TECHNIQUES FOR TERRAIN FOLLOWING OF AUTONOMOUS VEHICLE

A. Rehman, **A. Shahzad**, **W. A. Kamal** [1] and **R. Samar**

National Engineering and Scientific Commission, Pakistan
Email: waseem_kamal@hotmail.com

Abstract: In this paper two approaches for terrain following of an autonomous vehicle are presented. Both approaches assume that a two dimensional path is available in advance that has been calculated by using a suitable terrain avoidance technique. Range patches are used to adjust height above terrain leaving a set clearance. First technique uses a repeated process of removing and readjusting the node points according to practical vehicle constraints that finally gives a set of discrete points in an optimum manner. Second uses optimally computed cubic splines that parameterize the altitude to provide a very smooth reference path for autonomous vehicle to follow. The optimal path lies as close as possible to terrain, generated by solving a nonlinear problem that minimizes the deviation from the set clearance of terrain satisfying constraints.

Keywords: Autonomous Vehicle, Nonlinear Programming, Optimal Trajectory, Digital Elevation Map, Terrain Following.

1. INTRODUCTION

The fundamental idea in terrain following (TF) is to compute a reference path for an autonomous vehicle such that it follows terrain as close as possible leaving a minimum clearance margin, providing the tracking controller with feasible reference trajectories. These trajectories are required to comply with constraints inherent to the system or externally imposed such as system dynamics, path constraints, actuator constraints and end-point conditions. Trajectories that do not comply with a system's dynamics and constraints have a small likelihood of implementation, since they might place demands on the controller beyond its limitations. Global trajectory planning requires all information before any motion of a vehicle is performed. When global information is not known in advance or it is neither perfect nor predictable, then there is a tendency to design so called local trajectory planners. Although such planners lead to the loss of path optimality but their actions are still focused on target reaching while avoiding terrain. TF problem is also addressed by (Lu and Pierson, 1995) using full point-mass dynamic. In this paper two global terrain following techniques are presented namely Stair Algorithm (SA) and Cubic Spline Algorithm (CSA). In actual practice terrain data is obtained from Digital Elevation Map (DEM) but in this paper all experiments are performed using Markov Model of terrain as described in (Kuchar, 2001). SA can be used for both off-line and on-line modes for reference path generation. It can be used on-line mode because of its computational efficiency and easy implementation provided a forward looking sensor is used. While the (CSA) accepts more constraints and can be used for off-line planning. SA is relatively simple than CSA and is based on the ascend or descend start decisions while flying at some fixed altitude of current leg to reach at a desired altitude of next leg. These legs are formed by dividing horizontal range into small patches and initially

[1] Author for correspondence.

the altitude above terrain corresponding to a leg is adjusted at constant level that include ground clearance. It uses a repeated process of removing and readjusting the node points according to practical vehicle constraints that finally gives a set of discrete points in an optimum manner. The objective of this work is

- A fast and computationally efficient method.
- Easy to implement in a simulation

CSA has the potential of incorporating further constraints of curvature, kink and level flight in some area. Cubic spline algorithm parameterizes the altitude to provide a very smooth reference path for vehicle to follow. The cubic splines are optimally computed to lie as closed as possible to a terrain by solving a nonlinear programming problem that minimizes the deviation from the set clearance of the terrain but satisfying all the practical constraints. The issue of generating smooth trajectory by using cubic splines is also addressed by (Funk, 1977) where the constraints are only enforced to satisfy at the node points that correspond to the optimization parameters. So regions between these nodes are unconstrained which may create safety problems for the vehicle. This safety issue can be improved by increasing the optimization parameters but then computational time also increases. The trajectory generated by using our approach considers both these issues and produces a safe, computationally efficient trajectory using a small number of optimizing parameters. In this way a series of range segments, each containing a cubic polynomial that gives altitude as a function of range, is obtained. This approach has following advantages:

- Trajectory is smooth enough and therefore an actual autonomous vehicle can follow it precisely.
- All constraints are well satisfied.
- Full terrain masking over all portions of the terrain.

The Paper is organised as follows: Section 2 is concerned with problem formulation, including performance measure and constraints. A brief description of Markov Model of terrain generation (Kuchar, 2001) is given in Section 3 where the terrain data for a TF problem is generated and used in simulation experiments. Section 4 describes two TF algorithms in details while simulation results and comparison are given in Section 5. Conclusions are given in Section 6.

2. PROBLEM DEFINITION AND FORMULATION

The terrain following problem that is stated generally in the previous section will now be defined in more detail. The trajectory generation process consists of two phases namely take-off phase and low-altitude phase. In take-off phase, trajectory is designed using nonlinear programming techniques with nonlinear simulation that optimizes some performance index. In low-altidue phase, trajectory generation process consists of two stages. Terrain avoidance is the first stage which sets trajectory in lateral direction that avoids obstacles. Once trajectory has been set in lateral direction, the second stage is the terrain following. In this stage, trajectory is designed in vertical direction using terrain data and vehicle constraints. This paper concentrate on this stage and assumes that input data is available in range and height. Different performance measures in different situation can be selected. Here the performance measure is total error over the entire range and is defined as

$$J = \int_{R_0}^{R_N} e^2 dR \tag{1}$$

where R_0 and R_N are first and last points of discrete range data and e is error i.e difference of actual and reference altitudes that is, $e = h - T - C_{min}$ and h is actual height , T is terrain altitude and C_{min} is minimum ground clearance. The constraints of the problem are on error e and on first, second and third derivatives of height with respect to range. They are error e, slope s, curvature k and kink p.

Speed of UAV during low-altitude phase is assumed constant and so the rate of ascend and descend is expressed by $s = \frac{dh}{dR}$ or $\frac{\frac{dh}{dt}}{V}$. Therefore the slope constraint is

$$s_{min} \leq s \leq s_{max} \tag{2}$$

Curvature or the normal acceleration can be expressed by $k = \frac{ds}{dR}$ or $k = \frac{\frac{d^2h}{dt^2}}{V^2}$ and is bounded as

$$k_{min} \leq k \leq k_{max} \tag{3}$$

kink or jerk is defined by $p = \frac{dk}{dR}$ or $p = \frac{\frac{d^3h}{dt^3}}{V^3}$ and is bounded as

$$p_{min} \leq p \leq p_{max} \tag{4}$$

Finally the bound on error is

$$e \geq 0 \tag{5}$$

The objective function (1) is function of height h which is to be minimized subject to constraints (2-5). The curvature k and the kink p are in range domain and analogous to normal acceleration and jerk in time domain.

3. MARKOV MODEL OF TERRAIN

There are several options and methods to model a terrain field. For a random terrain generation, Markov model as described in (Kuchar, 2001) can be used to generate a random terrain data.

Table 1. Fitted autocorrelation function parameters

Terrain Type	σ	τ_0	β
Smooth	79	458	2.2×10^{-3}
Moderately smooth	269	1551	$6.4 \times 10{-}4$
Moderate	342	773	1.3×10^{-3}
Moderately steep	415	492	2.0×10^{-3}
Steep	1007	1633	6.1×10^{-4}

Markov model is a statistical model which can be used for description and generation of a terrain. Markov model involves representing terrain altitude as a stochastic process from which statistics of a terrain can be measured and used to create the terrain profile. Here is used Guass- Markov process to generate a terrain data. A wider range of terrain types can be generated by Guass-Markov process. Consider a discrete time Markov process that has taken values $h_0, h_1, \ldots, h_n$ up to present time n . It is the property of first Markov process that the probability of the next value in the sequence depends upon only the most recent value. A discrete time Markov process can be generated by equation

$$h_{i+1} = e^{-\beta} h_i + \varepsilon_i \tag{6}$$

It is Guass-Markov process when ε_i is a zero mean normally distributed random variable with variance $\sigma^2(1 - e^{-2\beta})$. σ is standard deviation of terrain altitudes in the sample and $\beta = \frac{1}{\tau_0}$ where τ_0 is length scale. Five types of terrain profiles namely smooth, moderately smooth, moderate, moderately steep, steep can be generated. The function parameters for the fitted autocorrelation function for each terrain category are σ, τ_0 and β shown in Table 1.

Using these parameters, a random terrain profile of any above described types can be generated with Eq.(6). This is a reasonable method by which terrain can be generated and handled probabilistically and compactly.

4. TF ALGORITHMS

Two terrain following algorithms are described here in detail.

4.1 Stair Algorithm

It is a simple and computationally fast algorithm. Different parameters are required as input to the Stair Algorithm and before listing some of these parameters need a little explanation. In actual practice terrain altitude vs range data is taken from DEM but here it will be generated from Markov model. Total range is divided into small patches and algorithm processes height of these patches according to imposed constraints. Two dimensional trajectory obtained from terrain avoidance procedure might involve no turn throughout, single turn or multiple turns and the rate of ascend and descend are different for turning and without turning phases. Therefore it is necessary to consider these different rates and for this turning start and end range points should be provided. In order to avoid discontinuities between take-off phase trajectory and low-altitude phase trajectory, the first leg is fixed at an altitude where take-off phase ends. For a smooth transition from descend to an ascend, a descend-ascend gap is selected according to vehicle dynamics. The inputs to the algorithm are:

- Terrain altitude vs range data
- Minimum ground clearance
- Desired patch length for range
- Rate of ascending and descending without turning
- Rate of ascending and descending with turning
- Descend-Ascend gap
- Take-off phase end height
- Constant low-altitude speed
- Turning start points
- Turning end points
- Minimum vertical gap between consecutive patches to merge as single patch

The procedure of Stair algorithm consists of the following steps:

Step 1 Total range is divided into equal intervals that are equal to patch length ΔR and also the corresponding altitude data is collected for each patch except for the first patch whose altitude is taken constant and is fixed at take-off end height. Find the maximum altitude in each interval h_i^{max} and set patch altitude for all patches except for first patch at

$$h_i = h_i^{max} + C_{min} \tag{7}$$

Collect all $h_0, h_1, \ldots, h_n$ and find differences of these altitudes as

$$dh_i = h_i - h_{i-1} \tag{8}$$

When dh_i is positive the vehicle is in ascending mode and vehicle is moving higher in order to avoid any interaction with terrain. Similarly when dh_i is negative the the vehicle is in ascending mode is going down to reduce unnecessary altitude as shown in Figure 1.

Step 2 In ascending case, select nodes as the first point of the next patch and last points of the current patch while descending.

Step 3 For up-stair find intersection point of a line,passing through node point and has ascent slope,with previous patch segment. If it is not on this segment go to the next previous one and continue this until find an intersection point within segment.This point is ascent start point. Collect all ascent start points for up stair and embed them with node points in order. If it is not on first patch then it is not possible to design low-altitude phase path for the given take-off phase trajectory because the first leg

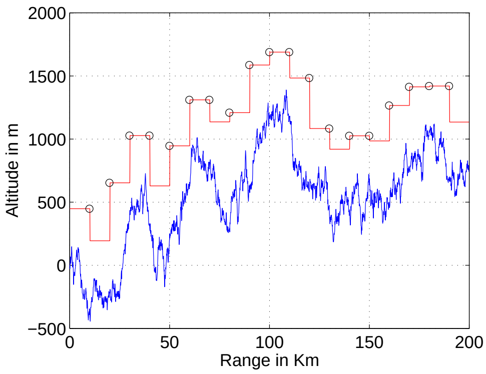

Fig. 1. Terrain Clearance of 300 m for 10 km patch length.

altitude at take-off phase end height is fixed. In this case one need to change either take-off trajectory or UAV start location.

Step 4 For down stair find intersection point of a line passing through node point and has descent slope with next patch segment. If it is not on this segment go to the next one and continue this until find an intersection point within segment. This point is decent stop point. Collect all descent stop points for down stair and embed them with node points. In this way, node list changes which have deleted some node points and at the same time have added ascent start points and descent stop points in increasing horizontal range order.

Step 5 In a valley, scenario becomes complex and two situations are likely to appear. It might happen that descent stop point having higher range is embedded first while ascent start point having lower range is embedded later in the node list as shown in Figure 2. In order to avoid such a situation and also to leave a descend-ascend gap, the following procedure is adopted:

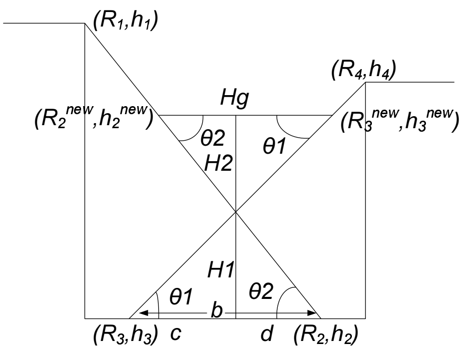

Fig. 2. Reference path calculation in an descend-ascend gap when slope intersect above bottom.

In Figure 2, $(R_1, h_1), (R_2, h_2), (R_3, h_3), (R_4, h_4)$ and the gradients $\tan\theta_1, \tan\theta_2$ are known. Also $b = R_2 - R_3$ and

$$H_1 = c \times \tan\theta_1 \quad \Rightarrow \quad c = \frac{H_1}{\tan\theta_1}$$

$$H_1 = d \times \tan\theta_2 \quad \Rightarrow \quad d = \frac{H_1}{\tan\theta_2}$$

$$\Rightarrow \quad b = c + d = \frac{H_1}{\tan\theta_1} + \frac{H_1}{\tan\theta_2} \tag{9}$$

$$\Rightarrow \quad H_1 = b \times \frac{\tan\theta_1 \times \tan\theta_2}{\tan\theta_1 + \tan\theta_2} \tag{10}$$

and H_2 is calculated as

$$H_2 = hg \times \frac{\tan\theta_1 \times \tan\theta_2}{\tan\theta_1 + \tan\theta_2} \tag{11}$$

Hence new node points $(R_2^{new}, h_2^{new}), (R_3^{new}, h_3^{new})$ can be found as

$$\begin{aligned} R_2^{new} &= R_2 - \frac{H_1 + H_2}{\tan\theta_2} \\ h_2^{new} &= h_2 + (H_1 + H_2) \\ R_3^{new} &= R_3 + \frac{H_1 + H_2}{\tan\theta_1} \\ h_3^{new} &= h_3 + (H_1 + H_2) \end{aligned} \tag{12}$$

Old node points $(R_2, h_2), (R_3, h_3)$ are replaced with new node points $(R_2^{new}, h_2^{new}), (R_3^{new}, h_3^{new})$ in the node list.

Step 6 Second situation occurs when Descend-Ascend gap exists in a valley but is less then given value hg. Raise decent stop and ascent start point up on their slope lines by using trigonometry as shown in Figure 3. Again $b =$

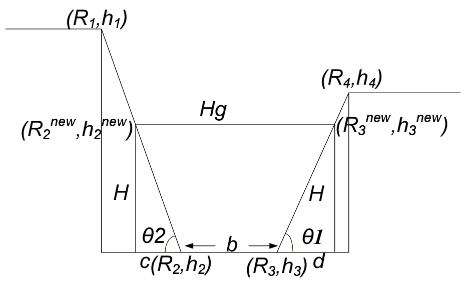

Fig. 3. Reference path calculation in an descend-ascend gap when slope intersect below bottom.

$R_3 - R_2$, $c = \frac{H}{\tan\theta_2}$, $d = \frac{H}{\tan\theta_1}$ and it can be seen from Figure 3 that

$$Hg = b + c + d = (R_3 - R_2) + \frac{H}{\tan\theta_2} + \frac{H}{\tan\theta_1} \tag{13}$$

Hence H can be found as

$$\Rightarrow \quad H = (Hg - R_3 + R_2) \times \frac{(\tan\theta_1 \times \tan\theta_2)}{\tan\theta_1 + \tan\theta_2} \tag{14}$$

and new node points are

$$\begin{aligned} R_2^{new} &= R_2 - \frac{H}{ns}, h_2^{new} = h_2 + H \\ R_3^{new} &= R_3 - \frac{H}{ps}, h_3^{new} = h_3 + H \end{aligned} \tag{15}$$

Remove old node points and embed new node points in the data.

Step 7 Find the intersection of a line formed by using target coordinates and decent slope with a patch starting from target and if intersection is not on this patch continue to the previous patch until an intersection point is found within a patch length. and stop when we get a point on one of the patches. This intersection point is decent start point toward target. Delete all other node points between this intersection point after the target.

This reference path generated considered slope constraint and other constraints can be implemented indirectly by using filters. Next a cubic spline algorithm will be described that implement all constraint at a time using sequential quadratic programming. The purpose is to compare the performance and speed of both approaches to find their effectiveness.

4.2 Cubic Spline Algorithm

Cubic splines (Hoffman, 2001) are third degree polynomials that connect each pair of data points. Each data interval contains its own polynomial that is determined by minimizing an objective function that is subjected into given constraints.For continuity and drivability, the value of the neighboring splines at each interior data point and also their first derivative (slope) and second derivative (curvature) are taken equal. In this way a smooth curve is obtained containing all data points. Here range R is divided into N subintervals of equal length ΔR as shown in Figure 4. So $N+1$ are the total data points and $N-1$ are interior points. Let $H_i : h_i(R) = a_i R^3 + b_i R^2 + c_i R + d_i$ defines a cubic spline in the ith interval $R_i \leq R \leq R_{i+1}$. The spline is optimally computed to lie as close as possible to the terrain and yet to satisfy the practical constraints. Optimizing

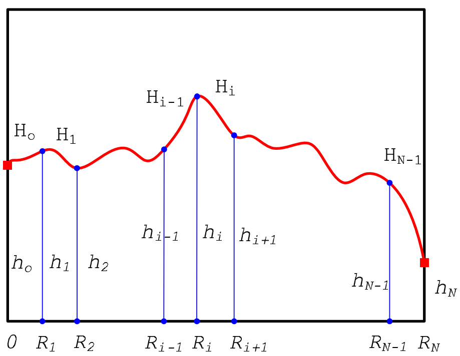

Fig. 4. Range partioning for cubic spline.

parameters are the altitudes $h_1, h_2, \ldots, h_{N-1}$ at the knot points. The altitudes h_0, h_N and also the gradients at h_1, h_{N-1} are fixed and do not change during optimization process. The constraints are forced to satisfy not only at optimizing parameters but also at each given data point and thus making this approach different from the approach given in (Funk, 1977). This will speed up the optimizing process and also at the same will guarantee safety. Mathematically constraint can be written as

$$h_{i-1}(R_i) = h_i(R_i) \tag{16}$$

$$\frac{d}{dR}h_{i-1}(R_i) = \frac{d}{dR}h_i(R_i) \tag{17}$$

$$\frac{d^2}{dR^2}h_{i-1}(R_i) = \frac{d^2}{dR^2}h_i(R_i) \tag{18}$$

Initial condition of slope on first spline and final condition of slope on last spline are

$$\frac{d}{dR}h_0(R_1) = s_0 \tag{19}$$

$$\frac{d}{dR}h_{N-1}(R_{N-1}) = s_N \tag{20}$$

The coefficients of cubic polynomials can be determined by using Equations(16-20) and these a_i,b_i,c_i,d_i are function of altitude h_1,h_2,...,h_{N-1}. The optimization problem is to minimize (1) subject to constraint (2-5) and (16-20). Initial guess of optimization parameters is provided and coefficients of fitted cubic polynomials are found for each interval by solving a system of linear equations. Optimization problem is nonlinear and different algorithms can be used to solve it. Sequential Quadratic Programming (SQP) (Betts, 2001) is used to optimize the test examples. An SQP method obtains search direction for a sequence of quadratic sub (QP) problems. Each QP sub problem minimizes a quadratic model of a certain Lagrangian function subject to linear constraints. A quadratic approximation to the Lagrangian and linear approximation to the constraints is made by using Taylor series. A QP problem is to minimize the quadratic objective function

$$F(h) = g^T h + \frac{1}{2} h^T H h \tag{21}$$

subject to linear equality and inequality constraints.

$$Ah = a \tag{22}$$

$$Bh \geq b \tag{23}$$

Where H is positive definite Hessian matrix. The quadratic objective function is minimized by solving the KKT system of equations

$$\begin{bmatrix} H & A^T & \tilde{B}^T \\ A & 0 & 0 \\ \tilde{B} & 0 & 0 \end{bmatrix} \begin{bmatrix} -p \\ \bar{\eta} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} \bar{g} \\ \bar{a} \\ \bar{\tilde{b}} \end{bmatrix}$$

Where $\tilde{B}$ is the Jacobin matrix of those inequality constraints which are satisfied as equalities and

$$\bar{g} = g^T h + Hh$$
$$\bar{a} = Ah - a$$

$$\bar{\bar{b}} = \bar{\bar{B}}h - b$$

Cubic Spline algorithm proceeds as follows:

Step 1 Give initial guess of altitudes at knot points.

Step 2 Call SQP that will find the optimized parameters by repeatedly solving QP sub-problems for modified hessian and jacobins of both equality and active inequality constraint.

5. SIMULATION RESULTS AND COMPARISON

Both algorithms were tested extensively and the simulation results are shown in Figures (5-7). It has been found that SA is computationally very efficient and can be used for all types of trains but CSA shows little convergence problems for steep terrains which are common for nonlinear programming techniques. Therefore, the comparisons for both algorithms are made using smooth and moderate-smooth types of terrains that are generated by Markov model. One hundred terrains are randomly generated for each type of terrain and performance index (area between optimized trajectory and terrain) is calculated for patch lengths of 5km, 10km and 20km and is shown in Figure 6. The bar graph in Figure 6 shows almost equal performance for same patch length. Standard deviation of performance index for different patch lengths in 100 trials is also shown in Figure 7. It reveals that standard deviation increases in both algorithms by increasing patch length and is almost equal for same patch length. Figure 5 shows trajectories for both CSA and SA for a particular case when terrain is taken from Markov model for 200 km range and patch length is fixed at 4km. This shows that the constraints are well satisfied and are within the desired limits.

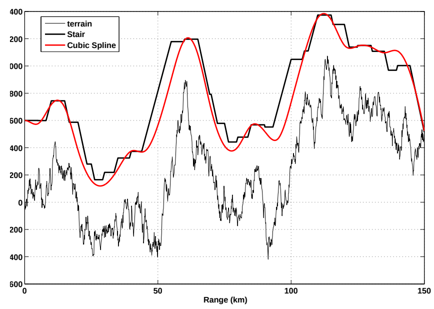

Fig. 5. Optimal trajectory for 160 km range

6. CONCULSION

Two approaches for terrain following of an autonomous vehicle were presented. Both approaches assume that a two dimensional path is available in advance that has been calculated by using a suitable terrain avoidance technique. Stair algorithm

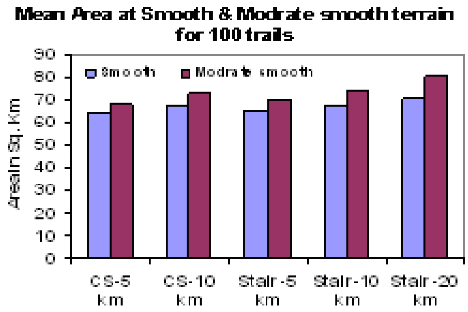

Fig. 6. Mean Area in 100 trails for CSA and SA

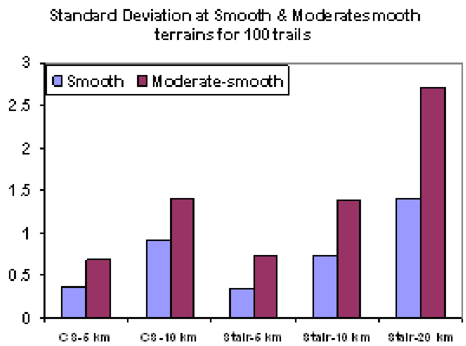

Fig. 7. Standard Deviation in 100 trails for CSA and SA

is fast and computationally efficient for terrain following that can be used online if a forward looking sensor in combination of digital elevation map is used and also easy to implement in a simulation. Cubic spline find an off-line trajectory that is smooth enough to be followed precisely by an actual autonomous vehicle and all constraints are well satisfied and perfect terrain masking over all portions of the terrain is possible. Statistical analysis shows that the standard deviation in performance index data is lesser in case of stair as compare to cubic spline algorithm.

REFERENCES

Betts, J. T . (2001). *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Society of Industrial and Applied Mathematics.

Funk, J. E. (1977). Optimal path precision terrain following system. *Journal of Aircraft* **14**(2), 128–134.

Hoffman, J. D. (2001). *Numerical Mehtods For Engineers and Scientists*. Marcel Dekker , INC.

Kuchar, J. K. (2001). Markov model of terrain for evaluation of ground proximity warning system threshholds. *Journal of Guidance , Control, and Dynamics.*

Lu, P. and B. L. Pierson (1995). Optimal aircraft terrain-following analysis and trajectory generation. *Journal of Guidance , Control, and Dynamics.*

This page has been purposely left blank