A Global-Local Approach for Trajectory Generation on Rough Terrain

Chen Lin*  Puneet Singla†  Tarunraj Singh‡

In this paper, we consider trajectory generation problem as an optimal control problem while minimizing a cost function and satisfying any boundary, terrain, path and system dynamics constraints. The optimal state and control profiles are approximated by a weighted average of independent local approximations. Specially derived weight functions are used to construct a globally continuous function. Such a method provides us a way of using local information to construct near-optimal trajectories, which is useful in the context of numerically solving nonlinear trajectory generation problems.

I. Introduction

Optimal trajectory generation is an important task for mobile robot navigation particularly in rugged environment such as Mars. In robotics this is more popularly known as motion planning and several methods have been proposed to generate an optimal trajectory.1 A trajectory can be feasible only if it satisfies various conditions such as, the differential equations of motion for the system, the saturation constraints on the actuators, and the path constraints consistent with the configuration space. On rugged terrain environment such as Mars, it is very important to take into account the vehicle’s interaction with the terrain to achieve a prescribed motion.

Various algorithms can be used to plan static obstacle-free paths and these trajectories can then be tracked by feedback control. Many methods like Voronoi diagram, road map Techniques, cell decomposition, potential function method etc. have been proposed to find an obstacle free path.1–3 The roadmap and cell decomposition methods reduce the path planning problem to that of searching a graph by first analyzing the connectivity of the free space. A major drawback of these methods is that they often do not lead to a continuous path. Unlike these methods, potential field and continuous path planning methods do not include an initial processing step aimed at capturing the connectivity of free space in a concise representation. Instead, they search a much larger graph area representing the adjacency among the patches contained in robot’s configuration space. In this paper, we consider trajectory generation based on an optimal control problem formulation.

*Graduate Student, ASME member, Department of Mechanical & Aerospace Engineering, University at Buffalo, Buffalo, NY-14260, Email: cl225@buffalo.edu.
†Assistant Professor, AIAA, AAS Member, Department of Mechanical & Aerospace Engineering, University at Buffalo, Buffalo, NY-14260, Email: psingla@buffalo.edu.
‡Professor, Department of Mechanical & Aerospace Engineering, University at Buffalo, Buffalo, NY-14260, Email: tsingh@eng.buffalo.edu.
Typically, the optimal control problem for trajectory generation is posed as a two-point boundary value problem. Shooting method and gradient based iterative approaches have traditionally been used to design optimal controllers to implement the development. Several numerical methods have been proposed to pose the optimal control problem as a nonlinear programming problem. Many methods including direct collocation pseudo-spectral methods and spline approximations have been used to solve the optimal trajectory generation problem. In Ref. [16] a spline based approximation for control input vector is discussed to generate continuous vehicle trajectories while accounting for terrain shape and models of vehicle dynamics. For computational efficiency it is desired that the feasible nonlinear trajectory is approximated by a few continuous basis functions with unknown parameters. The variability of feasible control input set is often non-uniform in time and it might not be efficient to pose the same structure on control input vector at all times. In Ref. [11], we present a method to transcribe an optimal control problem to nonlinear programming problem by parameterizing a globally smooth control input function using various local independent approximations. In this paper, our main objective is to make use of this method to solve the trajectory generation problem while accounting for terrain shape and models of vehicle dynamics as well as any other actuators, path or boundary constraints.

The structure of the paper is as follows: first a model for vehicle dynamics is developed which accounts for terrain shape; followed by the formulation of the trajectory generation problem as an optimal control problem. This is followed by the discussion on the proposed method to solve the resulting optimal control problem by approximating globally smooth state and control input vector from various independent local approximations. The paper concludes with results and summary.

II. Vehicle Motion Model

In this section, we briefly discuss the vehicle dynamics model which also account for terrain profile. Let us assume that \( \mathbf{v}_B = \{v_x, v_y, v_z\}^T \) and \( \mathbf{v}_I = \{\dot{x}, \dot{y}, \dot{z}\}^T \) represent the vehicle linear velocity vectors in the body frame attached to the vehicle and an inertial world frame, respectively. The mapping between body frame velocity vector \( \mathbf{v}_B \) and inertial frame velocity vector \( \mathbf{v}_I \) is given as:

\[
\mathbf{v}_I = \mathbf{R}(\gamma, \beta, \alpha) \mathbf{v}_B
\]

(1)

where, \( \mathbf{R}(\gamma, \beta, \alpha) \) is the orientation matrix projecting a vector in a body frame to an inertial frame and can be computed from Euler angles \( (\gamma, \beta, \alpha) \) as:

\[
\mathbf{R}(\gamma, \beta, \alpha) = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma)
\]

(2)
where
\[
R_z(\alpha) = \begin{bmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad R_y(\beta) = \begin{bmatrix}
\cos \beta & 0 & \sin \beta \\
0 & 1 & 0 \\
-\sin \beta & 0 & \cos \beta
\end{bmatrix}, \quad R_x(\gamma) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \gamma & -\sin \gamma \\
0 & \sin \gamma & \cos \gamma
\end{bmatrix}
\] (3)

Similarly, the mapping between the Euler angle rates \( \dot{\theta} = \{\dot{\gamma}, \dot{\beta}, \dot{\alpha}\}^T \) and the body frame angular velocities \( \omega = \{\omega_x, \omega_y, \omega_z\}^T \) can also be found by transforming the individual Euler rotation rates from their intermediate frames to the body-fixed frame:

\[
\dot{\theta} = \begin{bmatrix}
\dot{\gamma} \\
\dot{\beta} \\
\dot{\alpha}
\end{bmatrix} = \begin{bmatrix}
1 & \tan \beta \sin \gamma & -\tan \beta \cos \gamma \\
0 & \cos \gamma & \sin \gamma \\
0 & \frac{-\sin \gamma}{\cos \beta} & \frac{\cos \gamma}{\cos \beta}
\end{bmatrix} \begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\] (4)

Eq. (4) can be used to determine the body frame angular velocity vector, \( \omega \):

\[
\omega = \begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix} = \begin{bmatrix}
\dot{\gamma} + \sin(\beta)\dot{\alpha} \\
\cos(\gamma)\dot{\beta} - \sin(\gamma)\cos(\beta)\dot{\alpha} \\
\sin(\gamma)\dot{\beta} + \cos(\gamma)\cos(\beta)\dot{\alpha}
\end{bmatrix}
\] (5)

Eqs. (1) and (4) constitute the kinematic model for vehicle equation of motion in general 3-D space. It is important to notice that for a ground vehicle, orientation angles- \( \beta \) and \( \gamma \) and elevation \( z \) are determined by the pose and the terrain shape corresponding to current vehicle location. As a consequence of this, only the position variables, \((x, y)\) and orientation angle, \( \alpha \) need to be considered for trajectory planning purposes.
Hence, we have:

\[
\begin{aligned}
\dot{x} &= (\cos \alpha \cos \beta) v_x + (\cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma) v_y + (\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma) v_z \\
\dot{y} &= (\sin \alpha \cos \beta) v_x + (\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma) v_y + (\sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma) v_z \\
\dot{\alpha} &= -\frac{\sin \gamma}{\cos \beta} \omega_y + \frac{\cos \gamma}{\cos \beta} \omega_z
\end{aligned}
\]

(6)

We assume the wheeled mobile robots move rigidly with three degrees of freedom in the local terrain tangent plane and body-frame linear velocity vector, \( \mathbf{v}_B \) is constrained along the forward \( x \)-axis of the vehicle frame. Hence, body-frame linear velocity (\( v_x \)) and angular velocity (\( \omega_z \)) are natural candidates for the controls. The direct dynamic model of the system equations with terrain function are listed as follows:

\[
\begin{aligned}
\dot{x} &= \cos(\alpha) \cos(\beta) v_x \\
\dot{y} &= \sin(\alpha) \cos(\beta) v_x \\
\dot{\alpha} &= -\frac{\sin \gamma}{\cos \beta} \omega_y + \frac{\cos \gamma}{\cos \beta} \omega_z
\end{aligned}
\]

(7)

The pitch and roll angles corresponding to vehicle’s current location can be determined from terrain profile, \( z = z(x, y) \):

\[
\begin{aligned}
\gamma(t) &= \frac{\partial z(x, y)}{\partial y}, & \beta(t) &= \frac{\partial z(x, y)}{\partial x}
\end{aligned}
\]

(8)

Notice that angular velocity component \( \omega_y \) can be computed by making use of Eq. (5):

\[
\omega_y = \cos(\gamma) \dot{\beta} - \sin(\gamma) \cos(\beta) \dot{\alpha}
\]

(9)

Furthermore, the terrain slope in the direction of the vehicle motion can be computed as:

\[
\lambda(t) = \frac{\partial z(x, y)}{\partial x} \cos \alpha(t) + \frac{\partial z(x, y)}{\partial y} \sin \alpha(t)
\]

(10)

Eq. (7) along with Eqs. (8) and (9) constitute a kinematic level model for vehicle motion on rugged terrain. For flat terrain approximation, i.e., \( \beta = \gamma = 0 \) and \( v_y = v_z = 0 \), the kinematic state model decouples the translation motion from orientation motion, the above states are simplified as:

\[
\begin{aligned}
\dot{x} &= v_x \cos \alpha \\
\dot{y} &= v_x \sin \alpha \\
\dot{\alpha} &= \omega_z
\end{aligned}
\]

(11)
In Ref. [16], it is shown that the model error introduced due to the flat terrain approximation significantly effects the vehicle trajectory. Since terrain profile is usually known with a good accuracy, it makes sense to consider its effect during trajectory generation. Generally, addition of rough terrain to the problem compiles matters by coupling nonlinear equations of motion for vehicle dynamics.

### III. Main Idea: Optimal Control Problem Formulation

A general optimal trajectory problem can be stated as: given a model of system dynamics and the constraints on state and control variables, compute the appropriate control trajectory that will drive the system to the desired state position from a given initial position while minimizing a performance index:

$$
\min_{u(t)} J = \int_{t_0}^{t_f} g(s(t), u(t)) \, dt
$$

subject to

- **Boundary Constraints**: $s(0) = s_0, s(t_f) = s_f$
- **System Dynamics**: $\dot{s}(t) = f(s(t), u(t))$
- **Path Constraints**: $N(s(t)) \geq N_0$
- **Control Constraints**: $u_l \leq u(t) \leq u_u, \forall t$

where $s \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ represent system state and control vectors. The cost function here is the time integral of a potentially time-varying function $g(s(t), u(t))$ along an unknown path. The function $g(s(t), u(t))$ may include instantaneous energy consumption, distance to an obstacle, final time, irregularity of the terrain profile, smoothness of the path, etc.

#### A. Nonlinear Trajectory Generation Using Global-Local Approximation

To solve the optimal control problem of Eqs. (12)-(16), we assume that both vehicle state vector $s(t)$ and control input vector $u(t)$ are unknown and parameterize them as follows:

$$
\begin{align*}
    s(t) &= \sum_{i=1}^{n_1} \omega_s(\tau_i) g_i(t, a_{s_i}), \\
    u(t) &= \sum_{j=1}^{n_2} \omega_u(\tau_j) h_j(t, a_{u_j})
\end{align*}
$$

where $g_i(t)$ and $h_j(t)$ are continuous local functions valid over interval $(t_i, t_{i+1})$. $\omega_s(\tau_i)$ and $\omega_u(\tau_j)$ are special function used to blend two adjacent preliminary local approximations $g_i(t, a_{s_i})$ and $g_{i+1}(t, a_{s_{i+1}})$ or $h_j(t, a_{u_j})$ and $h_{j+1}(t, a_{u_{j+1}})$. The preliminary approximation $g_i$ and $h_j$ are completely arbitrary, as long as they are
smooth and represents the local behavior of \( s(t) \) and \( u(t) \) well. Furthermore, we use linear combination of basis function to represent these local approximations in their region of validity, i.e., \((t_i, t_{i+1})\).

\[
\mathbf{g}_i(t, a_{s_i}) = \phi_i^T(t)a_{s_i}, \quad \mathbf{h}_j(t, a_{u_j}) = \psi_j^T(t)a_{u_j}
\]

where \( \phi_i(t) \in \mathbb{R}^{N_1}, \psi_j(t) \in \mathbb{R}^{N_2} \) are the prescribed local basis functions and \( a_{s_i}, a_{u_j} \) are the corresponding unknown coefficients. So, Eq. (17) can be written as:

\[
s_i = \sum_{i=1}^{n_1} \omega_s(\tau_i) \phi_i^T(t)a_{s_i}, \quad u_j = \sum_{j=1}^{n_2} \omega_u(\tau_j) \psi_i^T(t)a_{u_j}
\]

where

\[
\Phi(t) = \{\phi_1(t), \cdots, \phi_{n_1}(t)\}^T, \quad \Psi(t) = \{\psi_1(t), \cdots, \psi_{n_2}(t)\}^T
\]

\[
\bar{a}_s = \{\bar{a}_{s_1}, \bar{a}_{s_2}, \cdots, \bar{a}_{s_{n_1}}\}^T, \quad \bar{a}_u = \{\bar{a}_{u_1}, \bar{a}_{u_2}, \cdots, \bar{a}_{u_{n_2}}\}^T
\]

\( n_1 \) and \( n_2 \) are the number of local approximations. The basis functions, \( \phi_i(t) \) and \( \psi_j(t) \), may be chosen from the space of polynomials, or they may be spacial functions, based on prior knowledge about the problem. For example, this may be eigen-functions of linear problem describing the local phenomenon. This is one of the greatest advantages of the proposed approach because it allows the use of local functions of different form and number in the individual sub-domains. While such freedom provides great flexibility, it generally prevents the basis functions from constituting a conformal space, i.e., the inter-element continuity of the approximation is not ensured. This task is accomplished by the special functions, \( \omega_s(\cdot) \), which merge together the various local approximations. A key question is: \textit{Is there a choice of weight function that will guarantee piecewise global continuity while leaving the freedom to choose any desired local functions?}

In Refs. [17–19], it is shown that if the weighting function of Eq. (17) satisfies the following boundary value problem of Eq. (16), then the weighted average approximation in Eq. (17) form a \( m^{th} \)-order continuous globally valid model with complete freedom in the choice of the local approximations.

\[
\begin{cases}
\omega(0) = 1, \quad \omega(1) = \omega(-1) = 0, \\
\omega(\tau) : \omega^k(0) = 0, \quad \omega^k(1) = 0, \quad k = 0, \ldots, m \\
\omega(\tau) + \omega(\tau - 1) = 1, \quad \forall \tau, \quad -1 \leq \tau \leq 1
\end{cases}
\]

where \( \omega^k \triangleq \frac{d^k \omega}{d\tau^k} \). Notice that the first two aforementioned conditions guarantee the continuity of the blended approximation, whereas the third condition guarantees the blended approximation is un-biased, i.e. satisfies
partition of unity. Assuming the weighting function to be polynomial in nature, and adopting the procedure listed in Ref. [19], the generic expression for $m^{th}$ order continuity weight function can be written as:

$$\omega(\tau) = 1 - \eta^{m+1} \left\{ \frac{(2m+1)!(-1)^m}{(m!)^2} \sum_{k=0}^{m} \frac{(-1)^k}{2m-k+1} \binom{m}{k} \eta^{m-k} \right\}, \quad \eta \equiv |\tau| \quad (23)$$

Fig. 2 shows the first few order weight functions with $\tau \in (-1, 1)$. Notice that the domain of weight function $\omega$, is a compact space and $\omega(\tau) > 0, \forall \tau \in (-1, 1)$. Observe that by choosing the weighting function given by Eq. (23), we are guaranteed global piecewise continuity for all possible continuous local approximations while retaining the freedom to vary the degree of the local approximation as needed. Thus, one can choose local models $g_i(t)$ based solely on local approximability, and rely upon $\omega(\tau)$ to enforce continuity across knot points, $t_i$.

This approach is illustrated in the following example.

**Example** Let us consider the dynamics of the Vanderpol oscillator over a time window $t$ in $[0 \ 6]$:

$$\begin{align*}
x'_1 &= x_2 \\
x'_2 &= -x_1 + (1 - x_1^2)x_2 + u
\end{align*}$$

Assuming that the system is driven by the initial conditions $x_1(0) = 1$ and $x_2(0) = 0$, let us consider the global approximation of the state trajectory $x_1(t)$ by local approximation at the nodes $t_0 = 0$, $t_1 = 2$, $t_2 = 4$ and $t_3 = 6$. Assume the external input $u = 0$, to calculate the nominal trajectories about which the jacobian $H_i$ are calculated at each of the nodes.
This matrix gives information about the local system dynamics around each of the nodes. Starting with the value of the state trajectories $x(t)$ at the nodes as the initial conditions, and the matrix exponentials as the basis functions, the local approximation of $x_1(t)$ for $i = 0, 1, 2$ are plotted in Fig. 3(a):

$$g_i(t) = \phi_i(t) x(t_i), \quad \phi_i(t) = e^{H_i(t-t_i)}$$

\[
H_i = \begin{pmatrix}
0 & 1 \\
-1 - 2x_1(t_i)x_2(t_i) & 1 - x_1^2(t_i)
\end{pmatrix}
\]

From Fig. 3(a), it is clear that although the local approximations capture the system dynamics locally,
but they do not connect to each other at the neighboring nodes. This brings us to our key question related to merging these local independent approximations while ensuring a desired order of global continuity. We assume that the second order global continuity is desired for blending purpose. The global approximation of $x_1(t)$ blending the local approximations at the nodes is plotted in Fig. 3(b).

It is to be noted that as the number of nodes increase, the approximation gets better. This can be seen from the Fig. 3(c), where 13 nodes are considered. This technique provides us a powerful way of blending arbitrary local approximations to obtain a desired order globally continuous function.

In Ref. [11], a general method is described to solve the nonlinear trajectory generation problem using arbitrary basis function for local approximations. So, the optimal trajectories are obtained iteratively by assuming an initial control profile and solving for the corresponding state trajectory evolution using the Global-Local approximation technique, based on which the control profile is updated for the next iteration step. Notice that the basis functions convey information about the local system dynamics and the dynamics of the system is an equality path constraint that needs to be satisfied at each time step, which becomes an equality constraint on the unknown coefficient vector $a = \{a_s, a_u\}^T$. The resulting nonlinear programming problem in unknown coefficient vector $a$ which can be solved using any commercially available software such as SNOPT$^{20}$ or NPSOL.$^{21}$

IV. Trajectory Generation for Rugged Terrain

In this section, we discuss in detail the application of Global-Local approximation method to generate 3D terrain optimal motion trajectory.

Introducing the vehicle state vector $s(t) = \{x, y, \alpha\}^T$ and control input vector $u(t) = \{v_x, \omega_z\}^T$, we define the optimal trajectory generation problem as follows:

$$
\min_{u(t)} J = \int_0^{t_f} g(s(t), u(t)) \, dt
$$

subject to

$$
\dot{s}(t) = f(s(t), u(t)) = \left\{ u_1 \cos s_3 \cos \beta, \ u_1 \sin s_3 \cos \beta, \ \frac{\sin \gamma}{\cos \beta} \omega_y + \frac{\cos \gamma}{\cos \beta} \omega_z \right\}^T
$$

$$
h(s(t), \beta(t), \gamma(t)) = 0, \ \forall t
$$

$$
h(s(t), \beta(t), \gamma(t)) \leq 0, \ \forall t
$$

$$
s(0) = s_0, \ s(t_f) = s_f
$$

$$
u_l \leq u(t) \leq u_u, \ \forall t
$$

American Institute of Aeronautics and Astronautics
where,

\[
\begin{align*}
\mathbf{h}(\mathbf{s}(t), \beta(t), \gamma(t)) = & \begin{cases} 
\gamma(t) - \frac{\partial z(x,y)}{\partial y} \\
\beta(t) - \frac{\partial z(x,y)}{\partial x} \\
\omega_y - \cos(\gamma)\dot{\beta} + \sin(\gamma)\cos(\beta)\dot{s}_3 \\
(\dot{s}_1 \cos s_3 + \dot{s}_2 \sin s_3) \sin \beta + \dot{z} \cos \beta
\end{cases} \\
\lambda(\beta, \gamma, \alpha) = & \beta(t) \cos s_3(t) + \gamma(t) \sin s_3(t)
\end{align*}
\]

(31)

We mention that differential and algebraic equality (DAE) constraints of Eqs. (26) and (27) represent constraints due to vehicle motion on rough terrain whereas inequality constraint of Eq. (28) represents the constraint on vehicle mobility due to the terrain profile. Eqs. (29) and (30) represent the constraints on initial and final value of vehicle state and control effort, respectively.

As discussed in the previous section, we approximate vehicle position and orientation parameters as a weighted average of many local approximations:

\[
\begin{align*}
\mathbf{x}(t) &= \sum_{i=1}^{n_1} \omega_i(t)f_{x_i}(t, \mathbf{a}_{x_i}), \quad \mathbf{y}(t) = \sum_{i=1}^{n_2} \omega_i(t)f_{y_i}(t, \mathbf{a}_{y_i}), \quad \mathbf{\alpha}(t) = \sum_{i=1}^{n_3} \omega_i(t)f_{\alpha_i}(t, \mathbf{a}_{\alpha_i}) \\
\gamma(t) &= \sum_{i=1}^{n_4} \omega_i(t)f_{\gamma_i}(t, \mathbf{a}_{\gamma_i}), \quad \beta(t) = \sum_{i=1}^{n_5} \omega_i(t)f_{\beta_i}(t, \mathbf{a}_{\beta_i})
\end{align*}
\]

(33)

(34)

Furthermore, we use linear combination of basis functions to represent these local approximations in their region of validity, i.e., \((t_i; t_{i+1})\):

\[
\begin{align*}
f_{x_i}(t, \mathbf{a}_{x_i}) &= \phi_i^T(t)\mathbf{a}_{x_i}, \quad f_{y_i}(t, \mathbf{a}_{y_i}) = \phi_i^T(t)\mathbf{a}_{y_i}, \quad f_{\alpha_i}(t, \mathbf{a}_{\alpha_i}) = \phi_i^T(t)\mathbf{a}_{\alpha_i} \\
f_{\gamma_i}(t, \mathbf{a}_{\gamma_i}) &= \phi_i^T(t)\mathbf{a}_{\gamma_i}, \quad f_{\beta_i}(t, \mathbf{a}_{\beta_i}) = \phi_i^T(t)\mathbf{a}_{\beta_i}
\end{align*}
\]

(35)

(36)

Now, the time derivative of Eqs. (33) and (34) leads to:

\[
\begin{align*}
\dot{\mathbf{x}}(t) &= \sum_{i=1}^{n_1} \dot{\omega}_i(t)f_{x_i}(t, \mathbf{a}_{x_i}) + \sum_{i=1}^{n_1} \dot{\omega}_i \dot{f}_{x_i}(t, \mathbf{a}_{x_i}), \quad \dot{\mathbf{y}}(t) = \sum_{i=1}^{n_2} \dot{\omega}_i(t)f_{y_i}(t, \mathbf{a}_{y_i}) + \sum_{i=1}^{n_2} \dot{\omega}_i \dot{f}_{y_i}(t, \mathbf{a}_{x_i}) \\
\dot{\mathbf{\alpha}}(t) &= \sum_{i=1}^{n_3} \dot{\omega}_i(t)f_{\alpha_i}(t, \mathbf{a}_{\alpha_i}) + \sum_{i=1}^{n_3} \dot{\omega}_i \dot{f}_{\alpha_i}(t, \mathbf{a}_{\alpha_i}), \quad \dot{\gamma}(t) = \sum_{i=1}^{n_4} \dot{\omega}_i(t)f_{\gamma_i}(t, \mathbf{a}_{\gamma_i}) + \sum_{i=1}^{n_4} \dot{\omega}_i \dot{f}_{\gamma_i}(t, \mathbf{a}_{\gamma_i}) \\
\dot{\beta}(t) &= \sum_{i=1}^{n_5} \dot{\omega}_i(t)f_{\beta_i}(t, \mathbf{a}_{\beta_i}) + \sum_{i=1}^{n_5} \dot{\omega}_i \dot{f}_{\beta_i}(t, \mathbf{a}_{\beta_i})
\end{align*}
\]

(37)

(38)

(39)

Notice that the vehicle input vector \(\mathbf{u} = (v_x, \omega_z)^T\) can be computed by making use of the following expres-
\[ v_x = (\dot{x} \cos \alpha + \dot{y} \sin \alpha) \cos \beta - \dot{z} \sin \beta \] (40)

\[ \omega_z = \dot{\beta} \sin \gamma + \dot{\alpha} \cos \gamma \cos \beta \] (41)

where, \( \dot{z} \) can be computed from the knowledge of terrain profile, \( z = f(x, y) \):

\[ \dot{z}(t) = \frac{\partial z}{\partial x} \dot{x} + \frac{\partial z}{\partial y} \dot{y} \] (42)

Now, the substitution of Eqs. (33), (34), and (37)–(41) in differential constraint equations of Eq. (26) leads to the following algebraic constraint equations:

\[ \dot{s}(t) = f(s(t), u(t)) = \begin{bmatrix} u_1 \cos s_3 \cos \beta, & u_1 \sin s_3 \cos \beta, & -\sin \gamma \cos \beta \omega_y + \cos \gamma \cos \beta \omega_z \end{bmatrix}^T \Rightarrow F(t, a_x, a_y, a_\alpha, a_\beta, a_\gamma) = 0 \] (43)

Similarly, the substitution of Eqs. (33), (34), and (37)–(41) in constraint equations of Eqs. (27)–(30) leads to the following set of algebraic constraint equations:

\[ \mathcal{H}(t, a_x, a_y, a_\alpha, a_\beta, a_\gamma) = 0, \ \forall t \] (44)

\[ -\lambda_0 \leq \Lambda(t, a_\alpha, a_\beta, a_\gamma) \leq \lambda_0, \ \forall t \] (45)

\[ A \begin{bmatrix} a_x \\ a_y \\ a_\alpha \end{bmatrix} = s_0, \quad B \begin{bmatrix} a_x \\ a_y \\ a_\alpha \end{bmatrix} = s_f \] (46)

\[ u_l \leq U(t, a_x, a_y, a_\alpha, a_\beta, a_\gamma) \leq u_u, \ \forall t \] (47)

Finally, the optimal control formulation of Eqs. (25)–(30) can be transcribed into the following nonlinear programming problem:

\[ \min_{a_x, a_y, a_\alpha, a_\beta, a_\gamma} J = \int_0^{t_f} \mathcal{G}(t, a_x, a_y, a_\alpha, a_\beta, a_\gamma) \, dt \] (48)

subject to algebraic constraints defined by Eqs. (43)–(47).

Furthermore, the inequality state constraint of Eq. (45) can also be represented as an integrated state by introducing a penalty function which will push the vehicle away from the constraint boundary. It is desired that such a penalty function have a high value when the constraints are violated and must be zero when the constraints are satisfied. In other words, a viable penalty function must have a compact support. We use
the weight functions of Eq. (23) for $m = 2$ as a viable penalty function since they have a compact support.

$$\mathcal{P}(t) = 1 - \eta^2(3 - 2\eta), \eta = \left| \frac{\lambda_0}{\lambda(t)} \right|$$  \hspace{1cm} (49)

Notice that $\eta(t) \in [-1, 1]$ and penalty function $\mathcal{P}(t) \neq 0$ only when $\lambda(t) \geq \lambda_0$, i.e., constraint is violated. So, now the inequality constraint of Eq. (45) can be replaced by the following equality constraint:

$$\int_0^{t_f} \mathcal{P}(t)dt = 0$$  \hspace{1cm} (50)

Finally, notice that at any given time only two local approximations constitute the global approximation of the optimal trajectory. As a consequence of this, one can always choose the local approximations associated with $t = 0$ and $t = t_f$ nodes in such a way that boundary conditions of Eq. (29) are always satisfied. Moreover, this also provides us the freedom to incorporate any prior information that we may have about the optimal solution or vehicle dynamics.

V. Numerical Simulation Results

In this section, we present the numerical simulation results to show the effectiveness of the proposed ideas for trajectory optimization. For this purpose, we consider the problem of optimal power trajectory generation, i.e., $g((s), u) = v_x^2$. We consider four test cases with different terrain profiles and for each of the test case, we use the following constraints on vehicle mobility:

$$-1 \leq v_x \leq 1, \quad -50^\circ/s \leq \omega_z \leq 50^\circ/s, \quad -30^\circ \leq \lambda(t) \leq 30^\circ$$  \hspace{1cm} (51)

Furthermore, we use optimization toolbox SNOPT to solve the resulting nonlinear programming problem.

**Case One:** For the first test case, we assume the following values for vehicle state and terrain profile:

$$x_0 = -3, \quad y_0 = -3, \quad \gamma_0 = \gamma_f = 0, \quad \beta_0 = \beta_f = 0, \quad \alpha_0 = \alpha_f = \frac{\pi}{2}, \quad x_f = 3, \quad y_f = 3$$

$$z(x, y) = xe^{(-x^2-y^2)}$$

Using the procedure described in the previous section, nonlinear programming problem of Eq. (48) was solved using MATLAB optimization toolbox SNOPT. The number of local approximations were varied from 3 − 5 and polynomial basis functions up to degree 3 were used to describe each local approximations. In the notation of tables and figures, $M_d$ represents the order of polynomial and $N$ is the number of local
Suboptimal Cost by the SNOPT Optimization package

<table>
<thead>
<tr>
<th>Intervals</th>
<th>Suboptimal Cost</th>
<th>Number of Optimization variables for each state</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>8.8300</td>
<td>12</td>
</tr>
<tr>
<td>100</td>
<td>8.8067</td>
<td>16</td>
</tr>
<tr>
<td>100</td>
<td>8.7782</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 1. Case 1: Suboptimal Cost vs. Number of Local Approximations.

polynomial approximation. Fig. 4 shows the various plots corresponding to test case 1. Fig. 4(a) shows an initial guess of trajectory connecting from the start position to the end position. Fig. 4(b) shows an overhead view of final optimal trajectory in the x, y plane while Fig. 4(c) shows a perspective view of the optimal trajectory. Fig. 4(d) shows the time profile for $v_x$, $\omega_z$ and $\lambda(t)$ with their prescribed bounds. From these figures, it is clear that all the constraint criterions were satisfied. Table 1 lists the suboptimal cost for different number of local approximations. As expected the total cost decreases with the increase in number of local approximation which brings us closer to the optimal solution.

![Figure 4. Simulation Results for Case One](image-url)
Case Two: For the second test case, we assume the following values for vehicle state and terrain profile:

\[ x_0 = -1, \; y_0 = -1, \; \gamma_0 = \gamma_f = 0, \; \beta_0 = \beta_f = 0, \; \alpha_0 = \alpha_f = \frac{\pi}{2}, \; x_f = 1, \; y_f = 6 \]

\[ z(x, y) = 1 - x^2 + 0.2 \cos(2\pi x) \]

Once again, the number of local approximations were varied from 3 – 5 and polynomial basis functions up to degree 3 were used to describe each local approximations. Fig. 5 shows the various plots corresponding to test case 2. Fig. 5(a) shows an initial guess of trajectory connecting from the start position to the end position. Fig. 5(b) shows an overhead view of final optimal trajectory in the \( x, y \) plane while Fig. 5(c) shows a perspective view of the optimal trajectory. Fig. 5(d) shows the time profile for \( v_x, \omega_z \) and \( \lambda(t) \) with their prescribed bounds. From these figures, it is clear that all the constraints were satisfied. Table 2 lists the suboptimal cost for different number of local approximations. As expected the total cost decreases with the increase in number of local approximation which brings us closer to the optimal solution.

<table>
<thead>
<tr>
<th>Intervals</th>
<th>Suboptimal Cost</th>
<th>Number of Optimization variables for each state</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>6.4377</td>
<td>12</td>
</tr>
<tr>
<td>60</td>
<td>6.3873</td>
<td>16</td>
</tr>
<tr>
<td>60</td>
<td>6.2523</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2. Case 2: Suboptimal Cost vs. Number of Local Approximations.

Case Three: The third simulation is run with the initial point at \( [x_0 = -4; y_0 = -4; \gamma_0 = 0; \beta_0 = 0; \alpha_0 = \frac{\pi}{2}] \), and end point at \( [x_f = 4; y_f = 4; \gamma_f = 0; \beta_f = 0; \alpha_f = \frac{\pi}{2}] \) and the following terrain profile:

\[ z(x, y) = \frac{1}{2}(1 - x)^2e^{-(x^2-(y+1)^2)} - 2(\frac{x}{5} - x^3 - y^5)e^{-(x^2-y^2)} - \frac{1}{3}e^{-(x+1)^2-y^2} \]  \hspace{1cm} (52)

Notice that many different sizes of hills create a cluttered environment for the vehicle to move. Once again, the number of local approximations were varied from 3 – 5 and polynomial basis functions up to degree 3 were used to describe each local approximations. Fig. 6 shows the various plots corresponding to test case 3. Fig. 6(a) shows an initial guess of trajectory connecting from the start position to the end position. Fig. 6(b) shows an overhead view of final optimal trajectory in the \( x, y \) plane while Fig. 6(c) shows a perspective view of the optimal trajectory. Fig. 6(d) shows the time profile for \( v_x, \omega_z \) and \( \lambda(t) \) with their prescribed bounds. From these figures, it is clear that all the constraint criterions were satisfied and the proposed approach was competent enough to find the near-optimal path though the challenging environment. Table 3 lists the suboptimal cost for different number of local approximations. As expected the total cost decreases with the increase in number of local approximation which brings us closer to the optimal solution.
Figure 5. Simulation Results for Case 2.

<table>
<thead>
<tr>
<th>Intervals</th>
<th>Suboptimal Cost</th>
<th>Number of Optimization variables for each state</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>9.4519</td>
<td>12</td>
</tr>
<tr>
<td>100</td>
<td>8.9450</td>
<td>16</td>
</tr>
<tr>
<td>100</td>
<td>8.4384</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 3. Case 3: Suboptimal Cost vs. Number of Local Approximations.
Figure 6. Simulation Results for Case 3
VI. Conclusion

In this paper, a global-local approach for optimal trajectory approximation is presented while taking into account the constraints due to vehicle dynamics, its interaction with terrain, boundary constraints on vehicle state vector and actuator constraints. The main idea is to approximate the vehicle state as well as the control input vector by a weighted average of many independent local approximations. These local approximations have a compact support and can be defined \textit{independently to each other} based upon any prior information. Thus, it is amenable for parameterizing trajectory space for numerical solution of optimal control problems with high flexibility. We demonstrated the method by considering the trajectory optimization problem for three different terrain profiles.

References


17P. Singla, “Multi-resolution methods for high fidelity modeling and control allocation in large-scale dynamical systems,” *Ph. d dissertation, Texas A&M University, College Station, TX*, 2006.


