trajectory as planed; however, this assumption is violated in realistic operation because of various disturbances including missed-thrust [3] (i.e. contingent coasting period due to temporary operational troubles) and navigation error.

When the system has uncertainty, the state (e.g. position and velocity) will not evolve as expected. Hence, in the trajectory optimization under uncertainty, it is significant to ensure the feasibility throughout every time step. To assure the feasibility throughout trajectory, we usually retain a "margin" in optimization for future feedback (this approach is called "constraint tightening" method [4][5]). Currently, this margin is empirically and intuitively implemented by, for example, duty cycle and forced-coast period [3]. The major motivation in this study is to suggest numerical method to design robust and optimal low-thrust trajectory with an appropriate margin.

Robust-optimal low-thrust trajectory design with an appropriate margin will increase in importance for following reasons:

1) *Robustness to missed-thrust*

   Most of the interplanetary mission using low-thrust propulsion system, such as Dawn (NASA) and Hayabusa (JAXA) spacecraft, experiences missed-thrust caused by operational anomaly such as malfunction of propulsion system or safe-mode. Missed-thrust is inevitable because the low-thrust propulsion system require long-time operation to achieve a certain delta-V.

2) *Reasonable mission by small spacecraft*

   In conventional mission, we make effort to reduce uncertainty as much as possible by expensive way (precious orbit determination by DDOR, highly reliable propulsion system, and so on). On the other hands, in the reasonable mission by small spacecraft, instead of using expensive way, we should permit certain amount of uncertainty.

3) *Advanced mission by solar sail spacecraft*

   Technology of solar sail spacecraft has been demonstrated by IKAROS (JAXA) and LightSail (Planetary Society). In the next stage, the practical mission by solar sail is proposed, such as NEA Scout (NASA). Solar sail has much uncertainty in thrust (because of uncertainty in optical parameter and area-mass ratio).

1.2. Related Works

Most of the previous works to design robust low-thrust trajectory against uncertainty is based on deterministic method with empirical margin [3]. The method to evaluate the margin against missed-thrust is also proposed [6]. However, except certain special cases, the optimal control for the system with uncertainty is different from that without uncertainty. Hence, the conventional approach is no longer optimal control for realistic system.

As for the optimal control for the system with uncertainty, Olympio and Yam[7] have suggested the surrogate-based method to solve the robust-optimal trajectory for one temporary engine failure. Olympio[8] has also suggested the sophisticated method using two-stage stochastic programming with indirect method; however this method is only focused on one temporary engine failure (because of two-stage). Therefore, we need to invent an innovative multi-stage robust-optimal trajectory design method for a general system with uncertainty.

Multi-stage robust optimization problem can be achieved by Robust Dynamic Programming (RDP)[9][10], which has recently received substantial attentions. Generally, RDP cannot be solved numerically without special assumption. Most methods to solve RDP are implemented with the assumption of linear system; however trajectory design problem is usually nonlinear problem.

As for Dynamic Programming to solve trajectory design problem, Differential Dynamic Programming (DDP) [11] has recently received substantial attentions again. To apply dynamic programming to continuous state system numerically, we notice that it has inherent difficulty called “curse of dimensionality” since the dimension of the state variables becomes incredibly large. To overcome this fundamental difficulty, DDP was created based on expanding the Principle of Optimality by second order around the reference trajectory. The classical DDP was only effective for smooth unconstrained problems; on the other hands low-thrust problems fundamentally have constrained bang-bang structure (i,e, DDP may converge slowly or may not converge at all for the low-thrust problems). Recent works for DDP[12][13] have been improving the applicability to the low-thrust problem by incorporating well-developed Nonlinear Programming (NLP) techniques to DDP[13].

The advantages of DDP-based low-thrust trajectory optimization methods are:

- High robustness to poor initial guess.
- Good applicability to large-scale problems, such as multi-revolution problem, because the computational effort per iteration of DDP increases only linearly with the number of nodes, whereas that of the common NLP-based method increases exponentially.
- Optimal feedback control policy can be also retrieved.

Robust optimal control by DDP has recently proposed in the field of reinforcement learning and optimal control by Todorov and Tassa[14] and Theodorou, Tassa and Todorov[15]. Both approaches are based on Stochastic Differential Dynamic Programming (SDDP) to minimize the expected value of objective function. One of the approaches, called iterative Local Dynamic Programming (iLDP), proposed by Todorov and Tassa uses collocation method with unscented transform; on the other hands, the other approach by Theodorou, Tassa and Todorov use the theoretical differential of stochastic system. The approach we propose in this paper follows their works.

1.3. Objectives

This study proposes a strategy to design robust optimal low-thrust trajectory for uncertain system to minimize the expected value of objective function by Differential Dynamic Programming.

2. Differential Dynamic Programming

2.1. Overview of Differential Dynamic Programming

Differential Dynamic Programming (DDP) is proposed by Jacobson and Mayne in 1966. Although it is introduced early, DDP had not been applied to trajectory design problem since it is difficult to apply DDP to constrained problem (i.e. bang-bang problem) until recently. Because there are good advantages in DDP, techniques to apply DDP to trajectory design problem have been recently well developed [12][13].

The main idea of DDP is using quadratic expansion of Bellman’s principle of optimality to overcome the curse of dimensionality in Dynamic Programming. There is iteration to calculate nonlinear optimal control problem. Every iteration of DDP involves backward sweep and forward sweep alternately shown in Fig.1. In backward sweep step, we calculate optimal control of quadratic system expanded around reference trajectory, and in forward sweep step we update reference trajectory using optimal control strategy obtained by backward sweep step.

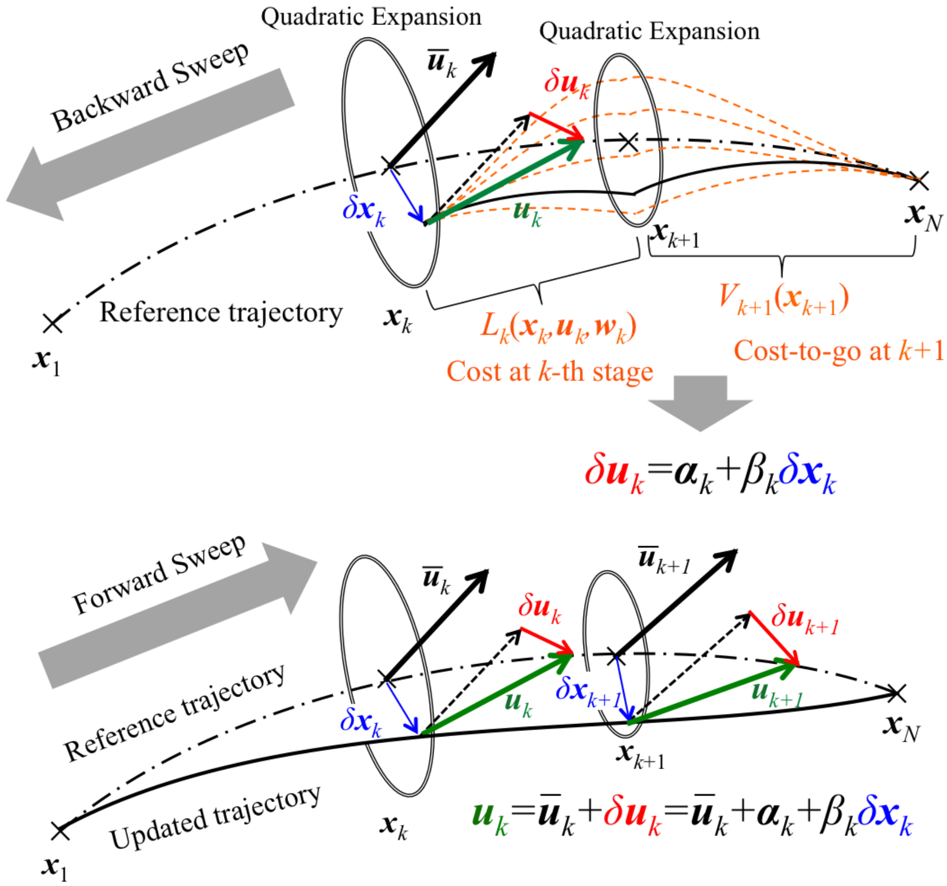

Figure 1. Calculation Step in DDP

2.2. Discrete-Time Dynamic Programming and Bellman's Principle of Optimality

Let us consider the discrete-time dynamical system described as following equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k(\boldsymbol{x}_k, \boldsymbol{u}_k;\ t_k) \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is state vector at discretized time $k \in \{1,2,\dots,N+1\}$, $\boldsymbol{u}_k \in \mathbb{R}^m$ is control vector, $t_k \in \mathbb{R}$ is time, and $\boldsymbol{F}_k(\cdot): \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$ is the nonlinear function to describe dynamical system. Given boundary conditions $\boldsymbol{x}_1 \in \mathbb{R}^n$ and $\boldsymbol{x}_{N+1} \in \mathbb{R}^n$, the general optimal control problem is to find the optimal control policy:

$$\{\boldsymbol{u}_k^*\} \coloneqq \{\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \dots, \boldsymbol{u}_N^*\} \tag{2}$$

to minimize the objective function:

$$J(\{\boldsymbol{u}_k^*\}) := \Phi_{N+1}(\boldsymbol{x}_{N+1};\ t_{N+1}) + \textstyle\sum_{k=1}^{N} L_k(\boldsymbol{x}_k, \boldsymbol{u}_k;\ t_k). \tag{3}$$

where, $\Phi_{N+1}(\cdot): \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is terminal cost function, and $L_k(\cdot): \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is cost function at discretized time k.

Instead of solving above problem directly, in Dynamic Programming, we obtain the control $\boldsymbol{u}_k$ at discretized time k by optimizing cost-to-go function $V_k(\cdot)\colon \mathbb{R}^n \to \mathbb{R}$ defined as follows:

$$V_k(\boldsymbol{x}_k) \coloneqq \Phi_{N+1}(\boldsymbol{x}_{N+1};\ t_{N+1}) + \textstyle\sum_{i=k}^{N} L_i(\boldsymbol{x}_i, \boldsymbol{u}_i;\ t_i). \tag{4}$$

According to Bellman's principle of optimality, we obtain a recursive optimization problem which optimize the cost at discretized time k and the rest of cost-to-go function:

$$V_k^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_k}[L_k(\boldsymbol{x}_k, \boldsymbol{u}_k;\ t_k) + V_{k+1}^*(\boldsymbol{x}_{k+1})]. \tag{5}$$

where $V_k^*(\cdot)\colon \mathbb{R}^n \to \mathbb{R}$ is optimal cost-to-go function.

2.3. Differential Dynamic Programming

To overcome the inherent curse of dimensionality of dynamic programming, DDP is fundamentally formulated by introducing quadratic expansions of Bellman's principle of optimality in the neighborhood of a reference trajectory. Define $\{\bar{\boldsymbol{x}}_k\}$ as reference trajectory, let us derive the quadratic expansion of Eq. (5) with respect to $\boldsymbol{x}_k (= \bar{\boldsymbol{x}}_k + \delta\boldsymbol{x}_k)$ and $\boldsymbol{u}_k (= \bar{\boldsymbol{u}}_k + \delta\boldsymbol{u}_k)$:

$$V_k^*(\boldsymbol{x}_k) \simeq\ V_k^*(\bar{\boldsymbol{x}}_k) + V_{\boldsymbol{x}}^{*(k)}\delta\boldsymbol{x}_k + \frac{1}{2}\delta\boldsymbol{x}_k^T V_{\boldsymbol{xx}}^{*(k)}\delta\boldsymbol{x}_k \tag{6}$$

$$L_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \simeq\ L_k(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k) + L_{\boldsymbol{x}}^{(k)}\delta\boldsymbol{x}_k + L_{\boldsymbol{u}}^{(k)}\delta\boldsymbol{u}_k + \frac{1}{2}\delta\boldsymbol{x}_k^T L_{\boldsymbol{xx}}^{(k)}\delta\boldsymbol{x}_k + \delta\boldsymbol{x}_k^T L_{\boldsymbol{xu}}^{(k)}\delta\boldsymbol{u}_k + \frac{1}{2}\delta\boldsymbol{u}_k^T L_{\boldsymbol{uu}}^{(k)}\delta\boldsymbol{u}_k \tag{7}$$

$$V_{k+1}^*(\boldsymbol{x}_{k+1}) \simeq\ V_{k+1}^*(\bar{\boldsymbol{x}}_{k+1}) + V_{\boldsymbol{x}}^{*(k+1)}\delta\boldsymbol{x}_{k+1} + \frac{1}{2}\delta\boldsymbol{x}_{k+1}^T V_{\boldsymbol{xx}}^{*(k+1)}\delta\boldsymbol{x}_{k+1} \tag{8}$$

where subscript except k denotes partial derivative with respect to $\boldsymbol{x}_k$ or $\boldsymbol{u}_k$ and superscript (k) denotes the value at discretized time k. $\delta\boldsymbol{x}_{k+1}$ is obtained from the equation of dynamical system (1) as follows:

$$\delta\boldsymbol{x}_{k+1} \simeq\ \boldsymbol{F}_{\boldsymbol{x}}^{(k)}\delta\boldsymbol{x}_k + \boldsymbol{F}_{\boldsymbol{u}}^{(k)}\delta\boldsymbol{u}_k + \frac{1}{2}\delta\boldsymbol{x}_k^T * \boldsymbol{F}_{\boldsymbol{xx}}^{(k)}\delta\boldsymbol{x}_k + \delta\boldsymbol{x}_k^T * \boldsymbol{F}_{\boldsymbol{xu}}^{(k)}\delta\boldsymbol{u}_k + \frac{1}{2}\delta\boldsymbol{u}_k^T * \boldsymbol{F}_{\boldsymbol{uu}}^{(k)}\delta\boldsymbol{u}_k \tag{9}$$

where the operator * is defined as $(A * B)_{jk} = A_i B_{ijk}$, where the subscripts denote the component in tensor notation.

Therefore, the right hand side of Bellman's principle of optimality (5) can be described as following form:

$$L_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + V_{k+1}^*(\boldsymbol{x}_{k+1}) = Q_0 + [q_{\boldsymbol{x}}^T\ q_{\boldsymbol{u}}^T]\begin{bmatrix}\delta\boldsymbol{x}_k \\ \delta\boldsymbol{u}_k\end{bmatrix} + \frac{1}{2}[\delta\boldsymbol{x}_k^T \quad \delta\boldsymbol{u}_k^T]\begin{bmatrix}Q_{\boldsymbol{xx}} & Q_{\boldsymbol{xu}} \\ Q_{\boldsymbol{xu}}^T & Q_{\boldsymbol{uu}}\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{x}_k \\ \delta\boldsymbol{u}_k\end{bmatrix} \tag{10}$$

where the coefficients Q ($Q_0, q_{\boldsymbol{x}}, q_{\boldsymbol{u}}$, ...) is defined in Appendix 6.1.

Assume $Q_{\boldsymbol{uu}}$ is a positive definite matrix, the optimal control variations can be obtained as stationary points of Eq.(10):

$$\delta\boldsymbol{u}_k = \boldsymbol{\alpha}_k + \beta_k\delta\boldsymbol{x}_k \tag{11}$$

where $\boldsymbol{\alpha}_k \coloneqq -Q_{\boldsymbol{uu}}^{-1}q_{\boldsymbol{u}}$ and $\beta_k \coloneqq -Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{xu}}^T$.

In backward sweep, calculate the optimal control policy $\boldsymbol{\alpha}_k$ and β_k to minimize $L_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + V_{k+1}^*(\boldsymbol{x}_{k+1})$. In forward sweep, update reference trajectory by using closed-loop optimal control $\boldsymbol{u}_k = \bar{\boldsymbol{u}}_k + \delta\boldsymbol{u}_k = \bar{\boldsymbol{u}}_k + \boldsymbol{\alpha}_k + \beta_k\delta\boldsymbol{x}_k$.

3. Stochastic Differential Dynamic Programming

To solve robust optimal low-thrust trajectory for uncertain system to minimize the expected value of objective function, we introduce Stochastic Differential Dynamic Programming (SDDP)[14][15]. The fundamental idea of SDDP is finding optimal control to minimize the expected value of cost-to-go function. We introduce unscented transform [16], which is well known in filtering problem, to obtain the expected value of cost-to-go function [14].

3.1. Unscented Transform

The Unscented Transform (UT) is the method to calculate the probability distribution of a random variable that undergoes a nonlinear transformation [16]. The basic idea of UT, which is similar to Monte-Carlo to estimate probability distribution by random sampling, is to estimate probability distribution by using a set of representative points, called *sigma points*.

Given n-dimensional random variable $\boldsymbol{x}$ with mean $\bar{\boldsymbol{x}}$ and covariance $P_{\boldsymbol{xx}}$, and nonlinear transformation $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, UT is used to estimate the mean $\bar{\boldsymbol{y}}$ and covariance $P_{\boldsymbol{yy}}$ of random variable $\boldsymbol{y}$ by following steps:

1) Calculate sigma-points $\boldsymbol{\mathcal{X}}_i$ and its weight W_i ($i = 0,1, \dots ,2n$):

$$\begin{aligned} \boldsymbol{\mathcal{X}}_0 &= \bar{\boldsymbol{x}} & W_0 &= \lambda/(n+\lambda) \\ \boldsymbol{\mathcal{X}}_i &= \bar{\boldsymbol{x}} + \left(\sqrt{(n+\lambda)P_{\boldsymbol{xx}}}\right)_i & W_i &= 1/2(n+\lambda) \\ \boldsymbol{\mathcal{X}}_{i+n} &= \bar{\boldsymbol{x}} - \left(\sqrt{(n+\lambda)P_{\boldsymbol{xx}}}\right)_i & W_{i+n} &= 1/2(n+\lambda) \end{aligned} \tag{12}$$

where $\lambda \in \mathbb{R}$ is free parameter, $(\cdot)_i$ means the i-th column of the matrix square root $\sqrt{(n+\lambda)P_{\boldsymbol{xx}}}$.

2) Obtain the set of transformed sigma points $\boldsymbol{\mathcal{Y}}_i$,

$$\boldsymbol{\mathcal{Y}}_i = \boldsymbol{f}(\boldsymbol{\mathcal{X}}_i) \tag{13}$$

3) The mean and covariance are given by using weight and transformed sigma-point as follows:

$$\bar{\boldsymbol{y}} = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{Y}}_i \tag{14}$$

$$P_{\boldsymbol{yy}} = \sum_{i=0}^{2n} W_i \{\boldsymbol{\mathcal{Y}}_i - \bar{\boldsymbol{y}}\}\{\boldsymbol{\mathcal{Y}}_i - \bar{\boldsymbol{y}}\}^T \tag{15}$$

Example of UT is shown in Fig.2. Here, nonlinear transformation is quadratic transformation, and a free parameter λ is chosen as -0.75.

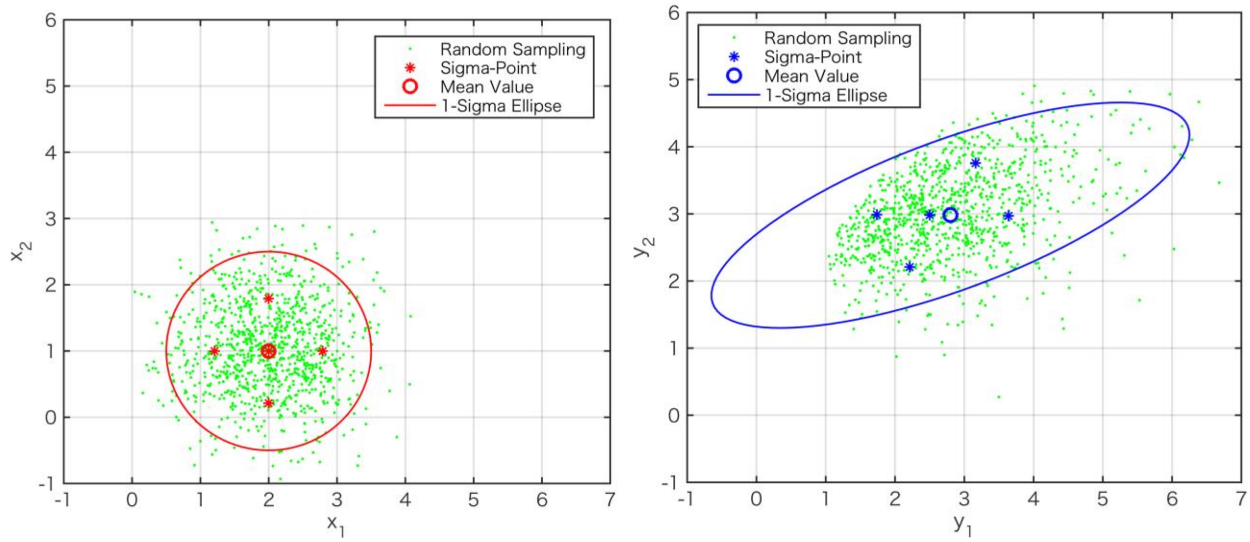

Figure 2. Unscented Transform (Before transform: Left, After transform: Right)

3.2. Stochastic Differential Dynamic Programming for State Uncertainty

We assume that the state has the uncertainty and it is formulated as:

$$\boldsymbol{X}_k = \boldsymbol{x}_k + \boldsymbol{w}_k \tag{16}$$

where $\boldsymbol{w}_k \sim \mathcal{N}\left(\boldsymbol{0}_n, P_{\boldsymbol{w}_k}\right)$ is uncertainty, $\boldsymbol{0}_n \in \mathbb{R}^n$ is null vector, and $P_{\boldsymbol{w}_k} \in \mathbb{R}^{n \times n}$ is covariance matrix. If there is no observation noise and time lag from observation to control, then we can employ the optimal feedback control law:

$$\boldsymbol{U}_k = \boldsymbol{u}_k + \beta_k(\boldsymbol{X}_k - \bar{\boldsymbol{x}}_k) \tag{17}$$

In a proposed SDDP, instead of solving Bellman's principle of optimality Eq.(5), we solve the following principle of optimality with expectation[11][15]:

$$V_k^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_k}\{E_{\boldsymbol{w}_k}[L_k(\boldsymbol{X}_k, \boldsymbol{U}_k;\ t_k) + V_{k+1}^*(\boldsymbol{X}_{k+1})]\} \tag{18}$$

where $E_{\boldsymbol{w}_k}[\cdot]$ is the expectation with respect to random variable $\boldsymbol{w}_k$, and the state vector $\boldsymbol{X}_{k+1}$ at discretized time $(k+1)$, which is influenced by uncertainty $\boldsymbol{w}_k$, is described as:

$$\boldsymbol{X}_{k+1} = \boldsymbol{F}_k(\boldsymbol{X}_k, \boldsymbol{U}_k). \tag{19}$$

The right hand side of Eq.(18) can be expanded as:

$$L_k(\boldsymbol{X}_k, \boldsymbol{U}_k;\ t_k) + V_{k+1}^*(\boldsymbol{X}_{k+1}) = Q_0(\bar{\boldsymbol{X}}_k, \bar{\boldsymbol{U}}_k) + [q_{\boldsymbol{x}}^T(\bar{\boldsymbol{X}}_k, \bar{\boldsymbol{U}}_k)\ q_{\boldsymbol{u}}^T(\bar{\boldsymbol{X}}_k, \bar{\boldsymbol{U}}_k)]\begin{bmatrix}\delta\boldsymbol{x}_k \\ \delta\boldsymbol{u}_k\end{bmatrix} + \cdots \tag{20}$$

where, we can easily derive $\delta\boldsymbol{X}_k = \delta\boldsymbol{x}_k$, $\delta\boldsymbol{U}_k = \delta\boldsymbol{u}_k$ by definition.

Therefore, the optimal control variations can be obtained as stationary points of Eq.(20) by:

$$\delta\boldsymbol{u}_k = \boldsymbol{\alpha}_k + \beta_k\delta\boldsymbol{x}_k \tag{21}$$

where we assume $E_{\boldsymbol{w}_k}[Q_{\boldsymbol{uu}}(\bar{\boldsymbol{X}}_k, \bar{\boldsymbol{U}}_k)]$ is a positive definite matrix, and $\boldsymbol{\alpha}_k$ and β_k in SDDP are:

$$\boldsymbol{\alpha}_k := -E_{\boldsymbol{w}_k}[Q_{\boldsymbol{uu}}(\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{U}}_k)]^{-1} E_{\boldsymbol{w}_k}[q_{\boldsymbol{u}}(\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{U}}_k)] \tag{22}$$

$$\beta_k := -E_{\boldsymbol{w}_k}[Q_{\boldsymbol{uu}}(\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{U}}_k)]^{-1} E_{\boldsymbol{w}_k}[Q_{\boldsymbol{xu}}^T(\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{U}}_k)] \tag{23}$$

To calculate $E_{\boldsymbol{w}_k}[Q_{\boldsymbol{uu}}(\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{U}}_k)]$, $E_{\boldsymbol{w}_k}[q_{\boldsymbol{u}}(\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{U}}_k)]$, and so on, we apply Unscented Transform in Section 3.1.

In the backward sweep in SDDP, we compute aforementioned equations. Meanwhile, in the forward sweep in SDDP, we propagate the mean value of state perturbed by uncertainty $\boldsymbol{w}_k$ using Unscented Transform and use the mean value as nominal state of SDDP reference trajectory.

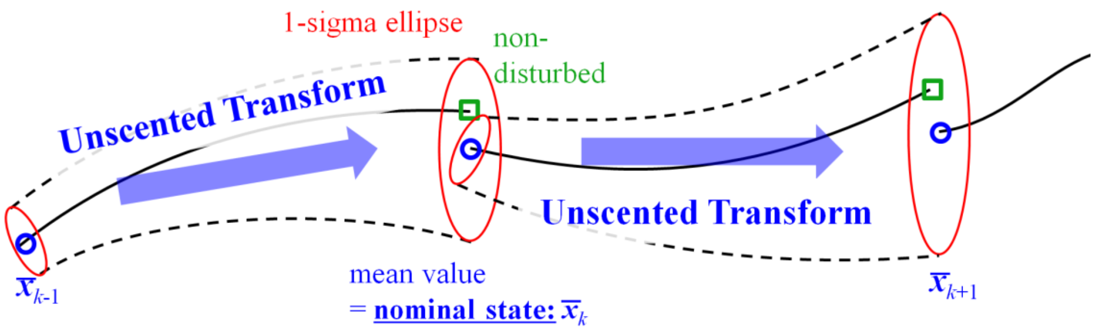

Figure 3. Forward Sweep in SDDP by Unscented Transform

We summarize the computational step in every iteration of SDDP. First, in backward sweep, we compute optimal control by Eq.(21), where we apply Unscented Transform to calculate expected value of coefficient Q. Then, using the robust optimal control law Eq.(21), we update the reference nominal trajectory in forward sweep, shown in Fig.3.

4. Numerical Example

Prior to applying a proposed SDDP to trajectory design problem, we demonstrate that our DDP algorithm successfully computes the correct solution, using classical brachistochrone problem as example [17].

4.1. Classical Brachistochrone Problem

The equation of motion of brachistochrone problem is described as:

$$\begin{aligned} x_{k+1} &= x_k + \Delta t \cdot v_x \\ y_{k+1} &= y_k + \Delta t \cdot v_y \\ v_{x,k+1} &= v_{x,k} + \Delta t \cdot \sqrt{v_0^2 - 2gy} \cdot \cos u \\ v_{y,k+1} &= v_{y,k} + \Delta t \cdot \sqrt{v_0^2 - 2gy} \cdot \sin u \end{aligned} \tag{24}$$

where $u \in \mathbb{R}$ is control variable, and v_0 and g are parameters which describe initial velocity and gravity constant, respectively. Given initial position $[x_1, y_1] = [0.0,\ 0.0]$, initial velocity $v_0 = 0.0$, and final position $[x_{N+1}, y_{N+1}] = [15.7,\ 10.0]$, find the optimal control law $\{u_k\}$ to minimize the transfer time. The optimal path obtained by DDP corresponds to the cycloid curve, which is known as theoretical solution, shown in Fig.4.

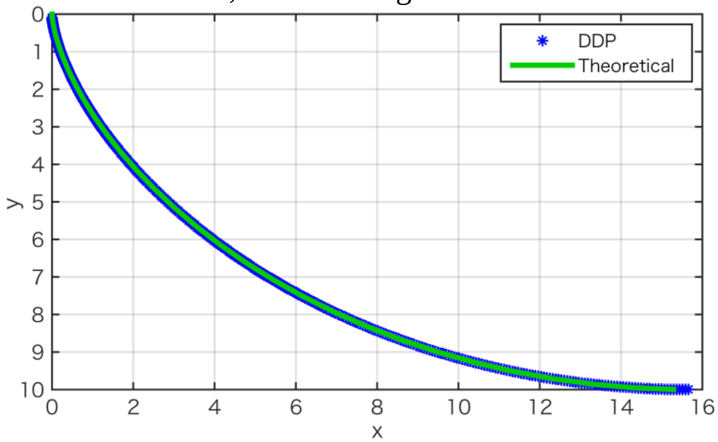

Figure 4. Trajectory of Classical Brachistochrone Problem

4.2. V-Infinity Leveraging Problem

The dynamical system, here we consider, is 2-dimensional two-body problem:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ -\frac{\mu}{(x^2+y^2)^{\frac{3}{2}}} x + T\cos u \\ -\frac{\mu}{(x^2+y^2)^{\frac{3}{2}}} y + T\sin u \end{bmatrix} \tag{25}$$

where $u \in \mathbb{R}$ is control variable, $\mu = 1.327 \times 10^{11} [km^3/m^2]$ is the gravity constant, and $T = 1.2 \times 10^{-4} [m/s^2]$ is thrust acceleration. The dynamical system is discretized by Runge-Kutta 4-th order method.

As for a deterministic optimal control problem, given initial state $[x_1, y_1] = [1.5 \times 10^8, 0.0](km)$, $[v_{x,1}, v_{y,1}] = [0.0, 29.8](km/s)$, final position $[x_f, y_f] = [1.5 \times 10^8,\ 0.0](km)$, and transfer time is 1 (year), find the optimal control law $\{u_k\}$ to minimize the objective function:

$$J = \Phi_{N+1}(\boldsymbol{x}_{N+1};\ t_{N+1}) = C_1 \cdot V_\infty(t_{N+1}) + C_2 \cdot \left(x_{N+1} - x_f\right)^2 + C_3 \cdot \left(y_{N+1} - y_f\right)^2 \tag{26}$$

where $V_\infty(t_{N+1}) \coloneqq \sqrt{\left(v_{x,N+1} - v_{x,1}\right)^2 + \left(v_{y,N+1} - v_{y,1}\right)^2}$, and weight is tuned as $C_1 = -10.0, C_2 = 1.0 \times 10^6, C_3 = 1.0 \times 10^6$. The solution is shown as DDP in Fig.5.

As for a robust-optimal control problem with state uncertainty, we consider following discretized stochastic dynamical system:

$$\boldsymbol{x}_k = \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_k \tag{27}$$

where, $\boldsymbol{F}_k(\cdot)$ is discretized dynamical system by Runge-Kutta 4-th order method with $\Delta t = 0.025$ [year], the unit is unified as [km] and [km/s], and $\boldsymbol{w}_k$ is a Gaussian random process with zero mean and covariance $P_{\boldsymbol{w}_k}$, which is described as:

$$P_{w_k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 8.9 \times 10^{-6} & 0 \\ 0 & 0 & 0 & 8.9 \times 10^{-6} \end{bmatrix} \tag{28}$$

Hence, the velocity disturbance is about 3[m/s]@1-sigma every 9[days]. The robust-optimal control problem is, when the same boundary condition as deterministic one is given, finding the optimal control law $\{u_k\}$ to minimize the expected value of cost-to-go function. The solution is shown as Stochastic DDP in Fig.5.

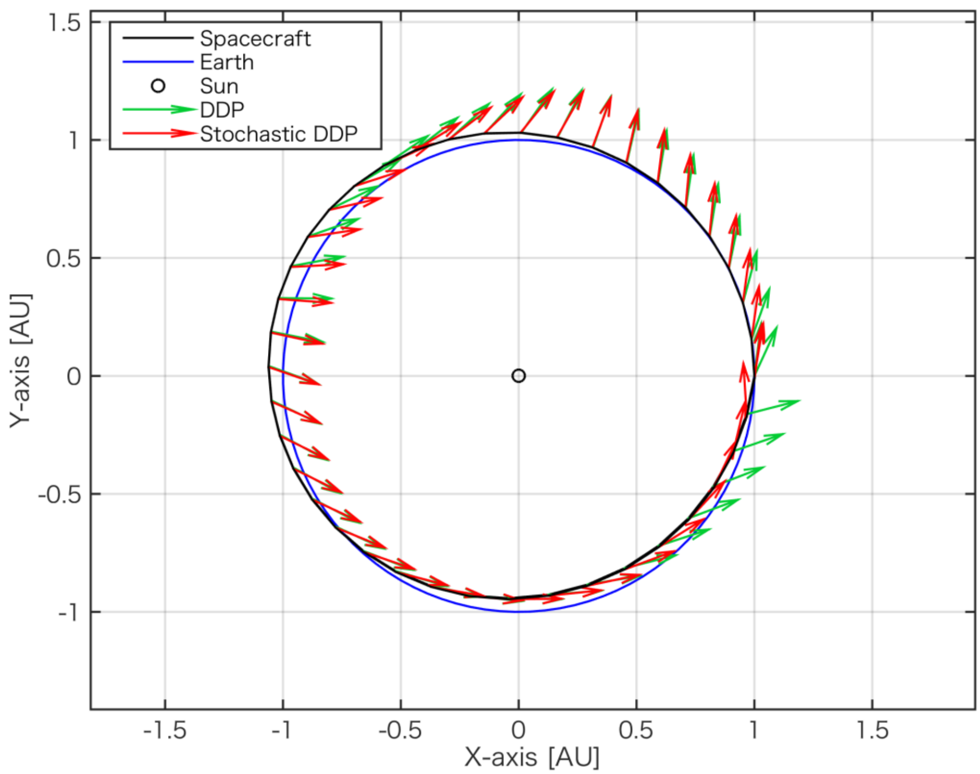

Figure 5. Trajectory of V-infinity Leveraging Problem

To evaluate the result of conventional DDP and a proposed SDDP, we apply Monte-Carlo method to both results. Both results obtained by DDP and SDDP has optimal feedback controller around nominal trajectory; therefore, the optimal feedback controller corrects the trajectories autonomously. We generate 5000 samples, and every sample has velocity Gaussian disturbance of 3[m/s]@1-sigma every 9[days]. The result is shown in Fig.6 and Fig.7.

As shown in Fig.6, only 30% of samples come back close to the Earth in DDP, while approximately 90% of samples do in SDDP, since the result of SDDP has more “margin” in optimal control than that of DDP. As shown in Fig.7, even though nominal V-infinity of SDDP (0.488[km/s]) is worse than one of DDP (0.575[km/s]), the expected value of V-infinity of SDDP (0.501[km/s]) is better than one of DDP (0.451[km/s]). We find that the performance of SDDP is better than DDP in the case that the system has uncertainty, as we expected.

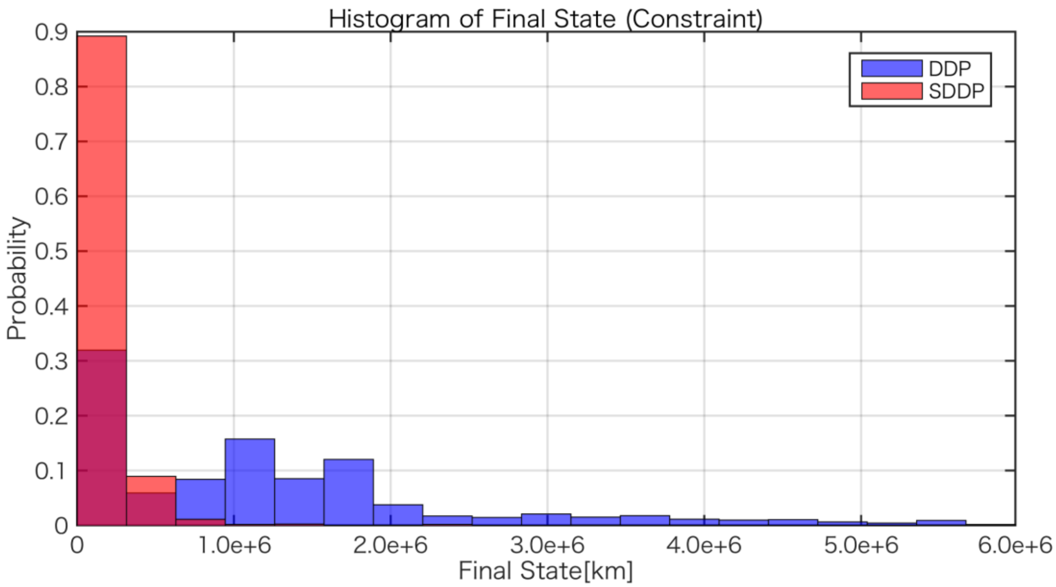

Figure 6. Result of Monte-Carlo Simulation (Terminal Position Deviation)

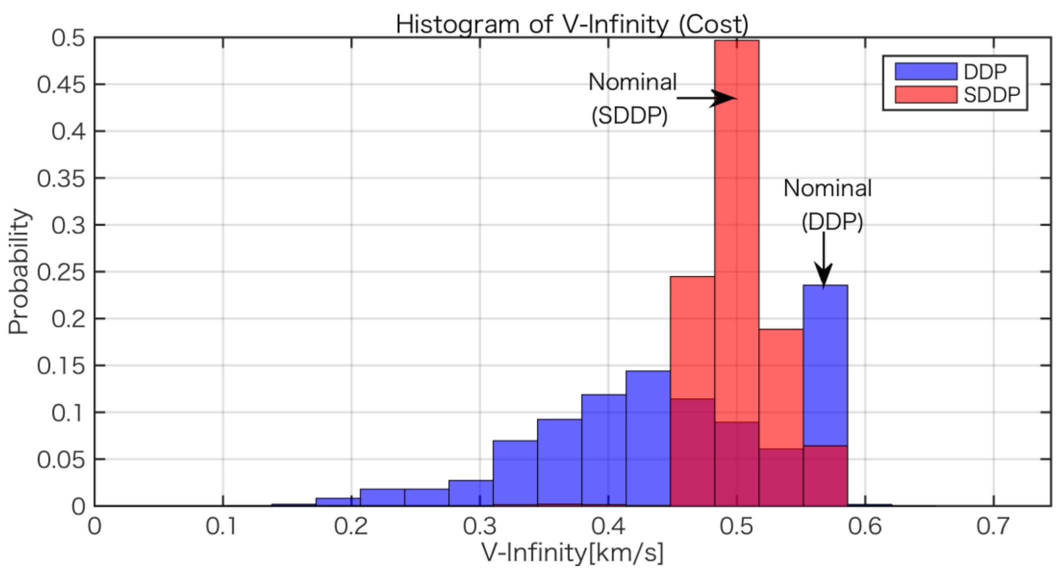

Figure 7. Result of Monte-Carlo Simulation (V-infinity at Terminal Condition)

5. Conclusion

This paper proposes a robust-optimal trajectory design method for uncertain system to minimize the expected value of objective function. The basic idea is solving Stochastic Differential Dynamic Programming (SDDP), which solve optimal control problem to minimize the expected value of cost-to-go function, with Unscented Transform, which is used to estimate the expected value. We apply a SDDP to V-infinity leveraging problem, and demonstrate that SDDP has better performance than conventional DDP when the system has uncertainty since the result of SDDP has “margin” in optimal control for future feedback. However, it is difficult to apply a proposed SDDP to bang-bang problem, such that the low-thrust trajectory designs with thrusting arc and coasting arc. We will extend a proposed SDDP to the bang-bang problem as future work.

6. Appendix

6.1. Coefficient Q of Quadratic Expansion of Bellman's Principle of Optimality

The coefficients Q ($Q_0, q_{\boldsymbol{x}}, q_{\boldsymbol{u}}, ...$) , which is introduced in Eq.(10), can be calculated by substituting Eqs.(7), (8), (9) to right hand side of Eq.(5), and defined as follow:

$$Q_0 \coloneqq L_k(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k) + V_{k+1}^*(\bar{\boldsymbol{x}}_{k+1}) \tag{29}$$

$$q_{\boldsymbol{x}} \coloneqq L_{\boldsymbol{x}}^{(k)} + V_{\boldsymbol{x}}^{*(k+1)} \boldsymbol{F}_{\boldsymbol{x}}^{(k)} \tag{30}$$

$$q_{\boldsymbol{u}} \coloneqq L_{\boldsymbol{u}}^{(k)} + V_{\boldsymbol{x}}^{*(k+1)} \boldsymbol{F}_{\boldsymbol{u}}^{(k)} \tag{31}$$

$$Q_{\boldsymbol{xx}} \coloneqq L_{\boldsymbol{xx}}^{(k)} + V_{\boldsymbol{x}}^{*(k+1)} * \boldsymbol{F}_{\boldsymbol{xx}}^{(k)} + \boldsymbol{F}_{\boldsymbol{x}}^{(k)T} V_{\boldsymbol{xx}}^{*(k+1)} \boldsymbol{F}_{\boldsymbol{x}}^{(k)} \tag{32}$$

$$Q_{\boldsymbol{xu}} \coloneqq L_{\boldsymbol{xu}}^{(k)} + V_{\boldsymbol{x}}^{*(k+1)} * \boldsymbol{F}_{\boldsymbol{xu}}^{(k)} + \boldsymbol{F}_{\boldsymbol{x}}^{(k)T} V_{\boldsymbol{xx}}^{*(k+1)} \boldsymbol{F}_{\boldsymbol{u}}^{(k)} \tag{33}$$

$$Q_{\boldsymbol{uu}} \coloneqq L_{\boldsymbol{uu}}^{(k)} + V_{\boldsymbol{x}}^{*(k+1)} * \boldsymbol{F}_{\boldsymbol{uu}}^{(k)} + \boldsymbol{F}_{\boldsymbol{u}}^{(k)T} V_{\boldsymbol{xx}}^{*(k+1)} \boldsymbol{F}_{\boldsymbol{u}}^{(k)} \tag{34}$$

7. References

[1] Betts, J.T. "Survey of Numerical Methods for Trajectory Optimization." Journal of Guidance, Control, and Dynamics, Vol. 21, No. 2, pp. 193-207, 1998.

[2] Conway, B. "Spacecraft Trajectory Optimization." Cambridge University Press, 2010.

[3] Rayman, M.D., Fraschetti, T.C. et.al. "Coupling of System Resource Margins Through the Use of Electric Propulsion: Implications in Preparing for the Dawn Mission to Ceres and Vesta." Acta Astronautica, Vol. 60, Nos. 10-11, pp. 930-938, 2007.

[4] Chisci, L., Rossiter, J.A. and Zappa, G. "Systems with Persistent Disturbances: Predictive Control with Restricted Constraints." Automatica, Vol. 37, No. 7, pp. 1019-1028, 2001.

[5] Richards, A. and How, J. "Robust Stable Model Predictive Control with Constraint Tightening." Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, June, 2006.

[6] Laipert, F.E. and Longuski, J.M. "Automated Missed-Thrust Propellant Margin Analysis for Low-Thrust Trajectories." Journal of Spacecraft and Rockets, Vol. 52, No. 4, pp. 1135-1143, 2015.

[7] Olympio, J.T. and Yam, C.H. "Deterministic Method for Space Trajectory Design with Mission Margin Constraints." 61th International Astronautical Congress, 2010.

[8] Olympio, J.T. “Designing Robust Low-Thrust Interplanetary Trajectories Subject to One Temporary Engine Failure.” 20th AAS/AIAA Space Flight Mechanics, AAS10-171, San Diego, CA, USA, 2010.

[9] Scokaert P.O.M. and Mayne, D.Q. “Min-Max Feedback Model Predictive Control for Constrained Linear Systems.” IEEE Transactions of Automatic Control, Vol.43, No.8, pp. 1136-1142, 1998.

[10] Diehl, M. and Bjornberg, J. “Robust Dynamic Programming for Min-Max Model Predictive Control of Constrained Uncertain Systems.” IEEE Transactions on Automatic Control, Vol.49, No.12, pp.2253-2257, 2004.

[11] Jacobson, D.H. and Mayne, D.Q. “Differential Dynamic Programming.” American Elsevier Publishing Company, Inc., 1970.

[12] Whiffen, G. “Static/Dynamic Control for Optimizing A Useful Objective.” US Patent 6,296,741, Dec.17, 2002.

[13] Lantoine, G. and Russell, R.P. “A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory” Journal of Optimization Theory and Applications, Vol.154, No.2, pp.382-417, 2012.

[14] Todorov, E. and Tassa, Y. “Iterative Local Dynamic Programming.” Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL '09. IEEE Symposium on, Nashville, TN, USA, March, 2009.

[15] Theodorou, E., Tassa, Y. and Todorov, E. “Stochastic Differential Dynamic Programming.” American Control Conference, Baltimore, MD, June-July, 2010.

[16] Julier, S. J. and Uhlmann, J. K. “A New Extension of the Kalman Filter to Nonlinear Systems.” Proceedings of SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 1997.

[17] Bryson, A.E. and Ho, Y.C. “Applied Optimal Control, Optimization, Estimation, and Control.” Hemisphere Publishing Corporation, Washington, D.C., USA, 1975.