DEVELOPMENT OF *SAMURAI* – SIMULATION AND ANIMATION MODEL FOR ROCKETS WITH ADJUSTABLE ISP

Tadashi Sakai,* **John R. Olds**,† **and Kristina Alemany**‡

An interplanetary trajectory calculation application *SAMURAI* - Simulation and Animation Model Used for Rockets with Adjustable I_{sp} - has been developed. *SAMURAI* is written in C++ and calculates transfer trajectories with variable thrust, variable I_{sp} (VASIMR type) engines as well as conventional constant low thrust, constant I_{sp} engines and high thrust engines. *SAMURAI* utilizes a calculus of variations algorithm to evaluate the thrust history that minimizes the fuel consumption from one planet to another. A trajectory with a planetary swing-by can also be calculated. After calculation, a 3D animation of the resulting transfer trajectory is created and can be viewed with a web browser using VRML. In this paper, equations and methods used in *SAMURAI*, and the capabilities of this application are presented. A few examples including a round trip from Earth to Mars have been analyzed, and trajectories with variable I_{sp} engines, constant I_{sp} engines, and high thrust engines are compared.

INTRODUCTION

Currently there are several projects ongoing worldwide relating to rocket engines that can modulate their I_{sp}. These projects include mechanical tests at ground facilities as well as trajectory simulations with computers. VASIMR (VAriable Specific Impulse Magnetoplasma Rocket), under development at the NASA Johnson Space Center, is a well-known example of a rocket engine with a variable I_{sp} and constant power. There have been several numerical research efforts for VSI, power-limited rocket (VASIMR-type) trajectory optimization problems [1] [2] [3] [4] [5] [6], each illustrating interesting features of VSI engines. Conventional low thrust rockets with fixed I_{sp} (such as ion thrusters) are defined as CSI engines (Constant Specific Impulse engines) in this paper.

To further investigate the characteristics of these engines, an interplanetary trajectory calculation application *SAMURAI* has been created to calculate trajectories using several types of engines introduced above. Because this application not only calculates trajectories with VSI engines, but also calculates trajectories with conventional CSI engines and high thrust engines, users can compare the results for these different types of engines and determine which engine is desirable for a mission. This application also has the capability of calculating swing-by trajectories with one swing-by planet. A 3D animation of the resulting transfer trajectory is created and can be viewed with a web browser using VRML.

In reality, I_{sp} for VSI engines can not take very high or very low values. Because of physical constraints such as heat and exhaust velocity, there are upper and lower limits on I_{sp}. In this paper, two types of VSI engines are considered: an engine whose I_{sp} is constrained by an upper bound, and an engine whose I_{sp} is unconstrained. Also, for CSI engines, two types of engines are considered: an engine that is always on throughout the mission, and an engine that has the capability of turning on and off.

*Graduate Research Assistant, Space Systems Design Laboratory, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, Phone: (404)894-7783, E-mail: tadashi_sakai@ae.gatech.edu, Student member AAS

†Associate Professor, Space Systems Design Laboratory, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, Phone: (404)894-6289, E-mail: john.olds@ae.gatech.edu

‡Graduate Research Assistant, Space Systems Design Laboratory, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, Phone: (404)894-7783, E-mail: kristina_alemany@ae.gatech.edu

The problem to be solved is defined as follows: find a history of control variables (thrust direction and magnitude) that achieves the minimum fuel consumption of a spacecraft traveling from the departure planet (initial position and velocity) to the arrival planet (final position and velocity) with a prescribed flight time. Control variables may have constraints.

Solution methods for this kind of trajectory optimization problem are typically identified as either direct methods or indirect methods. Direct methods discretize the optimization problem through events and phases, and the subsequent problem is solved using nonlinear programming techniques. These techniques include shooting, multiple shooting, and transcription or collocation methods. In the shooting method, the control history is discretized as a polynomial, with the trajectory variables as functions of the integrated equations of motion. In the collocation method, the trajectory is discretized over an entire trajectory as a set of polynomials for both state variables and control variables. Solutions obtained with these direct methods are generally considered sub-optimal due to the discretization of either the state or controls, or both.

Indirect methods use calculus of variations techniques to characterize the optimization problem as a two-point boundary value problem. The optimal control scheme is an indirect method. The optimal control uses a first variation technique to determine necessary conditions for an optimum, and second variation techniques are used to determine whether the point is the minimum, the maximum, or a saddle point [7]. This method involves applying calculus of variations principles and solving the corresponding two point boundary value problem [8]. Initial estimates of the Lagrange multipliers must be provided, but since they do not have physical meanings, guessing the initial values of the Lagrange multiplier is difficult and may lead to problems with convergence.

Each of the above methods has pros and cons: indirect methods are difficult to formulate, whereas with direct methods, mathematical suboptimal solutions are obtained. In this research, an indirect method is selected since it calculates an optimal solution rather than a suboptimal solution. The equations of motion used in this research are not very complicated and can be implemented in the application without difficulties. Also, there are several excellent literature sources available for programming with indirect methods [2][6][7][9][10].

In this paper, methods and equations of motion used in the application *SAMURAI*, and features of this application are presented. Then the validation with other reliable interplanetary trajectory calculation applications and an example of its application to a typical interplanetary trajectory are described.

OPTIMIZATION PROBLEMS

Without Terminal Constraints on State Variables

Consider the dynamic system described by the following nonlinear differential equations[§]:

$$x = \dot{f}[x(t), u(t), t], \quad x(t_i) \text{ given}, \quad t_i \leq t \leq t_f, \tag{1}$$

where $x(t)$, a n-vector function, is determined by $u(t)$, a m-vector function. Suppose we wish to choose the history of control variables $u(t)$ to minimize the performance index, J (scalar), of the form

$$J = \phi[x(t_f), t_f] + \int_{t_i}^{t_f} L[x(t), u(t), t]\, dt \tag{2}$$

where $\phi[x(t_f), t_f]$ is a scalar function that will be minimized, and $L[x(t), u(t), t]$ is the Lagrangian. For

[§]A table of notation can be found at the end of this paper.

convenience, define a scalar function H as follows:

$$H[x(t), u(t), t] = L[x(t), u(t), t] + \lambda^T(t) f[x(t), u(t), t]. \tag{3}$$

Then, to find a control vector function $u(t)$ that produces a stationary value of the performance index J, the following differential equations must be solved:

$$\dot{x} = f(x, u, t) \tag{4}$$

$$\dot{\lambda} = -\left(\frac{\partial f}{\partial x}\right)^T \lambda - \left(\frac{\partial L}{\partial x}\right)^T \tag{5}$$

$$\frac{\partial H}{\partial u} = 0 \qquad \text{or} \qquad \left(\frac{\partial f}{\partial u}\right)^T \lambda + \left(\frac{\partial L}{\partial u}\right)^T = 0. \tag{6}$$

With Terminal Constraints on State Variables

If q components of the state vector $x(t)$ are prescribed at t_f, the following variation of $u(t)$, $\delta u(t)$ is used to decrease J:

$$\delta u = -k \left\{ \frac{\partial L}{\partial u} + [p + \nu R]^T \frac{\partial f}{\partial u} \right\} \tag{7}$$

where k is a positive scalar constant and p is a n-component vector and R is a $n \times q$ matrix:

$$\dot{p} = -\left(\frac{\partial f}{\partial x}\right)^T p - \left(\frac{\partial L}{\partial x}\right)^T \tag{8}$$

$$p_j(t_f) = \begin{cases} 0, & j = 1, \cdots, q \\ (\partial \phi / \partial x_j)|_{t=t_f}, & j = q+1, \cdots, n \end{cases} \tag{9}$$

$$\dot{R}_i = -\left(\frac{\partial f}{\partial x}\right)^T R_i \tag{10}$$

$$R_{ij}(t_f) = \begin{cases} 0, & i \neq j \\ 1, & i = j, \quad j = 1, \cdots n \end{cases} \tag{11}$$

and

$$\nu = -Q^{-1} g, \tag{12}$$

where Q is a $q \times q$ matrix and g is a q-component vector:

$$Q_{ij} = \int_{t_i}^{t_f} R^T \left(\frac{\partial f}{\partial u}\right) \left(\frac{\partial f}{\partial u}\right)^T R \, dt, \quad i, j = 1, \cdots, q, \tag{13}$$

$$g_i = \int_{t_i}^{t_f} R^T \frac{\partial f}{\partial u} \left[\left(\frac{\partial f}{\partial u}\right)^T p + \left(\frac{\partial L}{\partial u}\right)^T \right] dt, \quad i = 1, \cdots, q. \tag{14}$$

Inequality Constraints on Control Variables

Suppose that there is an inequality constraint on the system:

$$C(u(t), t) \leq 0 \tag{15}$$

where $u(t)$ is the m-component control vector, $m \geq 2$, and C is a scalar function. For example, for a requirement that the I_{sp} be less than or equal to 30,000 sec, C is expressed as $C = I_{sp} - 30,000 \leq 0$. If

we define the Hamiltonian with a Lagrange multiplier $\mu(t)$ as

$$H = \lambda^T f + L + \mu^T C, \tag{16}$$

the necessary condition on H is

$$H_u = \lambda^T f_u + L_u + \mu^T C_u = 0 \tag{17}$$

$$\text{and} \quad \mu \begin{cases} \geq 0, & C = 0, \\ = 0, & C < 0. \end{cases} \tag{18}$$

The positivity of the multiplier μ when $C = 0$ is interpreted as the requirement that the gradient of the original Hamiltonian $(\lambda^T f_u + L_u)$ be such that improvement can only occur by violating the constraints.

The differential equations for costate vectors are

$$\dot{\lambda}^T(t) = -\frac{\partial H}{\partial x} = -\frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x} - \mu^T C_x = -\frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x}. \tag{19}$$

Therefore, Eqn. 5 can be used to calculate costate vectors because C is not a function of x. Boundary conditions should be chosen so that the initial and terminal constraints for state variables are satisfied.

Bang-off-bang Control

This type of control is applied to the fixed-time, minimum-fuel problem with constrained input magnitude. For example, a CSI rocket that can turn its engine on/off as needed would obey this control law. Consider a problem with the following linear system[10].

$$\dot{x} = Ax + Bu \tag{20}$$

Assume that the fuel used in each component of the input is proportional to the magnitude of that component. Then the cost function to be minimized is

$$J(t_i) = \int_{t_i}^{t_f} \sum_{i=1}^{m} c_i |u_i(t)| dt, \tag{21}$$

where c_i is a component of a m-vector $C = [c_1 \ c_2 \ \cdots \ c_m]^T$ and $u_i(t)$ is a component of a m-vector $|u(t)| = [|u_1| \ \ |u_2| \ \cdots \ |u_m|]^T$. Suppose that the control is constrained as

$$|u(t)| \leq 1 \qquad t_i \leq t \leq t_f. \tag{22}$$

The Hamiltonian is

$$H = C^T |u| + \lambda^T (Ax + Bu) \tag{23}$$

and according to the Pontryagin's minimum principle, the optimal control must satisfy

$$C^T |u^*| + (\lambda^*)^T (Ax^* + Bu^*) \leq C^T |u| + (\lambda^*)^T (Ax^* + Bu) \tag{24}$$

for all admissible $u(t)$. $(*)$ denotes optimal quantities. This equation can be reduced to

$$C^T |u^*| + (\lambda^*)^T Bu^* \leq C^T |u| + (\lambda^*)^T Bu. \tag{25}$$

If we assume that all of the m components of the control variables are independent, then

$$|u_i^*| + \frac{(\lambda^*)^T b_i u_i^*}{c_i} \leq |u_i| + \frac{(\lambda^*)^T b_i u_i}{c_i}, \tag{26}$$

where b_i are the columns of B. Since

$$|u_i| = \begin{cases} u_i, & u_i \geq 0 \\ -u_i, & u_i \leq 0 \end{cases} \tag{27}$$

we can write the quantity we are trying to minimize by selection of $u_i(t)$ as

$$q_i(t) = |u_i| + \frac{b_i^T \lambda u_i}{c_i} = \begin{cases} \left(1 + b_i^T \lambda / c_i\right) |u_i|, & u_i \geq 0 \\ \left(1 - b_i^T \lambda / c_i\right) |u_i|, & u_i \leq 0 \end{cases} \tag{28}$$

Fig. 1 shows the relationship between q_i and $b_i^T \lambda u_i$ for $u_i = 1, u_i = 0$, and $u_i = -1$. When $-1 < u_i(t) < 1$, $q_i(t)$ takes on values inside the shaded area. Therefore, if we neglect the singular points ($b_i^T \lambda / c_i = 1$ or -1), the control can be expressed as

$$u_i(t) = \begin{cases} 1, & b_i^T \lambda / c_i < -1 \\ 0, & -1 < b_i^T \lambda / c_i < 1 \\ -1, & 1 < b_i^T \lambda / c_i. \end{cases} \tag{29}$$

This is called a bang-off-bang control law.

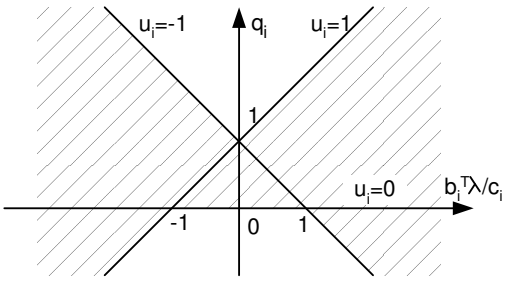

Figure 1: Bang-off-bang Control: Choosing Control to Minimize q_i.

EQUATIONS OF MOTION

From rocket propulsion fundamentals and using Newton's law for a variable mass body, the equation of motion of a spacecraft is

$$m\ddot{\vec{r}} = \dot{m}\vec{c} + m\vec{g} \tag{30}$$

where $\vec{c}$ is the exhaust velocity, and $\dot{m}$ is mass flow rate. This equation of motion is expressed in Cartesian coordinates by a set of differential equations for a position vector $\vec{r}$ and a velocity vector $\vec{V}$. The spacecraft's mass is also obtained by a differential equation for the mass m.

$$f = \begin{bmatrix} \dot{\vec{r}} \\ \dot{\vec{V}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \vec{V} \\ -\mu\vec{r}/r^3 + \vec{T}/m \\ -T^2/2P_J \end{bmatrix} \tag{31}$$

In order to minimize fuel consumption, the thrust vector $\vec{T}$ should be appropriately controlled throughout the trajectory. $\vec{T}$ is the only variable that can be controlled and is expressed by a control vector $\vec{u}$. The expression for $\vec{u}$ is defined for each different type of problem shown in the subsequent few pages. $\vec{u}$ is defined so that each problem is solved most effectively. Note that $\dot{m}$, P_J, I_{sp}, and T have the following relationship:

$$P_J = -\frac{T^2}{2\dot{m}} \tag{32}$$

$$T = -\dot{m} g_0 I_{sp} \tag{33}$$

VSI – No Constraints on I_{sp}

For this problem, control variables are the same as the components of the thrust vector.

$$\vec{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{34}$$

Then the equations of motion are

$$f = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ -\mu x/r^3 + T_x/m \\ -\mu y/r^3 + T_y/m \\ -\mu z/r^3 + T_z/m \\ -(T_x^2 + T_y^2 + T_z^2)/2P_J \end{bmatrix}. \tag{35}$$

This type of problem can be solved with the method explained in the subsection on VSI Constrained I_{sp} in the section SAMURAI.

VSI – Inequality Constraints on I_{sp}

For this type of problem, the following control variables are used:

$$\vec{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} l_x \\ l_y \\ l_z \\ T \end{bmatrix} \tag{36}$$

where T is the magnitude of thrust, l_x, l_y, l_z are the direction cosines of the direction of thrust in the inertial frame, and are subject to $l_x^2 + l_y^2 + l_z^2 = 1$. Then the equations of motion are

$$f = \begin{bmatrix} \dot{\vec{r}} \\ \dot{\vec{V}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \vec{V} \\ -\mu\vec{r}/r^3 + \vec{l}T/m \\ -T^2/2P_J. \end{bmatrix} \tag{37}$$

. The Hamiltonian of this system is

$$\begin{aligned} H &= \vec{\lambda}_r \cdot \vec{V} - \frac{\mu}{r^3}\vec{\lambda}_V \cdot \vec{r} + \vec{l} \cdot \vec{\lambda}_V T/m - \lambda_m T^2/2P_J \\ &= \vec{\lambda}_r \cdot \vec{V} - \frac{\mu}{r^3}\vec{\lambda}_V \cdot \vec{r} - \frac{\lambda_m}{2P_J}\left(T - \frac{\vec{l} \cdot \vec{\lambda}_V P_J}{m\lambda_m}\right)^2 + \frac{(\vec{l} \cdot \vec{\lambda}_V)^2 P_J}{2m^2\lambda_m}. \end{aligned} \tag{38}$$

According to the Pontryagin's maximum principle, the thrust vector must be selected in such a manner so as to maximize H at each instant of time. Therefore, we choose $\vec{l}$ parallel to $\vec{\lambda}_V$. Then $\vec{l}$ and T are expressed as

$$\vec{l} = \begin{bmatrix} l_x & l_y & l_z \end{bmatrix}^T = \vec{\lambda}_V/\lambda_V, \qquad \vec{\lambda}_V = \begin{bmatrix} \lambda_u & \lambda_v & \lambda_w \end{bmatrix}^T \tag{39}$$

$$T = \frac{P_J \lambda_V}{m\lambda_m} \tag{40}$$

. The vector $\vec{l}$ is called the primer vector[12]. The equations of motion and the control vector are

$$f = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ -\mu x/r^3 + l_x T/m \\ -\mu y/r^3 + l_y T/m \\ -\mu z/r^3 + l_z T/m \\ -T^2/2P_J \end{bmatrix}, \qquad \vec{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} l_x \\ l_y \\ l_z \\ T \end{bmatrix}. \tag{41}$$

Therefore, solving this type of problem is the same as finding the Lagrange multipliers.

CSI – Continuous Thrust

For continuous thrust, the propellant mass is proportional to the time of flight. That means that if the thrust magnitude is the same for any two missions with the same times of flight, the fuel requirements are the same. The problems dealt with in this research are fixed time problems, so the propellant mass cannot be used as the performance index. For this type of problem, the following performance index is used:

$$J = (u(t_f) - u_{target})^2 + (v(t_f) - v_{target})^2 + (w(t_f) - w_{target})^2 \tag{42}$$

and the terminal constraints are

$$\psi = \begin{bmatrix} x(t_f) - x_{target} \\ y(t_f) - y_{target} \\ z(t_f) - z_{target} \end{bmatrix} = 0. \tag{43}$$

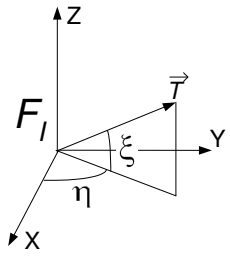

Figure 2: In-plane Thrust Angle η and Out-of-plane Thrust Angle ξ in the Inertial Frame.

Because the thrust magnitude is fixed, there are two control variables.

$$\vec{u} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} \eta \\ \xi \end{bmatrix} \tag{44}$$

Then the equations of motion are

$$f = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ -\mu x/r^3 + T\cos\eta\cos\xi/m \\ -\mu y/r^3 + T\sin\eta\cos\xi/m \\ -\mu z/r^3 + T\sin\xi/m \\ -T^2/2P_J \end{bmatrix}. \tag{45}$$

CSI – Bang-off-bang Control

For this type of problem, the thrust level is restricted so that it can take either the maximum value (T_{max}) or the minimum value (0). Because $T = 2P_J/c$, the Hamiltonian can be expressed as follows using the primer vector.

$$\begin{aligned} H &= \vec{\lambda}_r \cdot \vec{V} - \frac{\mu}{r^3}\vec{\lambda}_V \cdot \vec{r} + T\left(\vec{l} \cdot \vec{\lambda}_V/m - \lambda_m/c\right) \\ &= \vec{\lambda}_r \cdot \vec{V} - \frac{\mu}{r^3}\vec{\lambda}_V \cdot \vec{r} + T\,S \end{aligned} \tag{46}$$

where S is the switching function

$$S = \vec{l} \cdot \vec{\lambda}_V / m - \lambda_m / c \tag{47}$$

. To maximize the Hamiltonian, we choose the thrust level according to the switching function.

$$\begin{cases} \text{when } S > 0,\ T = T_{max} \\ \text{when } S < 0,\ T = 0 \end{cases} \tag{48}$$

The equations of motion and the control vector are the same as the VSI constrained case.

$$f = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ -\mu x/r^3 + l_x T/m \\ -\mu y/r^3 + l_y T/m \\ -\mu z/r^3 + l_z T/m \\ -T^2/2P_J \end{bmatrix}, \qquad \vec{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} l_x \\ l_y \\ l_z \\ T \end{bmatrix} \tag{49}$$

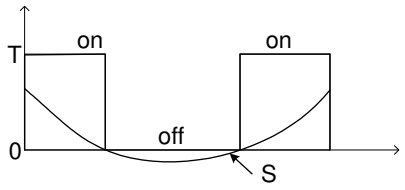

Figure 3: Switching Function and Switching Times for Bang-Off-Bang Control.

High Thrust

When a high thrust propulsion system (mostly chemical) is used, burn time is ususally very short compared to the entire mission duration. At the beginning of the mission the spacecraft fires an engine to gain enough velocity to reach the target planet, and once it approaches the target planet, it burns again for either an orbital insertion or a landing. This burn process can be simulated with two instantaneous burns at the beginning and at the end, and the rest of the time the spacecraft is assumed to obey Newton's law. Therefore a high thrust trajectory can be solved without integrating the equations of motion once we know positions of the planets. The Gauss problem (or Lambert's problem) solves such a trajectory.

The Gauss problem is defined as follows: Find $\vec{v}_1$ and $\vec{v}_2$ from a given $\vec{r}_1$, $\vec{r}_2$, the time of flight t from $\vec{r}_1$ to $\vec{r}_2$, and the direction of motion. The two vectors $\vec{r}_1$ and $\vec{r}_2$ uniquely define the plane of the transfer orbit unless they are collinear and in opposite directions($\Delta\nu = \pi$, $\Delta\nu$ is the angle between $\vec{r}_1$ and $\vec{r}_2$), and the relationship between the four vectors $\vec{r}_1$, $\vec{r}_2$, $\vec{v}_1$, and $\vec{v}_2$ are expressed by two scalar functions, f and g, and their time derivatives as follows:

$$\vec{r}_2 = f\vec{r}_1 + g\vec{v}_1 \tag{50}$$

$$\vec{v}_1 = \frac{\vec{r}_2 - f\vec{r}_1}{g} \tag{51}$$

$$\vec{v}_2 = \dot{f}\vec{r}_1 + \dot{g}\vec{v}_1 \tag{52}$$

where

$$f = 1 - \frac{r_2}{p}(1 - \cos\Delta\nu) \tag{53}$$

$$g = \frac{r_1 r_2 \sin\Delta\nu}{\sqrt{\mu p}} \tag{54}$$

$$\dot{f} = \sqrt{\frac{\mu}{p}} \frac{(1 - \cos\Delta\nu)}{\sin\Delta\nu} \left(\frac{1 - \cos\Delta\nu}{p} - \frac{1}{r_1} - \frac{1}{r_2} \right) \tag{55}$$

$$\dot{g} = 1 - \frac{r_1}{p}(1 - \cos\Delta\nu). \tag{56}$$

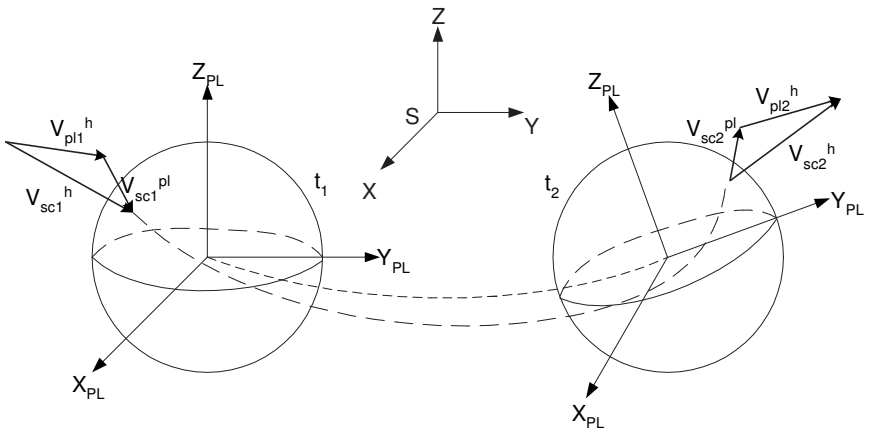

Figure 4: Schematic Diagram of Forming of Gravity Assist Maneuver.

The methods to solve the Gauss problems are described in several references such as Bate, Mueller, and White [14].

Swing-by

The use of a swing-by is a technique that is referred to as a "gravity assist"[16], and is used to reduce the propulsive velocity budget. Suppose that the position and velocity of the spacecraft in the heliocentric coordinate system ($\vec{r}_{sc1}^{\,h}$ and $\vec{V}_{sc1}^{\,h}$) are known at time t_1. The spacecraft enters the sphere of influence of a swing-by body whose position in the heliocentric coordinates is $\vec{r}_{pl1}^{\,h} = \vec{r}_{pl}(t_1)$ and velocity is $\vec{V}_{pl1}^{\,h} = \vec{V}_{pl}(t_1)$ as shown in Fig.)[15]. The coordinates are transferred from heliocentric coordinates to planetocentric coordinates and the position and velocity vector of the spacecraft with respect to the swing-by planet($\vec{r}_{sc1}^{\,pl}$ and $\vec{V}_{sc1}^{\,pl}$) are obtained.

$$\vec{r}_{sc1}^{\,pl} = [x_1\ y_1\ z_1]^T = R[h \to pl](\vec{r}_{rc1}^{\,h} - \vec{r}_{pl1}^{\,h}), \quad |\vec{r}_{sc1}^{\,pl}| = r_{SOI} \tag{57}$$

$$\vec{V}_{sc1}^{\,pl} = [u_1\ v_1\ w_1]^T = R[h \to pl](\vec{V}_{rc1}^{\,h} - \vec{V}_{pl1}^{\,h}) \tag{58}$$

Here r_{SOI} is the radius of the planetary sphere of influence (SOI) and $R[h \to pl]$ is the transformation matrix from heliocentric coordinates to planetocentric coordinates at this moment.

The gravity assist maneuver results in a rotation of the spacecraft velocity vector after hyperbolic flyby of the planet. The angle of rotation of the spacecraft velocity is

$$\phi = 2 \arctan \frac{\mu_{pl}}{\beta v_\infty^2}. \tag{59}$$

To determine the coordinates of the spacecraft "exit" point on the sphere of influence, the following relationships are used and shown in Fig.):

$$\phi^* = \pi + \phi - 2\gamma; \quad \gamma = \arcsin \frac{\beta}{r_{SOI}}. \tag{60}$$

Here μ_{pl} is the gravitational parameter of the swing-by planet, $v_\infty = (v_1^2 - 2\mu_{pl}/r_1)^{1/2}$ is the hyperbolic excess velocity of the spacecraft with respect to the swing-by planet, and *beta* is the impact parameter. Because there is no propulsive energy added to the spacecraft inside the SOI, $v_\infty = |\vec{V}_{sc1}^{\,pl}| = |\vec{V}_{sc2}^{\,pl}|$.

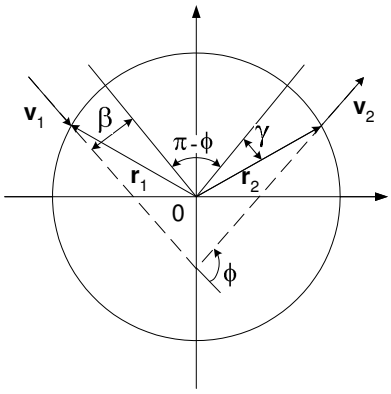

Figure 5: Swing-by: Inside the SOI.

Note that the following restriction may be imposed on β:

$$\beta \geq \beta_{min} = r_{p_{min}} \sqrt{1 + \frac{2\mu_{pl}}{r_{p_{min}} v_\infty^2}}, \tag{61}$$

where $r_{p_{min}}$ is the minimum admissible distance in the periapsis of the swing-by parabola and is determined by

$$r_{p_{min}} = r_{pl} + h_{atm} \tag{62}$$

where r_{pl} is the radius of the planet and h_{atm} is the height of the atmosphere if one exists.

With the above expressions for ϕ and ϕ^*, the coordinates of the spacecraft exit point from the sphere of influence, $\vec{r}_{sc2}^{\,pl} = [x_2 \ y_2 \ z_2]^T$, and the coordinates of the spacecraft velocity in the planetocentric coordinate system, $\vec{V}_{sc2}^{\,pl} = [u_2 \ v_2 \ w_2]^T$, can be written as:

$$\vec{r}_{sc2}^{\,pl} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \Omega(\phi^*) \vec{r}_{sc1}^{\,pl} = \Omega(\phi^*) \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{63}$$

$$\vec{V}_{sc2}^{\,pl} = \begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} = \Omega(\phi) \vec{V}_{sc1}^{\,pl} = \Omega(\phi) \begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} \tag{64}$$

. The transformation matrix Ω is written as

$$\Omega = \begin{bmatrix} h_x h_x (1 - \cos\phi) + \cos\phi & h_x h_y (1 - \cos\phi) - h_z \sin\phi & h_z h_x (1 - \cos\phi) + h_y \sin\phi \\ h_x h_y (1 - \cos\phi) + h_z \sin\phi & h_y h_y (1 - \cos\phi) + \cos\phi & h_y h_z (1 - \cos\phi) - h_x \sin\phi \\ h_z h_x (1 - \cos\phi) - h_y \sin\phi & h_y h_z (1 - \cos\phi) + h_x \sin\phi & h_z h_z (1 - \cos\phi) + \cos\phi \end{bmatrix} \tag{65}$$

where $[h_x, h_y, h_z]$ are the unit vector of angular momentum, $\vec{r}_{sc1}^{\,pl} \times \vec{V}_{sc1}^{\,pl} / |\vec{r}_{sc1}^{\,pl} \times \vec{V}_{sc1}^{\,pl}|$.

Usually the duration of the spacecraft inside the SOI, Δt, is assumed to be zero. This is possible because normally Δt is very small compared to the entire mission duration.

SAMURAI

Using all of the techniques introduced through the last section, a numerical analysis software application *SAMURAI* was developed in C++. As explained earlier, *SAMURAI* simulates interplanetary trajectories with different types of propulsion systems.

Capabilities

The engine types *SAMURAI* can deal with are as follows:

- VSI engine type I (variable thrust and variable I_{sp}, no limit for I_{sp})
- VSI engine type II (variable thrust and variable I_{sp}, with an upper limit for I_{sp})
- CSI engine type I (constant thrust and constant I_{sp}, continuous burn)
- CSI engine type II (constant thrust and constant I_{sp}, bang-off-bang control)
- High thrust engine (idealized instantaneous burn)

Fig. shows examples of thrust histories for these engines. A VSI engine type I can modulate its thrust and I_{sp} without limit. Without an upper limit, the I_{sp} for this type of engine may sometimes reach very high values, such as several hundred thousand seconds. This is impossible to achieve in reality because there are physical constraints on such an engine. For a VSI type II engine, users can specify an upper limit for I_{sp} in order to simulate such a constraint. For both VSI engine types, the power is fixed at its maximum level. Note that imposing an upper limit on I_{sp} is the same as imposing a lower limit

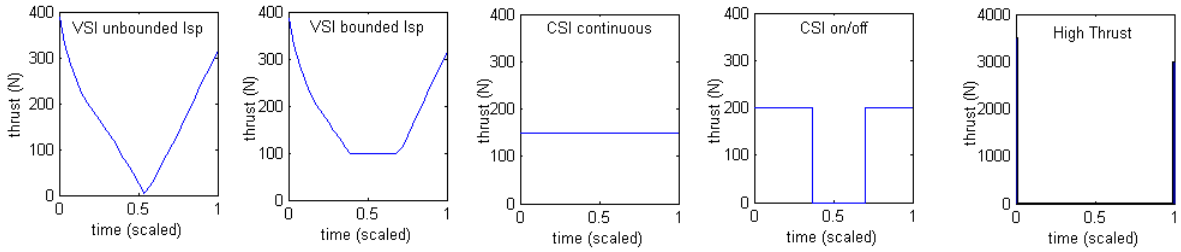

Figure 6: Examples of Thrust Histories.

on the thrust. A CSI type I engine operates with a given thrust magnitude throughout the mission, so total fuel consumption is proportional to the time of flight. *SAMURAI* calculates the minimum thrust level while satisfying the target conditions. A CSI type II engine can turn its power on and off to avoid unnecessary fuel consumption, resulting in a bang-off-bang control. Users need to input the thrust level of the engine so that the switching times will be calculated. The high thrust engine modeled in this research is a representative engine that fires for infinitesimally small amounts of time at departure and arrival. Users need to specify the value of I_{sp}.

For these different types of engines, *SAMURAI* calculates a control history (thrust direction and magnitude) that minimizes the fuel consumption for a given time of flight and given endpoint conditions (position and velocity vectors). Users can specify the endpoint conditions with the following options:

- If users would like to use the actual ephemeris data of the planets, the departure date, time of flight, ID number of the departure planet, and ID number of the arrival planet should be input. From the departure date and time of flight, planets' positions and velocities are calculated using the ephemeris data.
- If users would like to create their own planets, positions and velocities of departure and arrival points and time of flight are required. These values are directly used as the endpoint conditions.

In addition to calculating the trajectory for one value of the departure date and time of flight, *SAMURAI* has the ability to conduct a grid search with these two parameters by specifying the range of each parameter. Then *SAMURAI* finds the best launch date and time of flight for the given initial and final conditions. *SAMURAI* can also calculate swing-by trajectories with one swing-by planet.

Flow and Schemes

Fig. shows the flow chart of *SAMURAI*. First, the input data such as the number of time steps, jet power, initial mass, and upper limit for I_{sp} are read from an input file. The departure date and the arrival date are then set and the planets' positions are calculated. Once the positions of the departure and arrival planets are set, the high thrust trajectory calculation is performed using the Gauss method. With this calculation, velocities at both endpoints $V_{HT_{ini}}$ and $V_{HT_{fin}}$ are obtained. These values can be interpreted as the velocities required to travel between these two planets without any additional propulsive force. Users can input the maximum C_3's ($\sqrt{C_3} = V_\infty$) at departure and arrival to simulate the excess velocity V_∞ at each planet. The spacecraft's possible maximum velocities at two endpoints are calculated with this value and the planets' velocities (V_{pl}): $\mathbf{V}_{ini} = \mathbf{V}_{pl\_ini} + \mathbf{V}_{\infty\_ini}$ and $\mathbf{V}_{fin} = \mathbf{V}_{pl\_fin} + \mathbf{V}_{\infty\_fin}$. If the maximum $\mathbf{V}_{ini} \geq \mathbf{V}_{HT_{ini}}$ at departure and the maximum $\mathbf{V}_{fin} \geq \mathbf{V}_{HT_{fin}}$ at arrival, we do not need to calculate low thrust trajectories because the spacecraft reaches the target without any propulsive force. The results from the Gauss problem will be the answer in this case (See Fig.). If $\mathbf{V}_{ini} < \mathbf{V}_{HT_{ini}}$ or $\mathbf{V}_{fin} < \mathbf{V}_{HT_{fin}}$ or both, then the computation of a low thrust trajectory is required. With the input value of C_3 and the direction of motion of the spacecraft, $\mathbf{V}_{ini}$ and $\mathbf{V}_{fin}$ are calculated and are used as endpoint conditions for optimization.

When the trajectory does not do a swing-by maneuver, *SAMURAI* calculates the thrust history that minimizes the fuel consumption. For an optimization with a swing-by, a more complicated process is required as shown in Fig. . The first phase of the trajectory from the departure planet to the swing-by planet is calculated with a guessed value of the final velocity at the swing-by planet. At the SOI, the

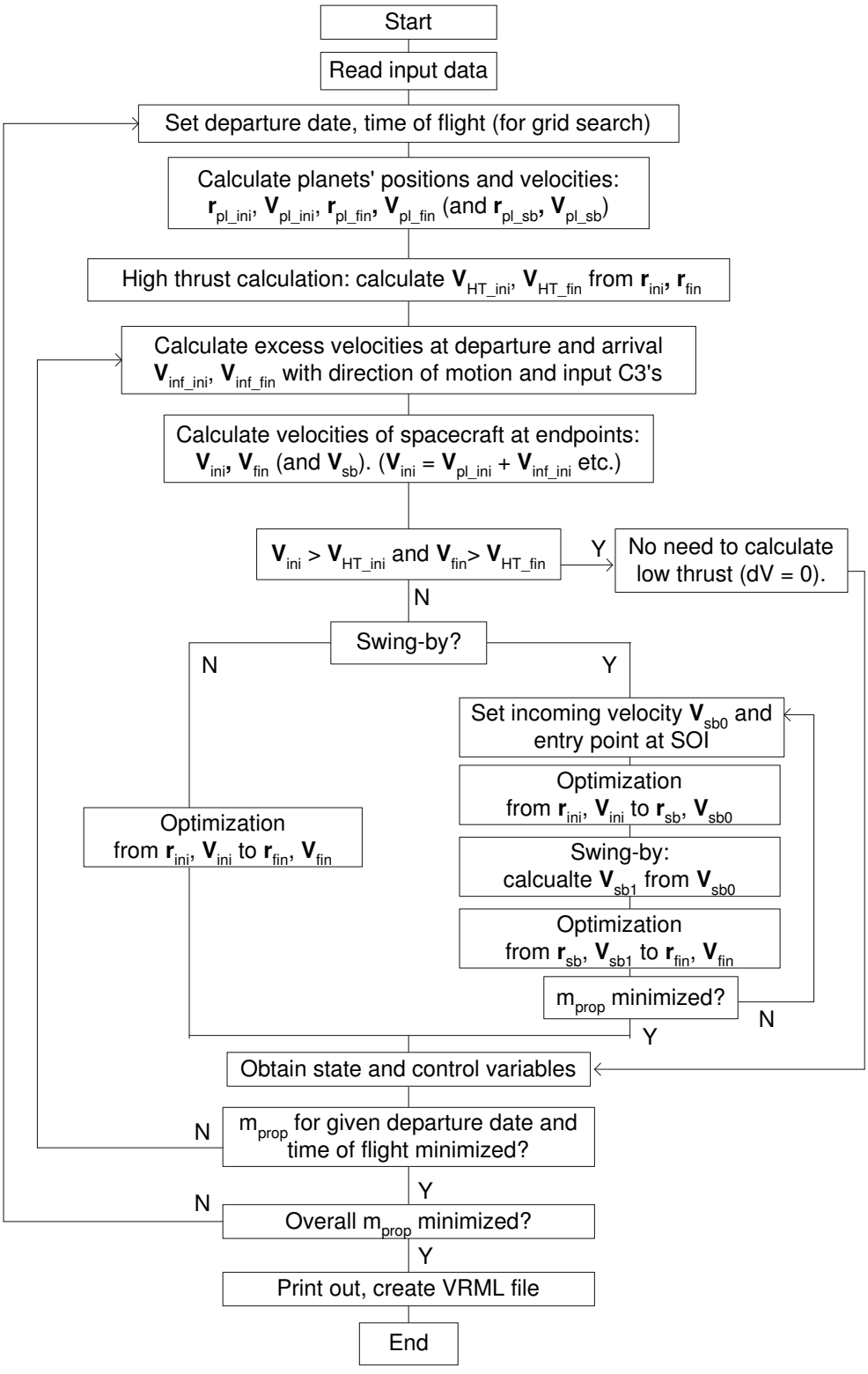

Figure 7: *SAMURAI* **Flowchart.**

swing-by planet's velocity is subtracted from the final velocity at the swing-by planet in the Heliocentric coordinates and then converted into the planetocentric incoming velocity. This incoming velocity is used as the initial velocity of the hyperbolic trajectory inside the SOI. The calculation of the trajectory inside the SOI is executed to compute the outgoing planetocentric velocity. This outgoing velocity is then converted back into the Heliocentric velocity and the swing-by planet's velocity is added. ΔV is calculated if needed and is added to the spacecraft velocity. Using this velocity as the initial velocity of the second phase, an optimization is executed from the swing-by planet to the target planet. The overall fuel consumption for the swing-by trajectory is minimized by adjusting the incoming velocity and the entry point at the swing-by planet. Therefore, an iterative process is required until the minimum-fuel trajectory is obtained.

With the above process, the minimum-fuel trajectory for a given time of flight starting with $\mathbf{V}_{ini}$ and ending with $\mathbf{V}_{fin}$ is obtained. But we may improve this trajectory by adjusting these endpoint velocities while keeping the time of flight fixed. This is done by adjusting the direction of motion of the spacecraft with respect to the planet. Therefore the above process is iterated by adjusting the direction of motion of the spacecraft at the departure and arrival planets until the fuel consumption is minimized.

When the actual ephemeris of planets is used, a grid search may be conducted to find the best launch opportunity and the time of flight that minimizes the fuel consumption over a range of departure dates and times of flight.

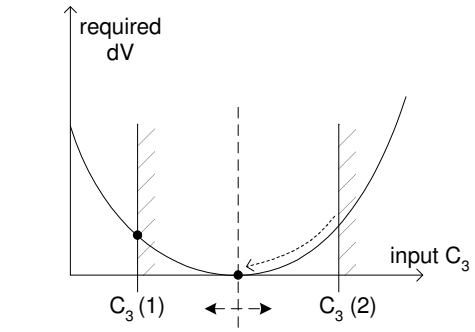

Figure 8: Input C3 and ΔV requirements.

After all of the above processes are finished, a VRML file that draws a 3D animation is created.

VSI Unconstrained I_{sp}. The computation method to obtain the histories of the control variables for an unconstrained problem is presented below. This method is called the first-order gradient algorithm[7].

1. Estimate a set of control variable histories, $u(t)$.
2. Integrate the system equations $\dot{x} = f(x, u, t)$ forward with the specified initial conditions $x(t_i)$ and the control variable histories from Step 1. Record $x(t)$, $u(t)$, and $\psi[x(t_f)]$.
3. Determine a n-vector of influence functions $p(t)$, and a $n \times q$ matrix of influence functions, $R(t)$, by backward integration of the influence equations, using $x(t_f)$ obtained in Step 2 to determine the boundary conditions.

$$\dot{p} = -\left(\frac{\partial f}{\partial x}\right)^T p - \left(\frac{\partial L}{\partial x}\right)^T;\ p_i(t_f) = \begin{cases} 0 & i = 1, \ldots, q, \\ (\partial\phi/\partial x_i)_{t=t_f} & i = q+1, \ldots, n, \end{cases} \tag{66}$$

$$\dot{R} = -\left(\frac{\partial f}{\partial x}\right)^T R;\ R_{ij}(t_f) = \begin{cases} 1, & i = j, \quad i = 1, \ldots n, \\ 0, & i \neq j, \quad j = 1, \ldots q. \end{cases} \tag{67}$$

4. Simultaneously with Step 3, compute the following integrals:

$$I_{\psi\psi} = \int_{t_i}^{t_f} R^T \frac{\partial f}{\partial u} W^{-1} \left(\frac{\partial f}{\partial u}\right)^T R\, dt \quad [(q \times q)\text{-matrix}] \tag{68}$$

$$I_{J\psi} = I_{\psi J}^T = \int_{t_i}^{t_f} \left(p^T \frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right) W^{-1} \left(\frac{\partial f}{\partial u}\right)^T R\, dt \quad [q\text{-row vector}] \tag{69}$$

$$I_{JJ} = \int_{t_i}^{t_f} \left(p^T \frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right) W^{-1} \left[\left(\frac{\partial f}{\partial u}\right)^T p + \left(\frac{\partial L}{\partial u}\right)^T\right] dt \tag{70}$$

where W is a $m\times$m positive-definite matrix and I_{JJ} is a scalar.

5. Define $\delta\psi = -\epsilon\psi[x(t_f)], 0 < \epsilon \leq 1$ such that the next nominal solution is closer to the desired values $\psi[x(t_f)] = 0$. Then determine ν from $\nu = -[I_{\psi\psi}]^{-1}(\delta\psi + I_{\psi J})$.
6. Repeat Steps 1 through 6, using an improved estimate of $u(t)$, where

$$\delta u(t) = -[W(t)]^{-1}\left[\frac{\partial L}{\partial u} + [p(t) + R(t)\nu]^T \frac{\partial f}{\partial u}\right]^T. \tag{71}$$

Stop when $\psi[x(t_f)] = 0$ and $I_{JJ} - I_{J\psi} I_{\psi\psi}^{-1} I_{\psi J} = 0$ to the desired degree of accuracy.

VSI Constrained I_{sp}. To solve constrained VSI problems, an initial guess for the Lagrange multiplier $\lambda(t_i)$ is required in order to estimate the control vector at initial time. For VSI type II calculations, the results from VSI type I are used. This is because the thrust histories are similar to each other for unconstrained arcs, and the Lagrange multipliers obtained in VSI type I calculations can be used as an initial guess for VSI type II calculations.

The following steps are taken to obtain the results for VSI type II engines:

1. Calculate a trajectory for VSI type I.
2. Obtain the Lagrange multipliers $\vec{\lambda}$ at initial time t_i from the previous step.
3. From $\vec{x}$ and $\vec{\lambda}$, calculate the control variables:

$$\vec{l} = [l_x\ l_y\ l_z]^T = \vec{\lambda_V}/\lambda_V, \qquad T = \frac{P_J \lambda_V}{m\lambda_m}. \tag{72}$$

4. Integrate $\dot{x}$ and $\dot{\lambda}$ forward from t_i to final time t_f with the control variables obtained in the Step 3:

$$\int_{t_i}^{t_f} \dot{x} dt, \qquad \int_{t_i}^{t_f} \dot{\lambda} dt \tag{73}$$

5. Check if the resulting $x(t_f)$ satisfies the terminal constraints $\psi(t_f) = 0$.
6. If not, return to Step 3 with the new values of $\lambda(t_i)$. $\lambda(t_i)$ should be chosen so that it satisfies $\psi(t_f) = 0$ AND minimizes the performance index J. Powell's method is used to estimate the next $\lambda(t_i)$.
7. Iterate until J is minimized and $\psi(t_f) = 0$ is satisfied with the desired degree of accuracy.

CSI Continuous Thrust. The CSI type I problem (continuous thrust) is not a constrained problem. The control variables are now only two (in-plane thrust angle and out-of-plane thrust angle), hence it is simpler to optimize than the VSI type I problem that has three control variables (thrust magnitude and two angles). For constant thrust problems, the optimizer cannot find a solution if the thrust magnitude is not sufficient to reach the target. For example, if a trajectory from Earth to Pluto is desired with a time of flight of 1 year, I_{sp} of 100,000 sec, and 1 kW of jet power, then the optimizer cannot find the answer because the thrust level is too low. On the other hand, a trajectory from Earth to Mars with a 300-day time of flight, 3,000 sec of I_{sp} and 50 MW of jet power is more than what is required.

SAMURAI finds the minimum thrust level while satisfying the terminal conditions using an iterative process. If the thrust level of the current iteration is not enough to reach the target, the thrust level of the next iteration is made a little bit larger than the current value. On the other hand, if starting with too much thrust, the thrust level of the next iteration is made a little bit smaller than the current iteration. The process is iterated until the minimum thrust level that satisfies the terminal conditions is found.

CSI Bang-Off-Bang. Finding the solution for CSI type II (bang-off-bang) requires finding the switching times, and the switching times are determined by the sign of the switching function. As shown previously, the switching function is a function of the Lagrange multipliers. Therefore, for CSI type II problems, an initial guess for the Lagrange multipliers is required to start the calculation. This is similar to the VSI type II case. The following steps are taken:

1. Calculate a trajectory for VSI type I.
2. Obtain the Lagrange multipliers $\vec{\lambda}$ at initial time t_i from the previous step.
3. From $\lambda(\vec{t}_0)$, calculate the control variables $u(t_0)$ at initial time:

$$u = [u_0\ u_1\ u_2\ u_3]^T = [l_x\ l_y\ l_z\ T]^T, l = [l_x\ l_y\ l_z]^T = \vec{\lambda}_V/\lambda_V \tag{74}$$

   T is determined by the sign of the switching function.

$$S = \vec{l} \cdot \vec{\lambda}_V/m = \lambda_m/c \tag{75}$$

   If S is positive, T is the prescribed value, and if S is negative, $T = 0$.
4. Integrate $\dot{x}$ and $\dot{\lambda}$ forward from t_i to final time t_f. Control variables need to be calculated as the time step proceeds. Control variables for the next step can be calculated by the equations in the previous step.
5. Check if the resulting $x(t_f)$ satisfies the terminal constraints $\psi(t_f) = 0$.
6. If not, return to step 3 with the new values of $\lambda(t_i)$. $\lambda(t_i)$ should be chosen so that it satisfies $\psi(t_f) = 0$ AND minimizes the performance index J. Powell's method is used to estimate the next $\lambda(t_i)$.
7. Iterate until J is minimized and $\psi(t_f) = 0$ is satisfied with the desired degree of accuracy.

The switching function method described above only estimates the solution, and the terminal constraints are not usually satisfactorily met. More computation is required by increasing the burn time step by step. For example, suppose that the total time step is 300 steps, and the switching function estimates switching times as the 50th and 250th steps. If the terminal condition is not satisfactory, that means more burn time is needed. The first switching time should be greater than 50, and the second switching time should be smaller than 250. Therefore, using steps 50 and 250 as initial guesses for the switching times, the burn time is increased one by one until the terminal constraints are satisfied.

Input and Output

SAMURAI calculates a transfer trajectory between two planets with or without a swing-by. As stated previously, in addition to the trajectory between the actual planets, users can make up their own planetary bodies and calculate the transfer trajectory for these planets. The mandatory input data is as follows:

- Option ID number (represents engine type)
- Planet's position and velocity are defined with either one of the following:
  - Coordinates of positions and velocities of planets in the Cartesian coordinates.
  - ID number for planets (3 for Earth, 4 for Mars, etc.)
- Departure date (yyyy/mm/dd) if actual planets are used
- Time of flight (days)
- Jet power (W)

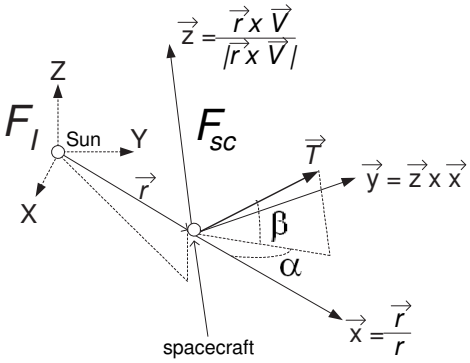

Figure 9: α (in-plane thrust angle) and β (out-of-plane thrust angle) in the Spacecraft-centered Coordinates.

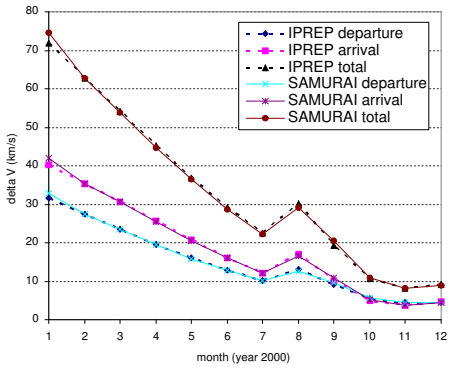

Figure 10: Results with *SAMURAI* and IPREP: ΔV Requirements for High Thrust Venus Transfer.

- Initial mass (kg) at the edge of the SOI
- Maximum allowable I_{sp} for VSI type II
- I_{sp} for CSI type II and high thrust

In addition to the above input data, users can also specify other parameters such as the maximum number of iterations and the tolerance for terminal conditions.

Output from *SAMURAI* is the following:

- History of state variables (x, y, z, u, v, w, m)
- History of control variables (thrust magnitude and direction)
- VRML file

Thrust direction is expressed by two angles in the spacecraft-centered coordinates. α is the in-plane thrust angle and β is the out-of-plane thrust angle (see Fig.). X, y, and z axes of the spacecraft-centered coordinates are defined as follows:

$$\vec{x} = \vec{r}/|\vec{r}| \quad (76)$$
$$\vec{z} = \vec{r} \times \vec{V}/|\vec{r} \times \vec{V}| \quad (77)$$
$$\vec{y} = \vec{z} \times \vec{x} \quad (78)$$

where $\vec{r}$ and $\vec{V}$ are the position and velocity vectors of spacecraft.

Validation and Verification

In order to validate *SAMURAI*, several analyses have been performed and the results with *SAMURAI* are compared to the results with other existing reliable interplanetary trajectory calculation programs. There are no applications to calculate general VSI trajectories. Therefore, validation for CSI engines and high thrust engines are performed. IPREP is used to compare the results for high thrust, and ChebyTOP is used for CSI trajectories.

Validation of High Thrust with IPREP. IPREP (Interplanetary PREProcessor) is a rapid grid-search optimizer, created by Martin Marietta Astronautics, on launch and arrival windows, minimum ΔV or mass optimization. IPREP is widely used to estimate ΔV for high thrust trajectories. To compare the results with *SAMURAI* and the results with IPREP, transfer trajectories are calculated from Earth to Venus, Mars, and Jupiter. Times of flight are set to 200 days for Venus transfer, 360 days for Mars transfer, and 500 days for Jupiter transfer. Twelve departure dates are considered: the first day of the month in the year 2000. Figs. to show the ΔV requirements (at departure and arrival, and total)

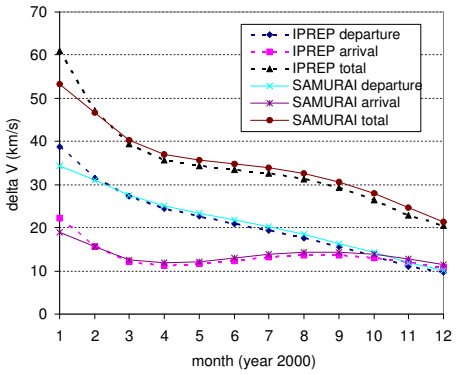

Figure 11: Results with *SAMURAI* and IPREP: ΔV Requirements for High Thrust Mars Transfer.

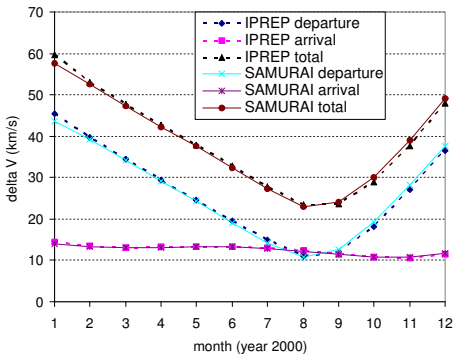

Figure 12: Results with *SAMURAI* and IPREP: ΔV Requirements for High Thrust Jupiter Transfer.

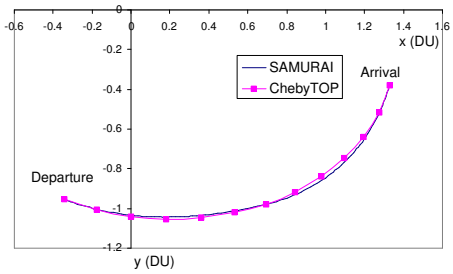

Figure 13: Results with *SAMURAI* and ChebyTOP: Trajectory Comparison for CSI type I Mars Transfer.

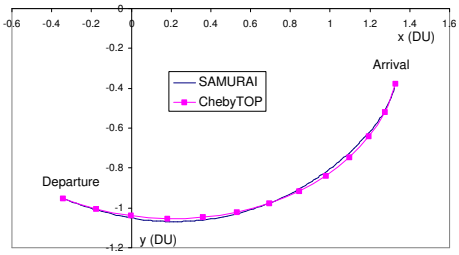

Figure 14: Results with *SAMURAI* and ChebyTOP: Trajectory Comparison for CSI type II Mars Transfer.

calculated with *SAMURAI* and IPREP. They show that the results obtained by *SAMURAI* match well with the results obtained by IPREP.

Validation of CSI with ChebyTOP. ChebyTOP (Chebyshev Trajectory Optimization Program) is an analysis tool developed by Boeing that enables the user to rapidly conduct a parametric analysis and optimization of interplanetary missions employing electrically propelled spacecraft. To compare the results from *SAMURAI* and ChebyTOP, transfer trajectories from Earth to Mars are analyzed for CSI type I and type II engines with the following assumptions: 1,000 kg initial mass, departure date of June 1, 2018, and 120-day TOF.

CSI type I The results obtained using *SAMURAI* are as follows: the maximum I_{sp} that satisfies the target conditions is 11,372 sec when the jet power is 100 kW, and the resulting final mass is 833.25 kg. When the same trajectory was calculated using ChebyTOP with an input I_{sp} of 11,372 sec, the resulting jet power requirement was 99.434 kW, and the final mass was calculated as 834.21 kg. Fig. 13 illustrates the calculated trajectory. Although ChebyTOP does not output the history of the thrust direction, this figure shows that the path obtained with *SAMURAI* is very close to that obtained with ChebyTOP.

CSI type II For a trajectory with a CSI type II engine, I_{sp} is set to 5,000 sec. For *SAMURAI*, jet power was 91.23 kW, the first switching time (t_1) was day 14.4, and the second switching time (t_2) was day 103.2. The resulting final mass was 787.5 kg. For ChebyTOP, jet power was 91.23 kW, t_1 was day 13.3 and t_2 was day 102.0. The resulting final mass was 794.9 kg. This trajectory is shown in Fig. 14. Again, the trajectories with *SAMURAI* and ChebyTOP are similar.

The validation conducted in this section shows that *SAMURAI* is precise enough to conduct general trajectory calculations for CSI engines and high thrust engines.

VRML. The Virtual Reality Modeling Language (VRML) can be used to craft three-dimensional virtual worlds on the internet. With a text file as an input file, VRML draws many types of objects as well as animations. In this research, trajectories are drawn with VRML for easy visualization. *SAMURAI* outputs a file that is used as a VRML input file. A three-dimensional trajectory is drawn on a web browser with the thrust direction vectors shown at several points along the trajectory.

It is sometimes difficult to choose the departure date or time of flight. If a user chooses a bad combination of these two values, the calculation will not converge. Because this drawing displays the positions of departure and arrival planets, it is helpful to determine when to depart and what time of flight to choose.

EARTH-MARS ROUND TRIP EXAMPLE

An example of the possible application of *SAMURAI* is a round trip from Earth to Mars. A grid search for the departure date from Earth is performed for a 10-year search range starting Jan. 1, 2010. Times of flight for both outbound and inbound legs are limited to less than or equal to 120 days. The departure date from Mars is determined so that the total fuel consumption is minimized. The jet power of the engine is assumed to be 10 MW for all engines. The initial mass at Earth is 100 MT, and the initial mass at Mars is fixed to 80 MT. Table shows the resulting departure dates, times of flight, and the fuel consumption (ΔV for high thrust). Zero C_3 is assumed at both departure and arrival for all cases. This table shows that, among VSI and CSI engines, VSI type I (unlimited Isp) requires the smallest amount of fuel. The VSI type II engine in this example has an upper limit on Isp of 30,000 sec (and therefore a lower limit on thrust at 67.99 N). The CSI type I engine is a fixed Isp engine with continuous burn, and for the outbound leg the required thrust level is 179.35 N for a 120-day transfer. For the inbound leg, TOF is 112 days if the same thrust level as the outbound leg is used. The CSI type II engine is a fixed Isp engine with the capability of switching on and off. For this example 5,000 sec of Isp is assumed when the engine is on. The engine is fired twice (at departure and arrival) for each of outbound and inbound legs.

Table 1: Minimum Fuel Round Trip from Earth to Mars for Various Thruster Types.

	Outbound (Earth → Mars)			Inbound (Mars → Earth)			Total	
	Departure date	TOF (day)	Fuel (kg)	Departure date	TOF (day)	Fuel (kg)	Time (day)	Fuel (kg)
VSI type I	06/03/2018	120	12,758	08/04/2020	120	9,437	913	22,195
VSI type II	06/03/2018	120	12,916	08/04/2020	120	9,615	913	22,531
CSI type I	06/01/2018	120	16,675	07/29/2020	112	15,589	901	32,264
CSI type II	06/01/2018	120	20,357	08/03/2020	120	17,079	914	37,436
High thrust	06/03/2018	120	(9.65)	08/05/2020	120	(10.34)	914	(19.99)

For high thrust, ΔV is shown in parenthesis.

Fig. 15 shows the resulting trajectories drawn with VRML for a VSI type II engine. Arrows along the trajectory show the direction and the magnitude of the thrust vector. Larger thrust is required near departure and arrival for both legs to accelerate the spacecraft to begin and end the transfer.

CONCLUSION

SAMURAI (Simulation and Animation Model Used for Rockets with Adjustable Isp), an interplanetary trajectory calculation application, calculates transfer trajectories with variable thrust, variable Isp (VASIMR-type) engines as well as conventional constant low thrust, constant Isp engines and high thrust engines. *SAMURAI* utilizes a calculus of variations algorithm to evaluate thrust history that

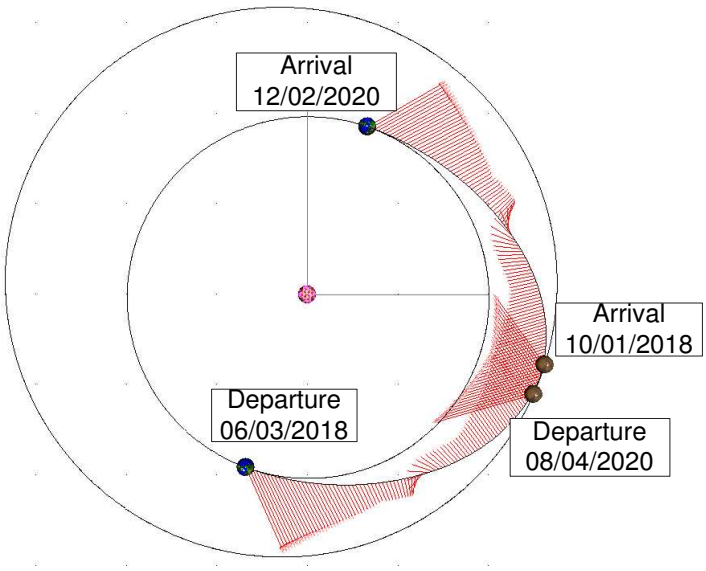

Figure 15: VRML Drawing of Round Trip from Earth to Mars with a VSI Type II Engine.

minimizes the fuel consumption from one planet to another. A trajectory with a planetary swing-by can also be calculated.

The results from *SAMURAI* have been compared with results from existing interplanetary trajectory calculation application. This validation shows that *SAMURAI* is precise enough to conduct general trajectory calculations for CSI engines and high thrust engines.

A few examples of trajectory simulation including a round trip from Earth to Mars have been analyzed. The final example showed that VSI type I is the most effective among VSI and CSI engines. Specifically, a VSI type I engine requires the least amount of fuel among VSI and CSI engines. This result should be true for any mission because the results of VSI type II and CSI type I and II are subsets of the results of VSI type I that has the more freedom than other engine types. More analyses must still be performed to further substantiate the above statement.

NOTATION

$\vec{g}$	Acceleration of gravity (g_0: at sea level)
H	Hamiltonian
J	Cost function
$\vec{l}$	Primer vector
m	Mass of the spacecraft
P_J	Jet power
$\vec{r}$	Spacecraft position vector ($= [x \quad y \quad z]^T$)
S	Switching function
$\vec{T}$	Spacecraft thrust vector ($= [T_x \quad T_y \quad T_z]^T$)
$\vec{x}$	State vector
$\vec{u}$	Control vector
$\vec{V}$	Spacecraft velocity vector ($= [u \quad v \quad w]^T$)
α	In-plane thrust angle in the spacecraft-fixed frame
β	Out-of-plane thrust angle in the spacecraft-fixed frame
$\Delta\nu$	True anomaly difference between Earth and the target planet
η	In-plane thrust angle in the inertial frame
λ_m	Lagrange multiplier for spacecraft mass
$\vec{\lambda}_r$	Lagrange multiplier vector for spacecraft position

$\vec{\lambda}_V$ Lagrange multiplier vector for spacecraft velocity
μ Gravitational constant
ϕ Turn angle of spacecraft due to swing-by
ξ Out-of-plane thrust angle in the inertial frame
ψ Terminal constraints

References

[1] Chang-Diaz, F.R., Hsu, M.M., Branden, E., Johnson, I., and Yang, T.F., "Rapid Mars Transits With Exhaust-Modulated Plasma Propulsion," *NASA Technical Paper 3539*, NASA, 1995.

[2] Kechichian, J.A, "Optimal Low-Thrust Transfer Using Variable Bounded Thrust," *Acta Astronautica*, Vol. 36, No. 7, pp. 357–365, 1995.

[3] Casalino, L., Colasurdo, G., and Pastrone, D., "Optimization of ΔV Earth-Gravity-Assist Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, pp. 991–995, 1998.

[4] Nah, R., and Vadali, S., "Fuel-Optimal, Low-Thrust, Three-Dimensional Earth-Mars Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, pp. 1100–1107, 2001.

[5] Seywald, H., Roithmayr, C., and Troutman, P.A., "Fuel-Optimal Orbital Transfer for Variable Specific Impulse Powered Spacecraft," *Advances in the Astronautical Sciences, Spaceflight Mechanics 2003*, Vol. 114, pp. 347–364, 2003.

[6] Ranieri, C.L., and Ocampo, C.A., "Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories Via an Indirect Method," *Astrodynamics 2003*, Vol. 116, AAS 03–572, pp. 1141–1160, Univelt Inc., San Diego, CA, 2003.

[7] Bryson Jr., A., and Ho,Y.C., *Applied Optimal Control*, Taylor & Francis, Bristol, PA, 1975.

[8] Craig A. Kluever, C.A. and Pierson, B.L., "Optimal Low-Thrust Three-Dimensional Earth-Moon Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 4, July – August, 1995.

[9] Hull, D.G., *Optimal Control Theory for Applications*, Mechanical Engineering Series, Springer, New York, NY, 2003.

[10] Lewis, F.L., *Optimal Control*, John Wiley & Sons, Inc., New York, NY, 1986.

[11] Sakai, T. and Olds, J.R., "Development of a Multipurpose Low Thrust Interplanetary Trajectory Calculation Code," *Astrodynamics 2003*, Vol. 116, AAS 03–667, pp. 2597–2612, Univelt Inc., San Diego, CA, 2003.

[12] Lawden, D.F., *Optimal Trajectories for Space Navigation*, Chap. 3, Butterworths, London, 1963.

[13] Vanderplaats, G.N., *Numerical Optimization Techniques for Engineering Design*, pp.98 – 102, Vanderplaats Research & Development, Inc., Colorado Springs, CO, 1998.

[14] Bate, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Chap. 5, Dover Publications, Inc., New York, NY, 1971.

[15] Labunsky, A.V., Papkov, O.V., and Sukhanov, K.G., *Multiple Gravity Assist Interplanetary Trajectories*, Chap. 1, Gordon and Breach Science Publishers, Amsterdam, The Netherlands, 1998.

[16] Hale, F.J., *Introduction to Space Flight*, p. 152, Prentice-Hall, Inc., Englewoods Cliffs, NJ, 1994.

[17] Battin, R.H., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, AIAA Education Series, Reston, VA, 1999.

[18] Escobal, P. R., *Methods of Orbit Determination*, Chap. 1, John Wiley & Sons, Inc., New York, NY, 1965.

[19] Humble, R.W., Henry, G.N., and Larson, W.J., *Space Propulsion Analysis and Design*, The MacGraw-Hill Companies, Inc., New York, NY, 1995.