

# TRAJECTORY OPTIMIZATION OF A SMALL AIRSHIP IN A MOVING FLUID

Eric Lanteigne<sup>1</sup>, Charles Blouin<sup>1</sup>, Wail Gueaieb<sup>1</sup>

<sup>1</sup>*Department of Mechanical Engineering, University of Ottawa, Ottawa, ON, Canada*

*Email: eric.lanteigne@uottawa.ca; cblou045@uottawa.ca; wgueaieb@uottawa.ca*

---

## ABSTRACT

Small airship offer low cost alternative for long endurance flights for telecommunications, surveillance and transport. Maneuvering in a real environment while taking into account the obstacles, the dynamical model of the airship, the wind and the energy consumption is a complex problem. This paper will present the dynamical model of a small airship and its experimental characterization, followed by the description of the path planning problem formulated as an optimal control problem. The problem will then be solved using a pseudo-spectral solver, implemented in the software PSOPT. This paper demonstrates that an optimal control approach can be used to generate complex and very realistic trajectories for small airships.

**Keywords:** optimal control; airship; dirigeable; trajectory optimization; psendo-spectral method.

---

## OPTIMIZATION DE LA TRAJECTOIRE D'UN PETIT DIRIGEABLE DANS UN FLUIDE EN MOUVEMENT

### RÉSUMÉ

Les petits dirigeables offrent une alternative à bas coût pour les vols de longue endurance pour les télécommunications, la surveillance, et le transport. Manoeuvrer dans un environnement réel, prenant en compte les obstacles, le modèle dynamique du véhicule, du vent, et de la consommation d'énergie est un problème complexe. Ce document présentera le modèle dynamique d'un petit dirigeable et sa caractérisation dynamique expérimentale, suivie par la description du problème d'optimisation de trajectoire présenté comme un problème de contrôle optimal. Le problème sera ensuite résolu en utilisant une méthode pseudo-spectrale, implémenté dans le logiciel PSOPT. Ce papier démontre qu'une approche de contrôle optimale peut être utilisée pour générer des trajectoires complexes et très réaliste pour un petit dirigeable.

**Mots-clés :** Contrôle optimal, dirigeable, trajectoire optimale, méthode psendo-spectrale.

## 1. INTRODUCTION

Airships use has seen a renewed interest since the beginning of the millennium. Their inherent safety, long range and time of flight makes them ideal as mobile telecommunication, surveillance, and monitoring units [1]. The United States army had a monitoring program involving an airship stationed at high altitude and a prototype was built in 2011 [2]. It was designed to provide 16kW to the surveillance payload, have an operating range of 2000 miles and stay in the air for 21 days. The long range airtime possible with airship makes them ideal for use as telecommunication platform. During the take-off phase and for close proximity operation, there is a need to precisely control the airship, and optimal control solver are well suited for those operations. They have been extensively used for planning the motion of airplanes [3] and rockets [4].

## 2. MODELING OF THE AIRSHIP

The airship, seen in Fig. 1, is modeled as a six degrees of freedom mass in a viscous fluid, with viscous damping in rotation and drag in its linear movements. The following assumptions are used:

1. Six degrees of freedom point mass.
2. Lift is negligible.
3. The drag is proportional to the square of the velocity.
4. Rigid body

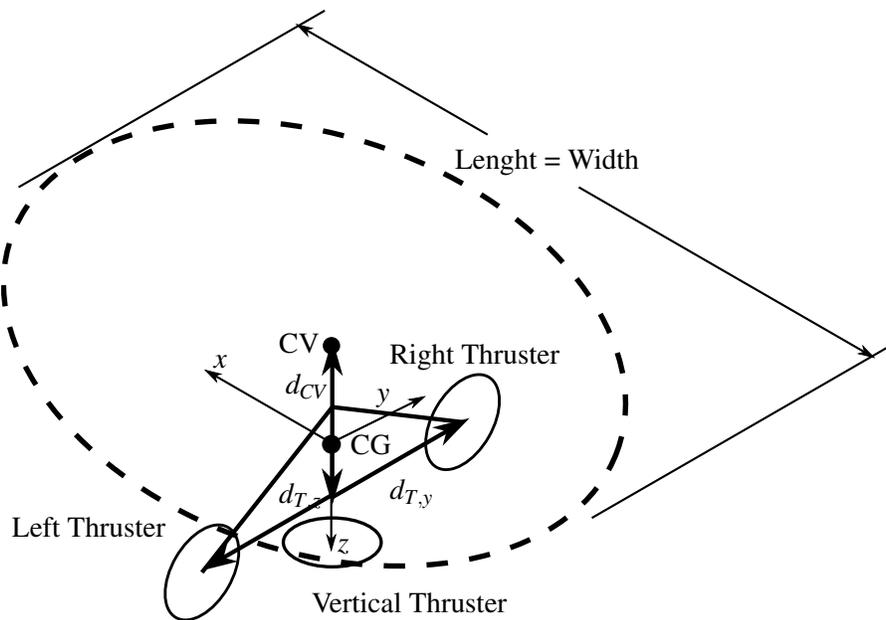


Fig. 1. Airship model

Airships are typically modeled using Newton's equations in a local frame of coordinate, with the axes fixed to the body. The center of reference often chosen for modeling is the center of volume. The large mass of air displaced by the airship, relatively to the mass of the airship itself, will create a virtual mass and inertia [5].

The governing equations for an airship can be expressed as:

$$M\ddot{x} = F_d\dot{x} + A\dot{x} + G + PU \quad (1)$$

Where  $M$  is a  $6 \times 6$  matrix mass matrix, including the rotational inertial and the virtual masses,  $x$  is the state of the vehicle,  $F_d \dot{x}$  is a dynamical term due to the offset between the center of axis and the center of mass,  $A \dot{x}$  is the aerodynamic term,  $P$  is the propulsion matrix, and  $U$  is the control vector. The vehicles are most often under-actuated, so the vector  $U$  will often have less than six rows.

The rotational drag terms are often ignored for large vehicle since, at high speed, the aerodynamic forces from the control surface become relatively large. The rotational drag coefficient is often estimated by approximating the vehicle as a large cylinder, using potential flow assumption, or in wind tunnel [6]. In the case of small airships, such as the one studied in this paper, the damping from fluid forces cannot be neglected. A method for experimentally determining the viscous drag term not requiring a wind tunnel will be presented.

The problem of finding a path for airships and planes is described with a cost function, which is generally an integral over the path. This integral is function of the local wind vector, the time, the path local orientation and vehicle's properties. To explore possible paths, one approach is to discretize the search space in nodes and segment connecting those nodes [7]. A weight based on the cost function is assigned to each segment, and algorithm such as A\* or B\* can be used [8]. Although fast, those methods are limited to long paths relative to the vehicle size, as they do not take into account the dynamical model of the vehicle.

In this article, the cost function will be minimized using the pseudo-spectral method for optimal control. Pseudo-spectral solution solver were initially investigated for solutions to PDEs in fluid dynamics. For example, solutions to the well known 1D burger equation as well as the 3D Euler equations were obtained using those solvers [9].

Optimal control for trajectory optimization has historically been used in aerospace for the optimal ascent of rockets as it takes into account all the differential constraints of the model and of the problem [10]. Since analytical solutions are only possible in very simple and specific cases, numerical solutions are used and will be presented here for an airship.

### 3. KINEMATIC EQUATIONS

The optimal control solver is independent of the method used for the pose description. A possible choice is the use of quaternion. Since they use four variable instead of three, the number of states describing the would increase and a degree of freedom is added, which is undesirable for solving the the problem with an optimal control solver, as it increases the size of the matrices involved and the memory used. The Euler angle description may cause a gimbal lock when pointing at zenith or nadir, but an airship is unlikely to perform this kind of extreme maneuver. Two reference frames are used: the body frame of reference, fixed to the airship and denoted with the superscript  $b$ , and an inertial frame of reference, fixed to the ground and denoted by the superscript  $i$ . The orientation of the vehicle is represented with a three angles:  $\phi$ ,  $\theta$  and  $\psi$ . The convention used will be roll-pitch-yaw. Fig. 2 show the angle convention used and the frame of reference.

We define the following rotation matrices:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}, R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, R_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

To transform a vector  $v^b$  in the inertial frame to a vector  $v_i$  in the inertial frame, we apply the following transformation:

$$R_z R_y R_x v^b = S^{b \rightarrow i} v^b = v^i \quad (3)$$

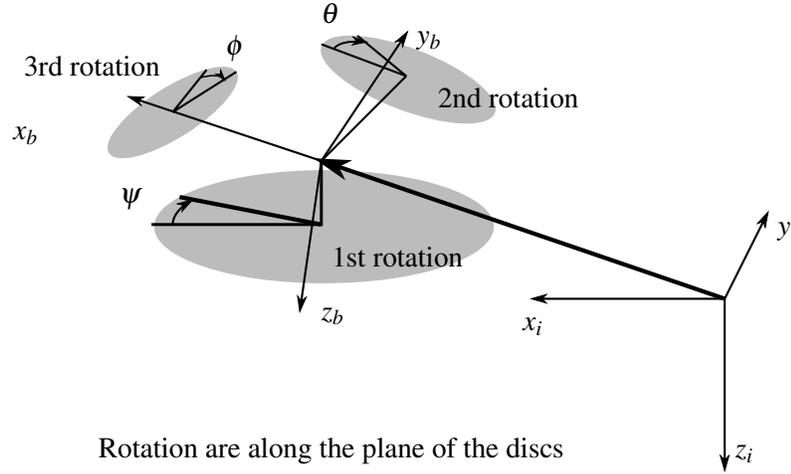


Fig. 2. Angle convention

$$\begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} v^b = v^i \quad (4)$$

The opposite transformation is done with a transpose of  $S^{i \rightarrow b}$  such that

$$S^{i \rightarrow b} v^i = v^b \quad (5)$$

#### 4. KINETIC EQUATIONS

The airship is modeled as a solid mass of mass  $m$  and of inertia  $I$ . To simplify the calculation for the solver, the equation will be written in the body reference frame, unlike other papers on airships. The forces acting on the vehicle are the gravity force  $F_g$ , the buoyancy for  $F_b$ , the thrust force  $T_f$  and the aerodynamic forces  $F_a$ . The moments acting of the vehicle are the moments due to the inertia  $I$ , the drag  $C_r$ , and the thrust forces.

The velocity change in the body reference frame is:

$$\begin{bmatrix} \dot{V}_x^b \\ \dot{V}_y^b \\ \dot{V}_z^b \end{bmatrix} = F^b / m - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \cdot \begin{bmatrix} V_x^b \\ V_y^b \\ V_z^b \end{bmatrix} \quad (6)$$

The position change in the inertial frame is:

$$\begin{bmatrix} \dot{x}_x^i \\ \dot{x}_y^i \\ \dot{x}_z^i \end{bmatrix} = R_z R_y R_x V^b \quad (7)$$

The angular rate of change in the body frame is:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \dot{\omega} = I^{-1} (M^b - \omega \times I \omega) \quad (8)$$

The airship angular rate of change in the body frame is found by rotating each vector of the Euler angular rate of change in the body frame.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & \sin -\phi & \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & \sin -\phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (9)$$

The jacobian matrix  $J^{-1}$ , function of  $\phi$ ,  $\theta$  and  $\psi$ , relates the rate of change of the vehicle in the body frame and the derivative of the pose in Euler angles.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (10)$$

The inverse of the matrix  $J^{-1}$  is shown in the following equation:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad J = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad (11)$$

In summary, the state space equations are Eq. (6), Eq. (7), Eq. (8) and Eq. (11).

The input forces in the body frame,  $F^b$ , and the input moments in the body frame,  $M^b$ , are both modeled around the center of gravity. Those input forces are due to the buoyancy force, the gravity, the aerodynamic drag and lift, as well as the thrust from the propellers.

$$F^b = S^{i \rightarrow b} B^i + S^{i \rightarrow b} g^i + D^b + T^b \quad (12)$$

For the experimental model, the buoyancy force was canceled by the gravity force using small weights before each experiment to make the vehicle neutrally buoyant. The drag  $D^b$ , seen in Eq. (13) in the body frame is proportional to the incoming airflow velocity  $V_r$  and the thrust is proportional to the propeller rotational speed. The function  $C(\phi, \theta, \psi, V_w)$  is the coefficient of drag, and it depends on the angle of attack of the airship.

$$D^b = C(\phi, \theta, \psi, V_w) * V_r^b * |V_r^b|, \quad V_r^b = V_v^b - S^{i \rightarrow b} V_w^i \quad (13)$$

The variable  $V_w$  is the wind velocity and  $V_v$  is the vehicle velocity. Fig. 3 shows visually the relationship between the velocities; a relationship that is true in the inertial frame and in the body frame.

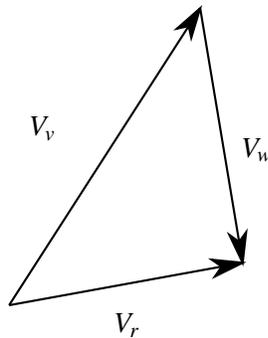


Fig. 3. Relative velocities of the airship

The thrust vector  $T$  is the sum of the thrust from the three thrusters on board, that are producing the thrust  $T_1$ ,  $T_2$  and  $T_3$ .

$$T = T_1 + T_2 + T_3 \quad (14)$$

The sum of moment is:

$$M^b = d_B S^{i \rightarrow b} B^i + d_D D^b + M_T^b \quad (15)$$

Where  $d_B$  is the distance between the center of gravity and the center of volume, and  $d_D$  is the distance between the center of gravity and the center of drag. The variable  $M_T^b$  is the sum of moments caused by the thrust forces from the thrusters.

#### 4.1. Drag Model

The drag is assumed to be perpendicular to the airflow. The drag coefficient was found experimentally, by capturing the airship position in free fall. The vehicle was weighted for this part of the experiment. The fall was recorded using the optical positioning system explained in the experimental section. By varying the position of the weight, it was possible to control the orientation of the vehicle during the fall.

The equation governing this experiment is given by:

$$ma = F_g - F_d \quad (16)$$

where  $m$  is the mass of the vehicle,  $F_g$  is the force of gravity and  $F_d$  is the drag force. The drag force is then:

$$F_d = Cv|v| \quad (17)$$

where  $v$  is the velocity of the vehicle relative to the wind. Since the vehicle is drop with no lateral velocity, the problem is simplified to a one dimension problem and the solution to Eq. (16) and Eq. (17) equation is:

$$x(t) = \frac{m \log \left( \cosh \frac{\sqrt{C} \sqrt{F_g} (c_1 m + t)}{m} \right)}{C} + c_2 \quad (18)$$

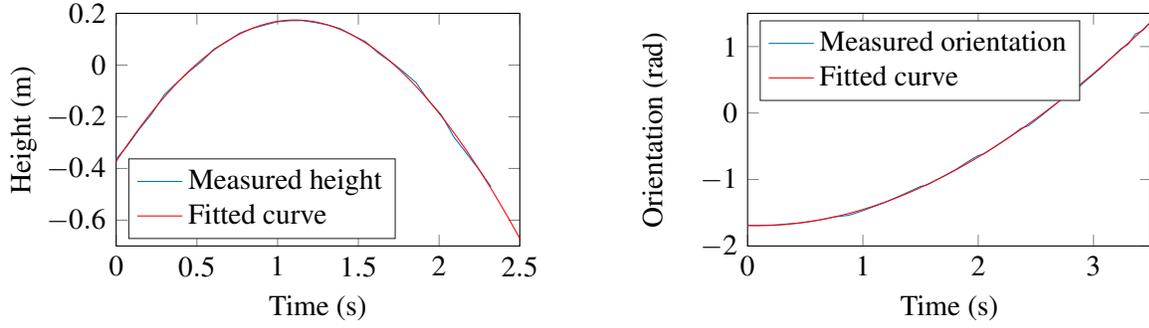
The equation was the fitted to the experimental results in Matlab. There are four unknown in this equation:  $c_1$ ,  $c_2$ ,  $m$  and  $C$ . The value of  $c_1$  and  $c_2$  are not not important here, since they depend on the time the experiment was started and the absolute value of the height measured. The value of  $C$  however is the drag coefficient for the orientation measured, and  $m$  is the mass of the airship added plus the mass of the added weight. Multiple drop test were performed to find the drag and the mass constants shown in Table 1. An example of one of those fit can be seen in Fig. (4). The mass is experimental since it includes the added mass from the displaced airflow. Multiple papers used an added mass equal to 50 % of the mass of the airship. The experimental data shows that the added mass depends on the orientation of the airflow and varies between 30 % of added mass to 110% of added mass for this particular shape.

#### 4.2. The general optimal control problem

Mathematically, the general optimal control problem is defined as followed. The notation use will be the same as the one used by the software PSOPT. Given the state trajectories  $x(t)$ ,  $t \in [t_0, t_f]$ , the static parameter  $p$ , and times  $t_o, t_f$ , find the optimal control trajectory  $u(t)$ ,  $t \in [t_0, t_f]$ , to minimize the following performance index:

$$J = \phi(x(t_0), x(t_f), p, t_0, t_f) + \int_{t_0}^{t_f} L(x(t), u(t), p, t) dt \quad (19)$$

subject to the differential constraints



(a) Height as a function of time for the drop test in the forward-aft direction (b) Orientation as a function of time during yaw characterization

Fig. 4. Characterisation experiment of the airship

Table 1. Experimental value characterizing the airship

Variable	Description	Value	unit	95 % conf.
$C_x$	Horizontal linear drag coefficient	0.046		$\pm 0.04$
$C_z$	Vertical linear drag coefficient	0.11		$\pm 0.04$
$m_{ex}$	Experimental total mass of the vehicle	0.077	kg	$\pm 0.010$
$m_{ez}$	Experimental total mass of the vehicle	0.117	kg	$\pm 0.020$
$m_b$	Mass of the gondola, battery and balloon	0.0357	kg	$\pm 0.005$
$m_g$	Theoretical mass of the gas	0.0258	kg	$\pm 0.001$
$m_b + m_g$	Mass of the vehicle without virtual mass	0.0565	kg	$\pm 0.005$
$\frac{m_{ex}}{m_b + m_g}$	Horizontal virtual mass ratio	1.4		$\pm 0.2$
$\frac{m_{ez}}{m_b + m_g}$	Vertical virtual mass ratio	2.1		$\pm 0.4$
$I_{ez}$	Experimental inertia around the $z$ axis	$2.7 \cdot 10^{-3}$	$\text{kg} \cdot \text{m}^2$	$\pm 0.2 \cdot 10^{-3}$
$I_{ex}$	Experimental inertia around the $x$ and the $y$ axis	$6.0 \cdot 10^{-3}$	$\text{kg} \cdot \text{m}^2$	$\pm * 10^{-3}$
$C_{rz}$	Rotational drag coefficient around $z$	$2.7 \cdot 10^{-4}$		$\pm 0.7 \cdot 10^{-4}$
$C_{rx}$	Rotational drag coefficient around $x$	$9.7 \cdot 10^{-4}$		$\pm 2 \cdot 10^{-4}$

$$\dot{x}(t) = f(x(t), u(t), p, t), t \in [t_o, t_f] \quad (20)$$

the path constraints

$$h_L \leq h(x(t), u(t), p, t) \leq h_U, t \in [t_o, t_f] \quad (21)$$

the events

$$e_L \leq e(x(t_o), u(t_o), x(t_f), u(t_f), p, t_o, t_f) \leq e_U, t \in [t_o, t_f] \quad (22)$$

the bound constraints:

$$\begin{aligned} u_L \leq u(t) \leq u_U, t \in [t_o, t_f] \quad x_L \leq x(t) \leq x_U, t \in [t_o, t_f] \\ p_L \leq p \leq p_U \quad t_{oL} \leq t_o \leq t_{oU} \\ t_{fL} \leq t_f \leq t_{fU} \end{aligned} \quad (23)$$

and the constraint  $t_f - t_o \geq 0$ . The path constraint  $h$  is a useful constrain to add obstacles, as restriction of the states  $x(t)$ ,  $y(t)$  and  $z(t)$ , expressed as an inequality. For example, the constraint  $x^2(t) + y^2(t) \geq 1$  will

exclude all the path going through a circle of radius 1 centered at the origin. The event constraint is used to specify a range for the start and end point of the time, the states, the control input and the static parameter of the problem.

The problem can be described in multiple phases, which is useful when discontinuities appear in the problem or when the equations governing the system change. The phase will then be described by the superscript  $i$ , with  $i = 1 \dots N_p$  where  $N_p$  is the number of phases. Additional events are usually added to link the phases.

The problem will be solved using a pseudo spectral solver. With this method, all the functions in the problem are represented with orthogonal functions. For this problem. the functions used are Legendre polynomial, which can be differentiated and integrated very quickly. The optimal control problem is then transformed in a non-linear programming problem, where the function to optimize is the Lagrangian  $L$ , while respecting the constraints imposed on the problem. For the trajectory problem studied, the Lagrangian is the time, and the constraints to respect are the differential constraints and the bound constraints, such as the maximum thrust of the vehicle. The advantages of the pseudo-spectral approach are that the convergence is very fast (exponential) and the results follow closely the dynamical model. The solver used is implemented in the software PSOPT, with Legendre polynomial and LGL nodes. The complete description of the method is available in [11].

## 5. SIMULATION RESULTS

This section will demonstrate the results the trajectory optimization of the vehicle. Two problems will be presented. The model used was presented in Section 2 and fully implemented in the solver.

### 5.1. Simulation 1

The first problem presented will starts with the vehicle at the origin and the axis oriented with the origin. The initial and final conditions for the first problem are presented in Table 2.

Table 2. Simulation parameter for the first experiment

Time	$x$ (m)	$y$ (m)	$z$ (m)	$\phi$ (rad)	$\theta$ (rad)	$\psi$ (rad)
Initial	0	0	0	0	0	0
Final	1.5	0.5	0	0	0	1
	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{z}$ (m/s)	$p$ (rad/s)	$q$ (rad/s)	$r$ (rad/s)
Initial	0	0	0	0	0	0
Final	0	0	0	0	0	0

Since the vehicle moves on a plane, the vertical axis is not shown in the results, although the full vehicle dynamics is used for simulation. In reality, the thrusts from the left and right propeller create a torque which makes the vehicle pitch up. The vertical thruster corrects this effect to keep the vehicle leveled.

The resulting trajectory is shown in Fig. 5. The vehicle accelerates in a straight line in the first second of the trajectory. As seen in Fig. 6b, the vehicle points out the curve and reverses its thrust to reorient the vehicle's velocity vector toward the final position. The final part of the path is in a straight line, where the airship accelerate and decelerate to finish in the position and the orientation desired. The normalized thrust for each motor is shown in Fig. 6b. This kind of optimal maneuverer would be very difficult to generate with traditional controllers or path planner. The dynamical model has to be taken into account to minimize the time.

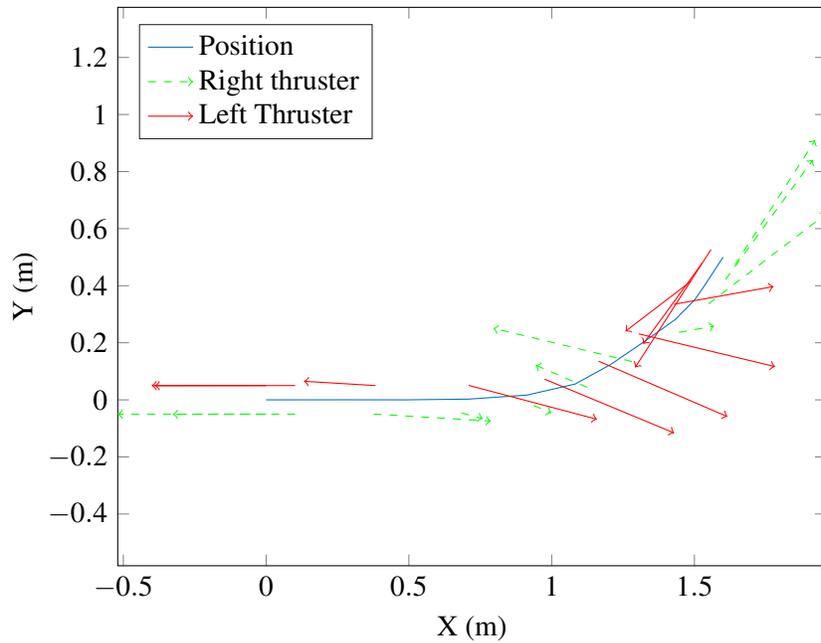


Fig. 5. Path of the vehicle and thrust of the right (green) and left (red) motor at different positions along the path.

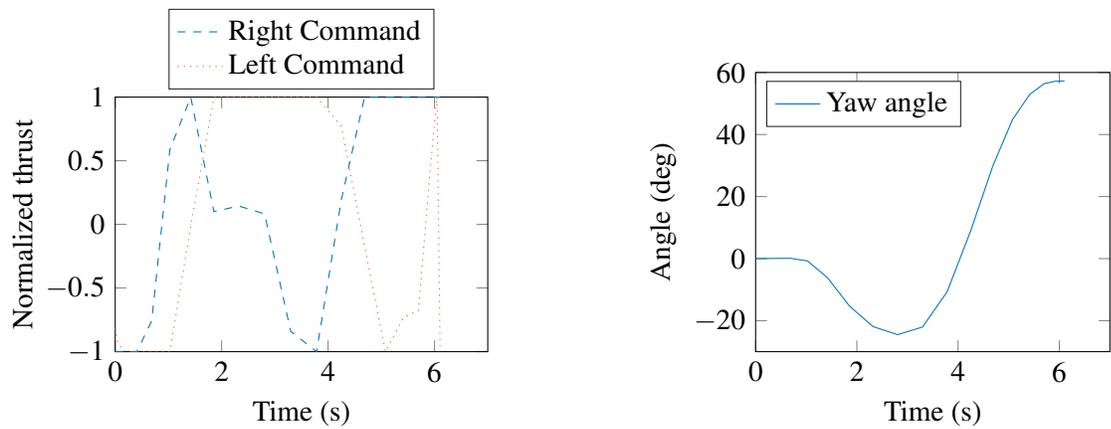
## 5.2. Second Simulation

The second simulation demonstrates how lateral movements are possible with the airship, even though it is under-actuated. The airship will finish this simulation to the left of the starting point, in the same orientation, but with a lateral speed. This kind of maneuver could be useful for storing the airship in a similar manner to a parallel parking for cars. Due to the lateral movement, the path would be impossible to generate with a traditional linear controller. The initial and final conditions for the second problem are presented in Table 3. The trajectory generated is shown in Fig. 7, the commands send to the motor are shown in Fig. 8a and the orientation of the vehicle as a function of time is shown in Fig. 8b.

Table 3. Simulation parameters for the second experiment

Time	$x$ (m)	$y$ (m)	$z$ (m)	$\phi$ (rad)	$\theta$ (rad)	$\psi$ (rad)
Initial	0	0	0	0	0	0
Final	0	0.3	0	0	0	0
	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{z}$ (m/s)	$p$ (rad/s)	$q$ (rad/s)	$r$ (rad/s)
Initial	0	0	0	0	0	0
Final	0	0.1	0	0	0	0

In the Fig. 7, the vehicle is shown accelerating, then turning to its right and reversing thrust. This maneuver causes the vehicle to gain lateral velocity. In the last part of the path, the vehicle cancels out its velocity in the x direction and orients itself to 0 radians, while keeping a velocity in the y direction. The air resistance slows down the vehicle to a rest position.



(a) Left and right thrust as a function of time (b) Yaw orientation of the vehicle as a function of time

Fig. 6. Thrust and yaw angle as a function of time for the second experiment

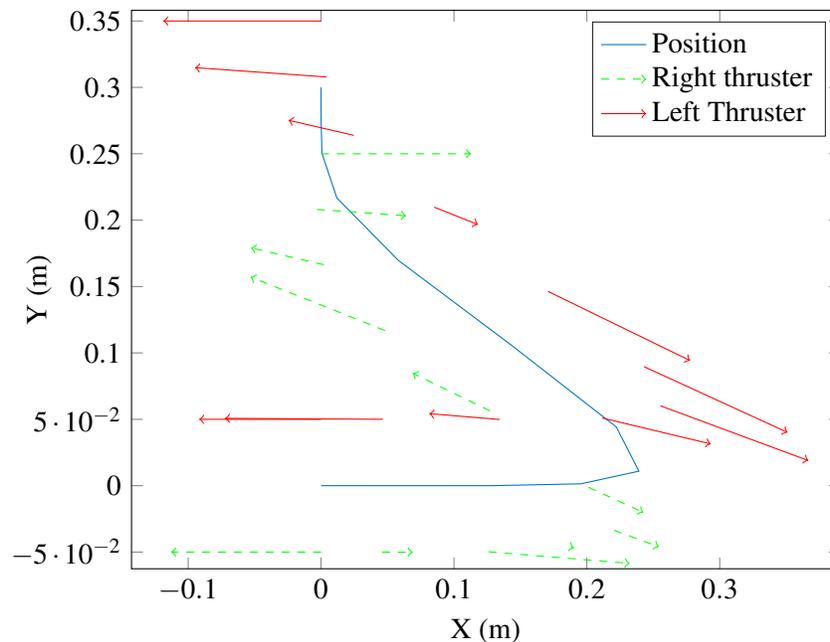
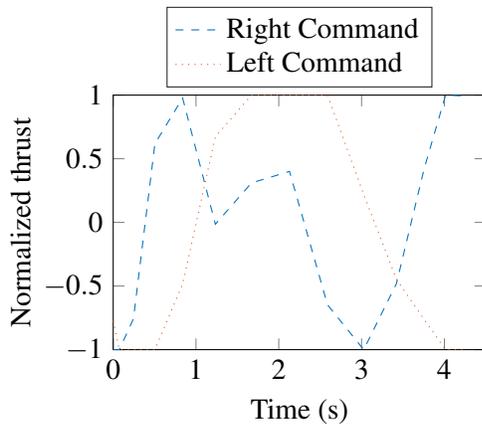


Fig. 7. Path of the vehicle and thrust of the right (green) and left (red) motor at different positions along the path.

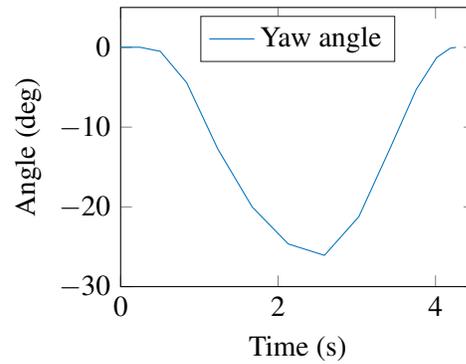
## 6. DISCUSSION

The optimal solver used generates very complex path locally optimal and respecting the differential equations. The time to solve is under 1 second on a modern computer. An important advantage of the optimal solver is the flexibility it offers, as almost any dynamical equation can be used. For instance, a propeller model taking into account the inertia and the drag of the propeller could be added by adding states and dynamical constraints. The solver and problem formulation would not change.

The limitations of the solver are the convergence issues, that the quality of the result depends on the accuracy and precision of the model, and that the polynomial used to represent the states and the controls have poor performances in the presence of discontinuities.



(a) Left and right thrust as a function of time



(b) Yaw orientation of the vehicle as a function of time

Fig. 8. Thrust and yaw angle as a function of time for the second experiment

Due to the very wide search space, the solution is not guaranteed to converge, especially if the initial guess is poor. Currently, a linear interpolation between the initial and final condition is used as a guess for every state, and the controls are initially assumed to be zero. In certain cases where the solution diverges significantly from guess, the solver fails to converge, which is unacceptable for real time application if the optimal solver is the only path planner used on an airship. Thus, an application on a flying vehicle would likely include a simple, discrete path planner that is used as the default trajectory, and the optimal controller can be used to improve the trajectory to reduce fuel consumption or the time required to attain the objective.

The second issue is that drift in the path will occur since there are small errors in the models and a controller still has to be used. The controller is out of the scope of this paper. Another solution is possible to re-plan the path using the optimal solver when too much error has accumulated. The dynamical model can be updated to better reflect the changing condition, such a wind condition, changing mass or even a broken actuator.

Another issue with a spectral solvers is that discontinuities are approximated with large error, unlike "smooth" solution. This issue can be tackled by using multiple phases since discontinuities are allowed between two phases.

Given the strengths and limitation of the optimal solver, the results could be used in real applications to precalculate specific open loop maneuver. For instance, parts of the landing and take-off procedure could be open loop, and the optimal solver could be used to generate the command. There could also be precalculated commands to reorient the airship quickly. The optimal control solver can also be used to verify the result of other path planner that use approximations, discretization, genetic algorithm or other.

## 7. CONCLUSIONS

This research described the theoretical and experimental modeling of a small airship. The optimal trajectory problem was then formulated as an optimal control problem and solved using a pseudo-spectral method. The results are showing that complex and locally optimal trajectory can be generated. The simulation demonstrated that the problem can be solved very quickly, in the order of magnitude of a second, which is fast enough from real time control of large vehicle. The path generated can be followed by a controller, or used in open loop for short maneuver. Optimal control solvers are ideal in situations in which the dynamical model is well known, when an optimal solution taking into account the dynamical model is needed, or to serve as a benchmark for other path planners.

## REFERENCES

1. Waishek, J., Dogan, A. and Bestaoui, Y. "Investigation into the time varying mass effect on airship dynamics response." In "47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition," January. ISBN 9781563479694, 2009.
2. Cummings, J.S. "Long Endurance Multi-Intelligence Vehicle (LEMV) Agreement Signed.", 2010.  
URL <http://www.army.mil/article/41024/long-endurance-multi-intelligence-vehicle-lemv-agreement-signed/>
3. Sorensen, J.a.S.J.a., Morello, S.a.M.S.a. and Erzberger, H.E.H. "Application of trajectory optimization principles to minimize aircraft operating costs." *1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, Vol. 18, pp. 415–421. ISSN 01912216. doi:10.1109/CDC.1979.270208, 1979.
4. Zhuang, Y., Ma, G., Huang, H. and Li, C. "Real-time trajectory optimization of an underactuated rigid spacecraft using differential flatness." *Aerospace Science and Technology*, Vol. 23, No. 1, pp. 132–139. ISSN 12709638. doi:10.1016/j.ast.2011.06.010, 2012.
5. Hygounenc, E., Jung, I.K., Souères, P. and Lacroix, S. "The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping." *The International Journal of Robotics Research*, Vol. 23, pp. 473–511. ISSN 02783649. doi:10.1177/0278364904042200, 2004.
6. Li, Y., Nahon, M. and Sharf, I. "Airship dynamics modeling: A literature review." *Progress in Aerospace Sciences*, Vol. 47, No. 3, pp. 217–239. ISSN 03760421. doi:10.1016/j.paerosci.2010.10.001, April 2011.
7. Wang, H., Li, Q. and Cheng, N. "Real-time path planning for low altitude flight based on A\* algorithm and TF/TA algorithm." In "Automation Science and Engineering (CASE)," pp. 837 – 842, 2012.
8. Meng, B.b. and Gao, X. "UAV Path Planning Based on Bidirectional Sparse A\* Search Algorithm." In "Intelligent Computation Technology and Automation," pp. 1106 – 1109, 2010.
9. Bernard Legras, D.G.D. "A Comparison of the Contour Surgery and Pseudo-spectral Methods." *Journal of Computational Physics*, Vol. 104, No. 2, pp. 287–302, 1993.
10. Casalino, L., Colasurdo, G. and Pastrone, D. "Optimal Low-Thrust Escape Trajectories Using Gravity Assist." *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 5, pp. 637–642.
11. Becerra, V.M. "PSOPT Optimal Control Solver User Manual (Release 3).", 2011.