Differentiable Trajectory Optimization Under Uncertainty

1 Nomenclature

h	altitude
v	vertical velocity
l	net lift
$w_{\dot{\ell}}$	disturbance on lift
w_v	disturbance on the vertical velocity of the balloon (wind)

2 System Dynamics and Altitude Control

High Altitude Balloons utilize a lifting gas that is lighter than air to produce a buoyant force to counteract the force of gravity on the balloon membrane and the payload. The buoyant force on the balloon minus the force of gravity on the balloon is refereed to as net lift, ℓ. ValBal uses a valve system to vent lifting gas and a ballast system to drop small mass pellets, to change the net lift of the balloon. This net lift produces an acceleration on the system, causing the balloon to accelerate upwards or downwards until the force of drag on the system equals the net lift, at terminal velocity. Because the balloon accelerates to terminal velocity quickly after a change in net lift, the balloon's vertical velocity, v, can be approximately modeled as always traveling at terminal velocity.

While the force of drag on the balloon is nonlinear with respect to velocity, for the purposed of modeling and control, it can be linearized within a range of reasonable velocities. After doing so, the system dynamics can be simplified to:

$$v(t) = k_d\ell \qquad \dot{h}(t) = v(t)$$

With k_d being the linearized coefficient relating net lift to velocity. The valve and ballast system of can be seen as changes to the derivative of the net lift of the balloon, $\dot{\ell}$. For example, when the valve is open, $\dot{\ell}$ becomes negative as the balloon looses lift. In addition, the atmosphere is full of disturbances, so in the system dynamics we consider 2 other terms: $w_{\dot{\ell}}$ as the atmospheric disturbance on the lift rate, and w_v, as the atmospheric disturbance of velocity. For example, at night the balloon cools and looses

lift, which is modeled as a negative $w_{\dot{\ell}}$, and vertical winds are modeled as a nonzero w_v. With these additions, the system dynamics become

$$\dot{v}(t) = k_d(\dot{\ell}(t) + w_{\dot{\ell}}(t)) \qquad \dot{h}(t) = v(t) + w_v(t)$$

Using the following substitution we can write the a state-space representation of the system.

$$x = \begin{bmatrix} h \\ v \end{bmatrix} \qquad u = \dot{\ell}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ k_d \end{bmatrix} u + \begin{bmatrix} w_v \\ k_d w_{\dot{\ell}} \end{bmatrix}$$

3 Motivation

4 Formulation

The physical system is represented by discrete states $s_k \in \mathcal{S}$ defined at times $t_k = t_s + k\Delta t$, for $k = 0, 1, \ldots$ In the case of ValBal, we can let $\mathcal{S} = \mathcal{H} \times \mathcal{V} \times \Lambda \times \Phi$, or $\mathcal{S} = \mathcal{H} \times \mathcal{L} \times \Lambda \times \Phi$, if we're considering altitudes, velocities, lifts, latitudes, and longitudes.

We will need functions that define the transition between states, and the value function, while guaranteeing differentiability on both.

4.1 Differentiable altitude trajectory

The transition function needs to generate a state s' from a state s, and take into account the optimization variables θ. These parameters can, for instance, be a set of altitude waypoints defined at various points of the flight, or altitude tolerances at each of those points. So a simple transition function could be $f(s, t; \theta) \to s'$. If our state consisted only of altitudes, and we were to simply linearly interpolate between altitude waypoints $\theta_0, \theta_1, \ldots$ (defined at $t_0^w, t_1^w, \ldots$) we would find the parameters relevant to the current timestep t_k such that $t_i^w < t_k < t_{i+1}^w$. Then we simply give:

$$f(h_k, t_k; \theta) = \theta_i + (t_k - t_i^w)\frac{\theta_{i+1} - \theta_i}{t_{i+1}^w - t_i^w}$$

which has a very simple derivative with respect to the optimization variables θ.

However, due to the uncertainties in the prediction of the transitions, we would like to ideally output a *distribution* of states s' we might end up on—each rollout of a trajectory would follow then a unique path. In order to preserve differentiability, we introduce a parameter λ that can be interpreted as a percentile of the distribution that we select. At the beginning of a rollout, a sequence $\lambda_0, \lambda_1, \ldots$ is randomly generated such that the variation results in a noise level consistent with flight data. In the limit where all λ are 0.5, the median trajectory is selected; in a more realistic simulation, the percentile would slowly fluctuate between smaller and larger numbers such that a particular rollout would

be flying low or high respectively with respect to a noise-free simulation. While we note that this only has an analytical expression for very simple distributions, a numerical derivative can easily be taken by keeping λ fixed and varying θ.

4.2 Differentiable horizontal trajectory

The horizontal integration needs to preserve differentiability to be able to compute a full gradient with respect to every optimization parameter. We use atmospheric data provided by NOAA which, in particular, includes the velocity field $w(h, \lambda, \phi, t) \rightarrow (u, v)$, where h is the altitude defined on a set of forecast altitudes;[1] λ and ϕ are the coordinates defined on a (not necessarily regular) grid of latitudes and longitudes;[2] t is a forecast time (usually separated by 1 to 3 hours) and (u, v) are the two horizontal components of wind.

For completely regular grids, bilinear interpolation will suffice. For irregular grids, a scheme such interpolation weighted by inverse distance $(\sum^K \alpha_i w_i)/(\sum^K \alpha_i)$ for the closest K points would do the job. So given a function $g(s_k, t_k)$ (which has no explicit dependence with respect to the optimization variables) we find $(u_k, v_k) = g(s_k, t_k)$. We subsequently perform an Euler step to get the coordinates of the next state:

$$x_{k+1} = x_k + u_k \Delta t \quad y_{k+1} = y_k + v_k \Delta t$$

and the conversion to (λ, ϕ) follows from a simple computation of spherical coordinates.

4.3 Differentiable value function

In this case, our value function can be extracted directly from the (differentiable) computations in Section 4.2. For a rollout of length K, the total value is the horizontal distance:

$$V = \sum_{k=0}^{K-1} u_k \Delta t \tag{1}$$

where Δt is a constant that does not affect the result. Other valid objective functions could include the distance to a particular point:

$$V = \sum_{k=0}^{K-1} \|r_k - r_{\text{target}}\|$$

5 Gradient optimization

We consider the objective function given in Eq. 1. We seek the gradient with respect to the optimization variables θ. Since the sum is trivially separable, we consider the

---

[1]The data from NOAA is defined at barometric pressures, which can be mapped monotonically to altitude.
[2]Some models are more regular than others; for instance, NAM is defined on a rather annoying Lambert conformal grid.

3

gradient of one of the horizontal velocities with respect to the one of variables, $\partial u_k / \partial \theta_i$. Then:

$$\frac{\partial u_k}{\partial \theta_i} = \frac{\partial u_k}{\partial s_{k,j}} \frac{\partial s_{k,j}}{\partial \theta_i}$$

Repeated indices implicitly imply summation. We note that, in the case from above, the only relevant component of the state is altitude, and the first term is simply:

$$\frac{\partial u_k}{\partial h_k} = \frac{\partial g(h_k, \lambda_k, \phi_k, t)}{\partial h_k}$$