

An Indirect Method for Optimal Guidance of a Glider

Romain Pepy, Bruno Hérisse

Onera - The French Aerospace Lab, F-91761 Palaiseau, France
{romain.pepy,bruno.herisse}@onera.fr

Abstract: This paper addresses the problem of finding the trajectory, for a glider, that maximizes the final velocity and respect both initial and terminal conditions on the state. A new set of differential equations governing the system, deduced using optimal control theory, is proposed. To solve this system of equations, an algorithm based on an indirect shooting method is described. It provides paths for which the final speed is higher than the one obtained by a current common closed-loop method.

1. INTRODUCTION

The purpose of this paper is to present a new guidance method for a glider. A glider is an unpowered aircraft that can maneuver using only aerodynamic forces generated by tail fins or wings. Many flying devices can be considered as gliders such as sailplanes, rockets or missiles during their non-propelled phases or high-speed gliders. To increase its range, a glider needs to maintain its speed. In other words, it needs to minimize the effect of the drag force along its trajectory while maintaining enough lift to be able to maneuver and reach the target position.

The guidance scheme we propose in this paper is efficient for all gliders which flies at a speed high enough to neglect the gravity. It ensures boundary constraints on both position and orientation, and maximizes the final velocity of the system. It takes into account the heterogeneous environment in which the glider moves. Indeed, when the glider is climbing, the density of air changes and the effects of both drag and lift forces are reduced.

The methods the most commonly used to solve such a problem come from the missile guidance community. These guidance laws simplify the problem in order to find an analytic solution to the problem in order to obtain a fully closed-loop guidance law. These laws can be embedded on real gliders and compute control inputs *online* since they require low computational time. Most analytical solutions are derived from the proportional navigation law [Newman, 1996]. They include a proportional navigation term that moves the vehicle towards the final position and a shaping term to reach this position with the desired orientation. These laws are very classical but the numerous approximations made in order to find an analytical form, imply that the resulting trajectories are often far from the optimal one. The best known is probably the kappa guidance law ([Lin, 1991]) which maximizes the final velocity of an unpowered system under several simplifying assumptions. More details on this law are given in section 5.1. Other closed-loop guidance laws exist, using singular perturbation theory [Cheng and Gupta, 1986], [Menon and Briggs, 1990], [Dougherty and Speyer, 1997] or linear quadratic regulators [Imado et al., 1990], [Imado and

Kuroda, 1992] for example. Simplifications and assumptions made to obtain these fully closed-loop guidance laws imply that they are either only locally optimal or optimal under some unrealistic conditions.

On the other hand, many direct numerical methods exist to solve this problem [Hull, 1997]. In direct methods, the states and control inputs of a problem are approximated using a specified functional form. Direct collocation methods, especially global orthogonal collocation methods (also called pseudo-spectral methods) are the most popular for solving constrained trajectory optimization problems. These methods use polynomials to approximate the state and collocation is performed at some chosen points. Chebyshev or Lagrange polynomials are often used. Collocation points can be Gauss-Lobato points [Elnagar et al., 1995] [Fahroo and Ross, 2002], Legendre-Gauss points [Benson et al., 2006] and more recently Legendre-Gauss-Radau points [Kameswaran and Biegler, 2008] [Fahroo and Ross, 2008] [Garg et al., 2011]. These methods are efficient to solve trajectory optimization problems. They are nowadays mainly used *offline* and cannot be easily embedded since they need a large computing power.

In this paper, we propose an indirect method (*i.e.* a method where optimality conditions are determined by the calculus of variations) for solving the problem of trajectory optimization. This method is an alternative to direct methods and has the advantage of being less expensive in computation time than a direct method. It also does not make many approximations as classical guidance laws. It allows to find optimal trajectories and can be considered embedded on a glider.

The body of the paper consists of three sections. The problem and models are detailed in section 2. In section 3, conditions of optimality are enumerated and, using the maximum principle, a new optimal guidance equation is presented. An implementation of this equation using a shooting method is described in section 4. Finally, compared results between our indirect method and kappa guidance are presented.

2. PROBLEM STATEMENT

The problem solved in this paper can be described as follows: find the trajectory for a high speed glider from an initial state $\mathbf{x}_0 = (x_0, z_0, \theta_0)$ with an initial velocity v_0 to a goal state $\mathbf{x}_f = (x_f, z_f, \theta_f)$ that maximizes the terminal velocity v_f .

2.1 System modeling

The glider is modeled as a rigid body of mass m maneuvering in a vertical 2D plane. The coordinate system $(x, z, \theta) \in \mathbb{R}^3$ is used to represent the position $\boldsymbol{\xi} = (x, z)$ of the center of mass G where z is the altitude and the orientation θ of the glider which is the angle between the glider and the x -axis. The missile dynamics are

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{z} = v \sin \theta \\ \dot{\theta} = \frac{f_L}{m v} \\ \dot{v} = \frac{f_T - f_D}{m} \end{cases} \quad (1)$$

where $v > 0$ is the longitudinal speed, gravity is neglected, the thrust force $f_T = 0$ since the glider is unpowered, f_D is the drag force defined as

$$f_D = \frac{\rho S C_D v^2}{2}$$

and f_L is the lift force defined as

$$f_L = \frac{\rho S C_L v^2}{2}$$

where ρ is the density of air, S is the glider reference area,

$$C_D = C_A \cos \alpha + C_N \sin \alpha, \quad (2)$$

$$C_L = C_N \cos \alpha - C_A \sin \alpha. \quad (3)$$

C_D and C_L are respectively the drag and the lift coefficients in velocity frame, C_A and $C_N = C_{N_\alpha} \alpha$ are respectively the drag and lift coefficients in body frame, α is the angle of attack of the missile and $C_{N_\alpha} = dC_N/d\alpha$ is the lift coefficient curve slope.

2.2 Environment modeling

In the followings, the density of air ρ is modeled as an exponential function decreasing with the altitude

$$\rho(z) = \rho(0) e^{-\frac{z}{z_r}} \quad (4)$$

where z_r is a reference altitude. As we consider a glider with only aerodynamic flight controls, the maneuvering capabilities are linked to the density of air (4) and approach zero at 35 km.

3. OPTIMALITY CONDITIONS

In this section, optimality conditions are determined using the maximum principle [Pontryagin, 1962].

Considering (1) and an unpowered aircraft, \dot{v} can be expressed as

$$\dot{v} = -\frac{1}{2m} \rho v^2 S C_D. \quad (5)$$

Since small angles of attack are considered, (2) and (3) become

$$\begin{cases} C_L = (C_{N_\alpha} - C_A) \alpha \\ C_D = C_A + (C_{N_\alpha} - C_A/2) \alpha^2 \end{cases}$$

and C_D can be written as

$$C_D = C_A + \frac{(C_{N_\alpha} - C_A/2)}{(C_{N_\alpha} - C_A)^2} C_L^2.$$

Let $u \in [-1, 1]$ denote the control input defined as

$$u = \frac{C_L}{C_{L_{\max}}}.$$

Thus, we obtain

$$C_D = C_A + \eta C_{L_{\max}} u^2 \quad (6)$$

where $\eta = \frac{(C_{N_\alpha} - C_A/2)}{(C_{N_\alpha} - C_A)^2} C_{L_{\max}}$ is an aerodynamic efficiency factor.

Using (5) and (6), \dot{v} becomes

$$\dot{v} = -(d + \eta c u^2) v^2 \quad (7)$$

where $d(z) = \frac{1}{2m} \rho S C_A$ and $c(z) = \frac{1}{2m} \rho S C_{L_{\max}}$ is the maximum curvature.

A change of variable is done from time t to curvilinear abscissa $s(t) = \int_0^t v(t) dt$. Thus, dynamics can be rewritten as

$$\begin{cases} x' = \cos \theta \\ z' = \sin \theta \\ \theta' = c(z) u \end{cases} \quad (8)$$

and

$$v' = -(d + \eta c u^2) v. \quad (9)$$

Our goal is to maximize the final velocity v_f . Let $w = \ln(v)$, then $w' = -(d + \eta c u^2)$ and maximizing the final velocity v_f is equivalent to maximizing $w_f = \ln(v_f)$.

Since w does not appear in the right side of equations (8), the dimension of the system considered in our optimization problem can be reduced to 3. The speed only appears in the cost function.

The subsequent optimal control problem is defined as follows: given the system of ordinary differential equations (8) with the initial state $\mathbf{x}_0 = (x_0, z_0, \theta_0)$ and the final state $\mathbf{x}_f = (x_f, z_f, \theta_f)$, it is required to find an optimal control, *i.e.* a function $u^*(s)$ and the corresponding path $\mathbf{x}^*(s)$ for which the functional

$$J = \int_0^{s_f} (d + \eta c u^2) ds \quad (10)$$

is minimized.

The Hamiltonian function is

$$H(\mathbf{x}, u, \boldsymbol{\lambda}) = (d + \eta c u^2) + \lambda_1 \cos \theta + \lambda_2 \sin \theta + \lambda_3 c u \quad (11)$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^T$ is the costate vector of the system. Pontryagin's minimum principle (Pontryagin [1962]) states that the optimal control u^* , path \mathbf{x}^* and costate vector $\boldsymbol{\lambda}^*$ must minimize the Hamiltonian H so that for all $u \in \mathbb{R}$

$$H(\mathbf{x}^*, u^*, \boldsymbol{\lambda}^*) \leq H(\mathbf{x}^*, u, \boldsymbol{\lambda}^*)$$

for every curvilinear abscissa s .

In the followings, we consider that u is not constrained thus $\frac{\partial H}{\partial u} = 0$, hence

$$u = -\frac{\lambda_3}{2\eta} \quad (12)$$

Using the costate equations, we obtain

$$\lambda_1' = -\frac{\partial H}{\partial x} = 0.$$

Using $\lambda'_2 = -\partial H/\partial z$, (4) and (12), it follows that

$$\lambda'_2 = \frac{1}{z_r} (d - \eta c u^2). \quad (13)$$

And finally λ'_3 is given by

$$\lambda'_3 = -\frac{\partial H}{\partial \theta} = \lambda_1 \sin \theta - \lambda_2 \cos \theta. \quad (14)$$

By differentiating (12) with respect to s and using (14),

$$u' = -\frac{\lambda_1 \sin \theta - \lambda_2 \cos \theta}{2\eta}$$

is obtained and its second derivative, using (13), is

$$u'' = \frac{d - \eta c u^2}{2\eta} \left(c u + \frac{\cos \theta}{z_r} \right). \quad (15)$$

To sum up, the Ordinary Differential Equations (ODE)

$$\begin{cases} x' &= \cos \theta \\ z' &= \sin \theta \\ \theta' &= c(z) u \\ u' &= \frac{du}{ds} \\ u'' &= \frac{d - \eta c u^2}{2\eta} \left(c u + \frac{\cos \theta}{z_r} \right) \end{cases} \quad (16)$$

solve the trajectory optimization problem for a glider. (16) has no analytic solution and can only be solved with numerical methods, for example the Runge-Kutta method or the Hermite-Simpson method.

One may notice that if ρ does not change with the altitude (*i.e.* $\rho(z) = \rho_0$) and if we consider only small curvatures $c u$, then solving (15) yields

$$u = A e^{-\sqrt{\frac{cd}{\eta}} s} + B e^{\sqrt{\frac{cd}{\eta}} s} \quad (17)$$

which is the equation of kappa guidance presented in [Lin, 1991] page 570. Consequently, (16) generalizes the kappa guidance law. However, it does not provide an explicit guidance law of the form $u = f(x, z, \theta)$.

4. IMPLEMENTATION

In this section, an implementation of (16) to solve the optimal trajectory problem is provided (section 4.1). This implementation is based on a simple shooting method and only aims to illustrate the efficiency of our approach to provide optimal trajectories, it does not focus on any other aspects such as computational time or robustness (section 4.2). This method is named Indirect Method for Glider Guidance (IMGG) in the followings.

4.1 Indirect Shooting Method

The propagation of the trajectory from the start to the end can be done using the solution of the ODE initial value problem as the control function $u(s)$ is defined for every curvilinear abscissae.

To demonstrate the efficiency of (16), a shooting method is used to solve the trajectory optimization problem. Since (16) has been obtained using the maximum principle, this shooting method is said *indirect*. This method is described in the algorithm 1. It requires an initial guess on the unknown conditions: the control input u_0 , its first

Algorithm 1 Indirect Shooting Method

Input: Initial guesses on initial values u_0, u'_0, s_f

Output: Trajectory from \mathbf{x}_0 to \mathbf{x}_f that maximize v_f

- 1: **repeat**
 - 2: Integrate (16) on $s = [0, s_f]$ to obtain $\hat{\mathbf{x}}_f$
 - 3: Constraints evaluation in terminal conditions, evaluate $\psi(\hat{\mathbf{x}}_f)$ using (18)
 - 4: Update initial values u_0, u'_0, s_f
 - 5: **until** $\psi(\hat{\mathbf{x}}_f) < \epsilon$
-

derivative u'_0 and the path length s_f . (16) is integrated from $s = 0$ to $s = s_f$ using 4th order Runge-Kutta method. The final state is $\hat{\mathbf{x}}_f$. The boundary condition $\psi(\hat{\mathbf{x}}_f)$ between \mathbf{x}_f and $\hat{\mathbf{x}}_f$ is computed on line 3. $\psi(\hat{\mathbf{x}}_f)$ is then used to update the initial values u_0, u'_0 and s_f . These three lines are repeated until the error $\psi(\hat{\mathbf{x}}_f)$ becomes less than a given ϵ .

The cost function $\psi(\hat{\mathbf{x}}_f)$ that evaluates the terminal constraints has to consider both position $\boldsymbol{\xi} = (x, z)$ and orientation θ at the final state. An example of cost function is the pseudo-metric

$$\psi(\hat{\mathbf{x}}_f) = \left(\frac{\|\hat{\boldsymbol{\xi}}_f - \boldsymbol{\xi}_f\|}{\|\boldsymbol{\xi}_f - \boldsymbol{\xi}_0\|} \right)^2 + \|\hat{\theta}_f - \theta_f\|^2. \quad (18)$$

Gradient methods cannot be used without finite-difference approximations of derivatives to update the values u_0, u'_0 and s_f since the derivative of the cost function cannot be obtained analytically. In our implementation, we use a Nelder-Mead algorithm (Nelder and Mead [1965]) as described in Lagarias et al. [1998] which allows to avoid approximations of derivatives and appears to be more efficient than gradient methods to get out of local minima.

4.2 Issues

The main issue with this simple implementation of the indirect shooting method is the initialization of the iteration process. It is hard to find a first estimate (also called a *guess*) of the unknown initial conditions that provides a trajectory that is not “wild” as mention in Bryson and Ho [1975] page 215. This initialization is all the more important as the solutions are very sensitive to changes in the initial conditions.

Since our goal in this paper is to prove the effectiveness of our indirect method to find optimal trajectories and not to improve its execution time or its robustness, we will not go further into the analysis of this well known problem. The reader can refer to Bryson and Ho [1975] or Betts [1998] for more information on it.

5. RESULTS

In this section, the kappa guidance is presented in section 5.1 and compared results between our indirect method IMGG and kappa guidance are given in section 5.2.

5.1 Reminder on kappa Guidance

The kappa guidance is a closed-loop guidance law described in Lin and Tsai [1987] and Lin [1991]. This curvature-based guidance law, designed for non-propelled

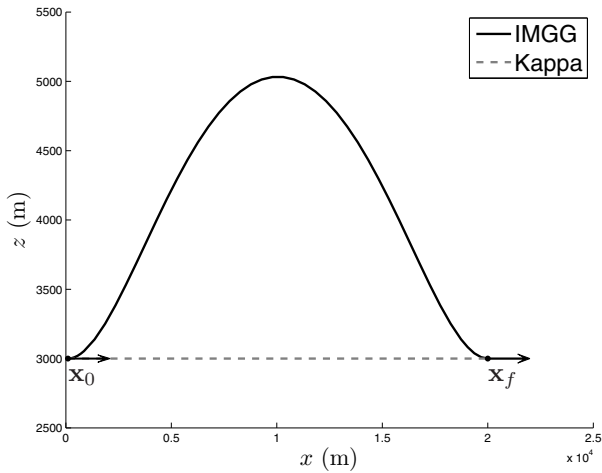


Fig. 1. IMGG, trajectory (x, z) , $\mathbf{x}_0 = (100, 3000, 0)$, $\mathbf{x}_f = (20000, 3000, 0)$

rocket, aims to maximize its final velocity. Since dynamics (1) are close to those used in Lin [1991], the kappa guidance law can be used to guide a glider efficiently. This law makes numerous of approximations (see section 3), especially on the environment modeling. The control input \mathbf{a} returned by this guidance law is a lateral acceleration perpendicular to the longitudinal velocity vector \mathbf{v} :

$$\mathbf{a} = \frac{\mathbf{v}}{\|\mathbf{v}\|} \times \left(\boldsymbol{\gamma} \times \frac{\mathbf{v}}{\|\mathbf{v}\|} \right)$$

with

$$\boldsymbol{\gamma} = \frac{K_1}{t_{go}}(\mathbf{v}_f - \mathbf{v}) + \frac{K_2}{t_{go}^2}(\boldsymbol{\xi}_f - \boldsymbol{\xi} - \mathbf{v} t_{go})$$

where K_1 and K_2 are gains obtained using (17), t_{go} is the time-to-go (*i.e.* the estimate duration to reach the goal), \mathbf{v}_f is the terminal velocity vector and $\boldsymbol{\xi}_f$ is the terminal position vector.

$$\frac{K_2}{t_{go}^2} (\boldsymbol{\xi}_f - \boldsymbol{\xi} - \mathbf{v} t_{go}),$$

is called the *proportional navigation* term and tends to moves the aircraft to the target position.

$$\frac{K_1}{t_{go}} (\mathbf{v}_f - \mathbf{v})$$

where $\|\mathbf{v}_f\| \stackrel{\text{def}}{=} \|\mathbf{v}\|$, is called the *shaping* term and tries to reach the target with the desired orientation. $\|\mathbf{a}\|_{\max}$ is the maximum lateral acceleration the glider can perform. It is straightforward to see that $u = \|\mathbf{a}\| / \|\mathbf{a}\|_{\max}$.

This law does not provide globally optimal trajectories.

As many closed-loop guidance laws, a good guess of the *time-to-go* t_{go} is needed to compute the control input \mathbf{a} . Such a good guess is hard to obtain since it depends on the trajectory from the current position to the target position.

The trajectories provided by kappa guidance are only locally optimal if ρ changes since it does not anticipate future changes in flight conditions along the trajectory. Moreover, if the flight conditions change significantly between the start point and the end point, the trajectories returned by the kappa guidance tend to be impossible to follow by the real glider: lateral accelerations returned by the guidance law are higher than that the glider is actually capable of performing.

Finally, the kappa guidance law does not take into account the dynamic of ρ and so it does not take advantage of the heterogeneity of the environment. For example, if the air density is identical between $\boldsymbol{\xi}_0$ and $\boldsymbol{\xi}_f$, the kappa guidance law does not take into account the heterogeneous nature of the environment, *i.e.* the fact that the effect of the drag force can be reduced by climbing up and then climbing down to reach $\boldsymbol{\xi}_f$. Thus, the terminal velocities obtained with the kappa guidance can be far from the optimal ones.

These remarks are illustrated in the next section.

5.2 Compared results

In this section, results showing the trajectories provided by IMGG are presented. These results are compared with trajectories returned by the kappa guidance law.

The first example is typical of how our indirect method based on (16) works. In this example, the initial and terminal points, respectively $\mathbf{x}_0 = (100, 3000, 0)$ and $\mathbf{x}_f = (20000, 3000, 0)$, are located at the same altitude (3000 m). The initial speed of the glider is $v_0 = 1000$ m/s.

In the case of kappa guidance, the trajectory between those two points is a straight line which length is 19900 m, the control input u is always equals to 0. As explained in the previous section, this is due to the fact that the kappa guidance does not handle properly heterogeneous environments. Instead, using IMGG (Figure 1), the glider climbs to an altitude of 5000 meters to reduce the effect of the drag force along its trajectory (since the density of air decreases with altitude). The generated trajectory is 20404 m long. Thus, although the path is longer than in straight line, the terminal velocity is higher: 290 m/s for our method against 263 m/s for the kappa guidance.

The second example shows a case where IMGG is able to find an admissible trajectory (*i.e.* a trajectory where u always stays between -1 and 1) while in the case of kappa guidance, the normalized control input $\|\mathbf{a}\| / \|\mathbf{a}\|_{\max}$ does not stay in this interval. The initial configuration is $x_0 = 100$ m, $y_0 = 500$ m, $\theta_0 = \pi/8$ rad and the final configuration is $x_f = 5000$ m, $y_f = 15000$ m, $\theta_f = \pi/8$ rad. The initial speed is still $v_0 = 1000$ m/s.

The solid line in figure 2(a) represents the trajectory generated by the indirect method described in this paper. The path is made up of two main turns, the first one starts at about $z = 2000$ m and the second one starts at $z = 8000$ m. Our method can take into account the fact that the final state is at an altitude where the air density is low. At this altitude, the aircraft can not operate effectively. So in order to find a solution, it is necessary to anticipate this and to start shaping the trajectory from the beginning of the flight. As we see on figure 2(b) (solid line), u stays between -0.8 and 0.5 .

Such an anticipation is impossible for closed-loop laws such as kappa guidance. These laws tend to favour the proportional navigation term at the beginning of the flight (when the time-to-go is large) and only really consider the shaping term at the end. This explains why the first turn starts later than for IMGG (dashed line on figure 2(a)). It starts at an altitude $z = 3000$ m. Then, the second turn starts at $z = 9500$ m and the kappa guidance law

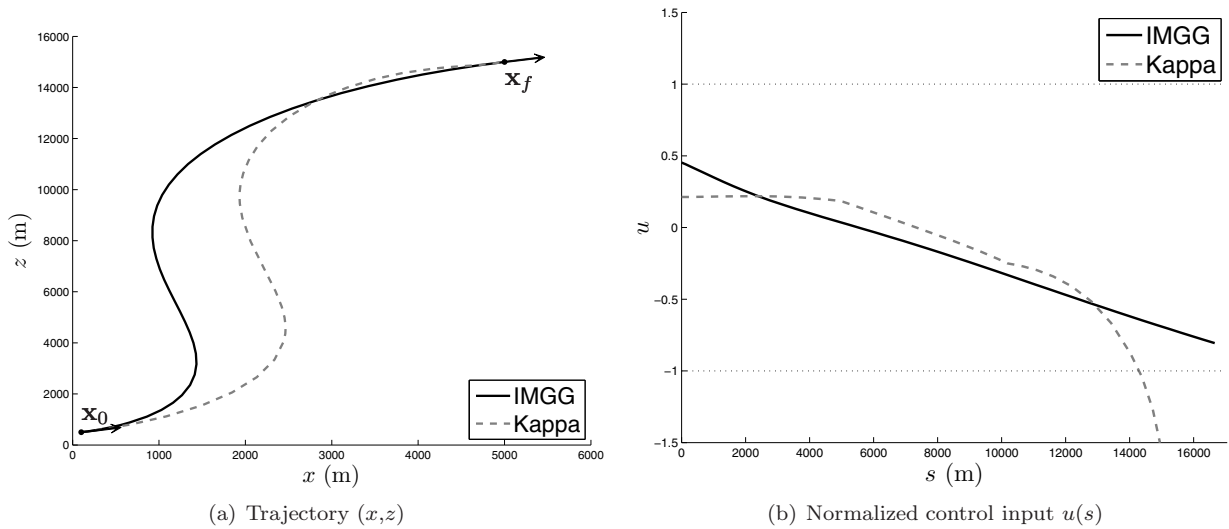


Fig. 2. Compared results between IMG and kappa guidance, $\mathbf{x}_0 = (100, 500, \pi/8)$, $\mathbf{x}_f = (5000, 15000, \pi/8)$

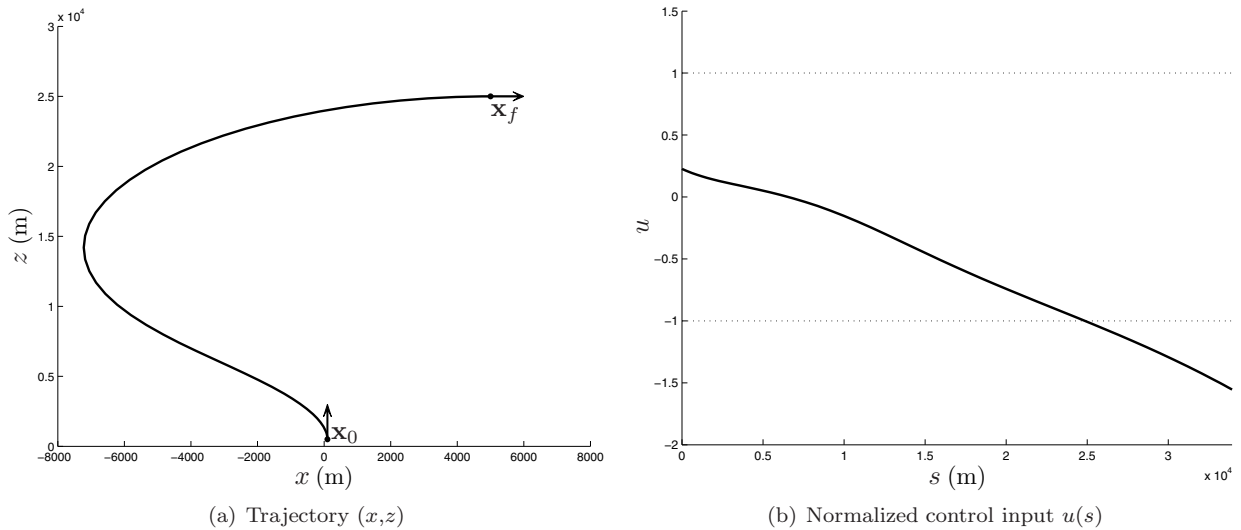


Fig. 3. IMG, $\mathbf{x}_0 = (100, 500, \pi/2)$, $\mathbf{x}_f = (5000, 25000, 0)$

needs to increase the control inputs to reach the terminal conditions (figure 2(b), dashed line). Since the glider can no longer operate effectively at this altitude and speed, its normalized control input $\|\mathbf{a}\| / \|\mathbf{a}\|_{\max}$ becomes greater than 1: the path is invalid since he could not be followed by the real glider.

6. PERSPECTIVES

In section 5.2 are shown some results to illustrate the effectiveness of (16) presented in section 3 using a simple implementation (section 4). This implementation needs to be improved to satisfy constraints such as real time execution, robustness to initial unknown conditions and the need to take into account saturation. Thus, it could allow our indirect method to be embedded on an actual autonomous glider. This is the purpose of this section, which presents our current and future works, in addition to the transition from 2D to 3D.

6.1 Real-time computation

One interesting perspective is related to the computation time of the trajectories generated by IMG. Although the implementation done in section 4 is not optimized for that, the computation time is still less than what can be achieved with optimization software using pseudo-spectral methods (eg GPOPS (Rao et al. [2010])).

The first step in this optimization process aim at changing the algorithm that searches for the minimum of the cost function (line 5 of the algorithm 1). The Nelder-Mead algorithm, at least the basic implementation we used, allows us to get results but tends to converge slowly to the optimal solution.

The second way to improve the computation time is to change the shooting method. This method, in addition to being highly sensitive to initial conditions, cannot be optimized from the computation time point of view. Indeed, it requires to fully integrate the system (16) from $s = 0$ to $s = s_f$ before moving to the next stage of the

algorithm. No parallelization of the algorithm is possible. Using a multiple-shooting method (by dividing the interval $s = [0, s_f]$ in small pieces) could allow us to parallelize a piece of the algorithm. Furthermore, multiple shooting methods are more robust to initial conditions (see Betts [1998]).

The underlying objective is to run the algorithm in real-time and to obtain a fully closed-loop guidance law.

6.2 Saturations

Saturations, *i.e.* cases where $|u|$ becomes greater than 1 and so curvatures are greater than those that the real glider can achieve, are not taken into account in (16).

For example, in figure 3, the initial state is $\mathbf{x}_0 = (100, 500, \pi/2)$ and the final state is $\mathbf{x}_f = (5000, 25000, 0)$. In this example, IMGG is able to find a trajectory. However, as the altitude at the final state is very high, the maneuvering capabilities of the glider are almost zero near \mathbf{x}_f . The glider must go at the opposite of the desired final position at the beginning of its flight to minimize the curvature at the end. Despite this, the control input is greater than 1 at the end of the trajectory. This problem appears when the problem is really hard to solve or when an admissible solution does not exist.

The results of Hérissé and Pepy [2013] could be used to address this problem and to take into account saturations in the guidance algorithm. The authors generate the shortest paths by extending the concept of Dubins curves in a heterogeneous environment and illustrate this using a glider. The trajectories are composed of curves where the control input u is either 1, -1 (maximum curvature curves) or 0 (segment lines) analogously to Dubins' curves.

7. CONCLUSION

This paper presented a new indirect method for finding the optimal trajectory between two states of a glider that maximizes its terminal velocity. Equations of guidance are deduced using optimal control theory. A simple implementation is performed using an indirect shooting method. The results presented illustrate the performance of our indirect method compared to current fully closed-loop methods. Future work will focus on improving the computation time of a path and making the algorithm robust to initialization errors.

REFERENCES

- D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao. Direct trajectory optimization and costate estimation via an orthogonal collocation method. *Journal of Guidance, Control, and Dynamics*, 29:1435–1440, 2006.
- J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.
- A. E. Bryson and Y. C. Ho. *Applied Optimal Control: Optimization, Estimation and Control*. Taylor & Francis, 1975.
- V. H. L. Cheng and N. K. Gupta. Advanced midcourse guidance for air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 9(2):135–142, 1986.
- J. J. Dougherty and J. L. Speyer. Near-optimal guidance law for ballistic missile interception. *Journal of Guidance, Control, and Dynamics*, 20(2):355–362, 1997.
- G. Elnagar, M. Kazemi, and M. Razzaghi. The pseudospectral legendre method for discretizing optimal control problems. *IEEE Transactions on Automatic Control*, 40(10):1793–1796, 1995.
- F. Fahroo and I. M. Ross. Direct trajectory optimization by a chebyshev pseudospectral method. *Journal of Guidance, Control, and Dynamics*, 25(1):160–166, 2002.
- F. Fahroo and I. M. Ross. Pseudospectral methods for infinite-horizon nonlinear optimal control problems. *Journal of Guidance, Control, and Dynamics*, 31:927–939, 2008.
- D. Garg, M. A. Patterson, C. L. Darby, C. Francolin, G. T. Huntington, W. W. Hager, and A. V. Rao. Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method. *Computational Optimization and Applications*, 49(2):335–358, 2011.
- B. Hérissé and R. Pepy. Shortest paths for the Dubins' vehicle in heterogeneous environments. In *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, 2013. To appear.
- D. G. Hull. Conversion of optimal control problems into parameter optimization problems. *Journal of Guidance, Control, and Dynamics*, 20(1):57–60, 1997.
- F. Imado and T. Kuroda. Optimal guidance system against a hypersonic targets. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 1992.
- F. Imado, T. Kuroda, and S. Miwa. Optimal midcourse guidance for medium-range air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 13(4):603–608, 1990.
- S. Kameswaran and L. T. Biegler. Convergence rates for direct transcription of optimal control problems using collocation at radau points. *Computational Optimization and Applications*, 41(1):81–126, 2008.
- J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
- C. F. Lin. *Modern Navigation Guidance and Control Processing*. Prentice-Hall, Inc., 1991.
- C. F. Lin and L. L. Tsai. Analytical solution of optimal trajectory-shaping guidance. *Journal of Guidance, Control, and Dynamics*, 10(1):61–66, 1987.
- P. K. Menon and M. M. Briggs. Near-optimal midcourse guidance for air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 13(4):596–602, 1990.
- J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.
- B. Newman. Strategic intercept midcourse guidance using modified zero effort miss steering. *Journal of Guidance, Control, and Dynamics*, 19(1):107–112, 1996.
- L. S. Pontryagin. *Mathematical Theory of Optimal Processes*. John Wiley & Sons, 1962.
- A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington. Gpops: A matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Transactions on Mathematical Software*, 37(2):22, 2010.