

Optimal Control for the Trajectory Planning of Micro Airships*

Charles Blouin¹ and Eric Lanteigne¹ and Wail Gueaieb²

Abstract—The objective of this paper is to demonstrate the application of optimal control for generating dynamically constrained minimal time trajectories in micro unmanned airships. By design, airships are generally underactuated and underpowered limiting their maneuvering capabilities. Using a simplified dynamic model derived with experimentally derived coefficients, two trajectory planning simulations are solved using optimal control. The generated trajectories are then evaluated experimentally on a micro airship to demonstrate that optimal control can be used in open loop over short distances.

I. INTRODUCTION

Airships can perform numerous maneuvers such as move from one point to another in space, hover, loiter, take-off and land, or follow a target. Whether the airship is used for surveillance or payload delivery missions, energy efficiency is important to maximize flight time and distance covered. Consequently, a wide range of techniques have been developed to plan the path and control airships based on factors such as energy requirement, risk factor, and time associated to completion [1], [2]. In the case of a UAVs, the cost function is typically time, energy, safety of the path, coverage, or a combination of the previous factors, and the search space is typically restricted by the terrain, the maximum altitude reachable by the vehicle, exclusion zones due to airspace restrictions and air traffic, as well as a security buffer.

In general, path planning for UAVs can be separated into two categories: methods that respect differential constraints, and methods that do not. Differential constraints arise from the dynamic model, the maximum velocity and acceleration. Mathematically, adding differential constraints is equivalent to adding constraints on the derivative of the state vector. The methods not using differential constraints, such as network search algorithms on discretized search spaces can be used. Methods including the Markov decision process, A*, D*, or rapidly exploring random trees [3], [4] are typically faster and more suited for large scale problem, but can lead to unfeasible paths. Using differential constraints is computationally expensive and requires an accurate dynamic model, but has the advantage of generating optimized and precise trajectories [2].

Optimal control solvers have also been applied to the motion planning of airplanes [5]. In the case of [6], the

parameter being optimized is the time that the camera is pointed at the payload. Since an aircraft cannot stay in place, there will be time that the camera will not be pointed at the target, such as when the vehicle is banking to turn around. In another example, the performance index being minimized is the accumulated heat in the Apollo capsule during descent [7]. The ability of optimal control solvers to take into account wind velocity has been demonstrated in [8] where a collocation technique was used to increase the energy of the vehicle using wind gradients. Direct collocation, solves the optimal control problem by approximating the solution with polynomials and minimizing the differential errors. In direct collocation using a pseudo-spectral method, the solution is approximated with polynomial functions, which are also evaluated to check if they respect the dynamic constraints of the problem.

Optimal control solvers can also optimize the path of multiple vehicles by simply adding more states to the problem description and dynamic restrictions on those states [9]. Optimal control has recently been applied for the trajectory optimization of a simulated large airship over multiple kilometers [10]. Their airship model included wind tunnel data and CFD simulations of estimating the added mass from the mass of air displaced by the vehicle. The authors concluded that their optimization technique using optimal control resulted in a very realistic flight trajectory due to the precise airship model, the flight restriction, and the environment model used.

In many cases, a precise vehicle model is not available or impractical to obtain. In other cases, such as with light-than-air (LTA) vehicle, certain states of the linearized model may be prone to instability and thus difficult to control using conventional controllers. Moreover these types of vehicles are typically underactuated and underpowered making precise maneuvers challenging [11]. In this paper, it will be shown that complex and feasible trajectories for a very small LTA vehicle can be generated and tracked the Legendre pseudo-spectral optimal control method operating of a simplified 6 degree of freedom model with experimentally modeled drag and thrust coefficients. The paper also outlines a methodology for obtaining drag coefficient on any small lighter-than-air vehicles. The remainder of the paper is organized as follows. Section 2 outlines the development of a simplified vehicle and thruster model, and summarizes the state-space model equations used by the optimal control problem. Section 3 describes the general optimal control problem and its application to the trajectory generation of the airship. Section 4 presents two short-distance trajectory problems, and the resulting simulated and experimental trajectories. The

*This work was supported by NSERC Discovery grant RGPIN-2014-04501

¹Charles Blouin and Eric Lanteigne are with the Department of Mechanical Engineering, University of Ottawa, Ontario, Canada, K1N6N5 eric.lanteigne@uottawa.ca

²Wail Gueaieb is with the School of Electrical Engineering and Computer Science with the Department of Mechanical Engineering, University of Ottawa, Ontario, Canada, K1N6N5

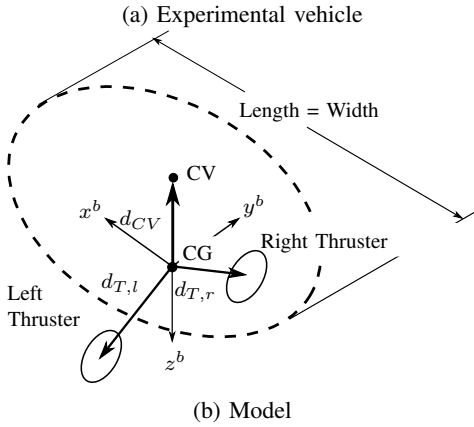


Fig. 1: Micro unmanned airship

paper concludes with a discussion on implementation, solver limitations and computation time in sections 5 and 6.

II. THEORY AND MODELING

Airships are typically modeled using Newton's equations in a local coordinate frame fixed to the center of volume (CV) of the vehicle. Since the solving time of the optimal control solver is dependent on the number of parameters to be computed, the airship is modeled around the center of gravity (CG) as less terms are required to in the problem formulation. An image and a model view of the airship is shown in Figures 1a and 1b. The micro airship is based on the Microblimp from Plantraco with a custom oblate spheroid helium envelop to increase lift due to the weight of the tracking marker. The helium envelop has a major radius of 50 cm and a minor radius of 30 cm. The nacelle is assembled on a single PCB and propulsion is provided by two forward-facing propellers driven by brushed DC micro-motors located at distance $d_{T,(y,z)} = [6 \ 5]$ cm from the center of gravity. Given the spheroid shape of the helium envelope, the airship was modeled as a six degrees of freedom rigid body with decoupled linear and rotational terms.

A. Kinematic equations

The optimal control solver is independent of the method used to describe the vehicle pose, however, minimizing the number of variables simplifies the problem by reducing the

TABLE I: Experimental parameters characterizing the airship

Variable	Value	95 % conf.
\bar{T}	0.010 N	
C_x	0.046	± 0.04
C_z	0.13	± 0.04
m_{ex}	0.077 kg	± 0.010
m_{ez}	0.117 kg	± 0.020
m_b	0.0357 kg	± 0.005
m_g	0.0258 kg	± 0.001
$m_b + m_g$	0.0565 kg	± 0.005
$\frac{m_{ex}}{m_b + m_g}$	1.4	± 0.2
$\frac{m_{ez}}{m_b + m_g}$	2.1	± 0.4
I_{ez}	$2.7 * 10^{-3} \text{ kg} \cdot \text{m}^2$	$\pm 0.2 * 10^{-3}$
I_{ex}	$6.0 * 10^{-3} \text{ kg} \cdot \text{m}^2$	$\pm * 10^{-3}$
C_{rz}	$2.7 * 10^{-4}$	$\pm 0.7 * 10^{-4}$
C_{rx}	$9.7 * 10^{-4}$	$\pm 2 * 10^{-4}$
d_p	2.1	± 0.6

size of the matrices and the memory allocated to solve the path. Given the inherent stability of airships, reaching singular positions is unlikely and Euler angles were adopted as opposed to quaternions since they can re.

The body reference frame is fixed to the CG of the airship and denoted with the superscript b . The inertial reference frame is denoted by the superscript i . The orientation of the vehicle is expressed using the roll (ϕ), pitch (θ), and yaw (ψ) convention. As such, the relation between the coordinates of the body reference frame to the fixed reference frame can be expressed as:

$$\mathbf{S}^{b \rightarrow i} \dot{\mathbf{x}}^b = \dot{\mathbf{x}}^i \quad (1)$$

where the homogeneous transformation matrix $\mathbf{S}^{b \rightarrow i}$ is given by:

$$\mathbf{S}^{b \rightarrow i} = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi s\theta + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2)$$

for velocities expressed as $\dot{\mathbf{x}} = [\dot{x}_x \ \dot{x}_y \ \dot{x}_z]^T$.

B. Kinetic equations

The airship is modeled as a rigid body with mass \mathbf{m} and inertia \mathbf{I} . If the linear and rotational terms are decoupled, the acceleration of the vehicle in the body reference frame $\ddot{\mathbf{x}}^b = [\ddot{x}_x^b \ \ddot{x}_y^b \ \ddot{x}_z^b]$ can be simplified to:

$$\ddot{\mathbf{x}}^b = \begin{bmatrix} \ddot{x}_x^b \\ \ddot{x}_y^b \\ \ddot{x}_z^b \end{bmatrix} = \begin{bmatrix} F_x^b/m_x \\ F_y^b/m_y \\ F_z^b/m_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} \dot{x}_x^b \\ \dot{x}_y^b \\ \dot{x}_z^b \end{bmatrix} \quad (3)$$

where m_x , m_y and m_z are the virtual masses along the body axes, and the product between the angular velocity vector $\boldsymbol{\omega} = [p \ q \ r]^T$ in the body reference frame and the linear velocity in the body reference frame corresponds to forces due to the non-inertial frame of reference. The input forces $\mathbf{F}^b = [F_x^b \ F_y^b \ F_z^b]^T$ are modeled around the CG in the body reference frame. Those input forces are due to the buoyancy force \mathbf{B} , the gravity \mathbf{G} , the aerodynamic drag and

lift \mathbf{D} , as well as the thrust from the propellers \mathbf{T} .

$$\mathbf{F}^b = \mathbf{S}^{i \rightarrow b} \mathbf{B}^i + \mathbf{S}^{i \rightarrow b} \mathbf{G}^i + \mathbf{D}^b + \mathbf{T}^b \quad (4)$$

In this work, the vehicle is assumed to be neutrally buoyant, therefore the buoyancy force is equal and opposed the gravity force. The drag force \mathbf{D}^b is proportional to the relative vehicle velocity $\dot{\mathbf{x}}_r^b = \dot{\mathbf{x}}^b - \mathbf{S}^{i \rightarrow b} \dot{\mathbf{x}}_w^i$ of the vehicle velocity in the body reference frame with respect to the wind velocity $\dot{\mathbf{x}}_w$. In this paper the linear drag is simplified to:

$$\mathbf{D}^b = C(\phi, \theta, \psi, \dot{\mathbf{x}}_w) \dot{\mathbf{x}}_r^b |\dot{\mathbf{x}}_r^b| \quad (5)$$

where the drag coefficient $C(\phi, \theta, \psi, \dot{\mathbf{x}}_w)$ depends on the angle of attack of the airship and the wind velocity. The drag coefficient for the airship in Figure 1a was modeled experimentally by measuring the response to a known input using the vision tracking system; the complete procedure of which is described in [13]. Lastly, the thrust vector $\mathbf{T} = [T_{left} + T_{right} \quad 0 \quad 0]^T$ is the sum of the thrust from left and right thrusters.

Similar to the input forces, the input moments \mathbf{M}^b are modeled around the center of gravity in the body reference frame. The moments acting on the vehicle include those due to the rotational drag \mathbf{D}_r^b , the offset between the CV and the CG, and the thrust forces.

$$\mathbf{M}^b = \mathbf{d}_{CV} \times \mathbf{S}^{i \rightarrow b} \mathbf{B}^i + \mathbf{d}_D \times \mathbf{D}^b + \mathbf{M}_D^b + \mathbf{M}_T^b \quad (6)$$

where \mathbf{d}_{CV} is the distance vector between the center of gravity and the CV and \mathbf{d}_D is the distance between the center of gravity and the center of drag. The rotational drag is given by $\mathbf{M}_D^b = [C_{d_{rx}} p^2 \quad C_{d_{ry}} q^2 \quad C_{d_{rz}} r^2]$ with experimentally determined drag coefficients. The methodology used to determine these coefficients is described in [13].

In the simulation, the center of volume was approximated to be at the same position as the center of drag. The variable $\mathbf{M}_T^b = \mathbf{d}_{T,l} \times [T_{left} \quad 0 \quad 0]^T + \mathbf{d}_{T,r} \times [T_{right} \quad 0 \quad 0]^T$ is the sum of moments caused by the thrust forces from the thrusters.

The vehicle's angular acceleration in the body frame is then given by:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\mathbf{M}^b - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}) \quad (7)$$

The rate of change of the Euler angles can then be obtained from the angular rate of change in the body frame from:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8)$$

Equations (1), (3), (7) and (8) represent the state-space model that will be used by the optimal control solver. Further details on airship dynamics and modeling can be found in [10], [14], [15], [16].

C. Thrust model

There is non-negligible lag between the command and the thrust relative to the simulation time. Therefore a thruster model was developed for the Microblimp. The model includes the inertia of the motor and propeller J_p , the angular

velocity Ω , the propeller torque τ , a damping constant D_p due to the drag of the air, and a constant C_p that relates the square of the angular velocity of the propeller and the thrust [17]. The governing equations are

$$J_p \dot{\Omega} = \tau - D_p \Omega^2 \quad (9)$$

$$T = C_p \Omega^2 \quad (10)$$

Measuring the velocity of brushed micro motors directly is difficult. Since the dominant frequency of the sound produced by the motor is proportional to the rotational velocity, the normalized velocity profile Ω_n can be acquired by measuring and analyzing the frequency spectrum plot of multiple short sound samples of the motor accelerating and decelerating. The results were fitted with theoretical results of the differential equations as seen in Figure 2 to obtain the normalized constants D_p . The drag constant D_p is normalized by setting the inertia J_p to 1. The normalization can be done since only the propeller velocity normalized with respect to the maximum propeller velocity is needed to model the airship, as we are looking to establish the relationship between the input and the motor thrust. The absolute propeller velocity will not affect the result, as the thrust is proportional to the square of the propeller velocity using the constant C_p , and this constant is computed experimentally using the nominal thrust \bar{T} .

The damping constant D_p was computed first since no torque is applied during deceleration and (9) reduces to $\dot{\Omega}_n = -D_p \Omega_n^2$. The sound of the motor responding to an inverse step input of full throttle to zero throttle was recorded. The solution to the deceleration is the normalized velocity $\Omega_n(t)$:

$$\Omega_n(t) = \frac{1}{D_p t + c_1} \quad (11)$$

and the solution to the general model (9) is:

$$\Omega_n(t) = \frac{\sqrt{\tau} \tanh(\sqrt{\tau} \sqrt{D_p} c_1 + \sqrt{\tau} \sqrt{D_p} t)}{\sqrt{D_p}} \quad (12)$$

where the torque τ is the input command sent to the DC motor. The value of D_p is found by fitting this function in (12) to the experimental data. The precision of the model decreases significantly at low speed. However its effects are negligible since the thrust produced at low speeds is very small.

III. OPTIMAL CONTROL IMPLEMENTATION

The following section summarizes the governing equation of general optimal control problem and provides details on its implementation for the trajectory planning of a micro airship.

Mathematically, the general optimal control problem is defined as follows: given the state trajectories $\mathbf{s}(t)$, $t \in [t_0, t_f]$, and times t_0 and t_f , find the optimal control trajectory $\mathbf{u}(t)$, $t \in [t_0, t_f]$ that minimizes the cost function:

$$G = \Phi(\mathbf{s}(t_0), \mathbf{s}(t_f), t_0, t_f) + \int_{t_0}^{t_f} L(\mathbf{s}(t), \mathbf{u}(t), t) dt \quad (13)$$

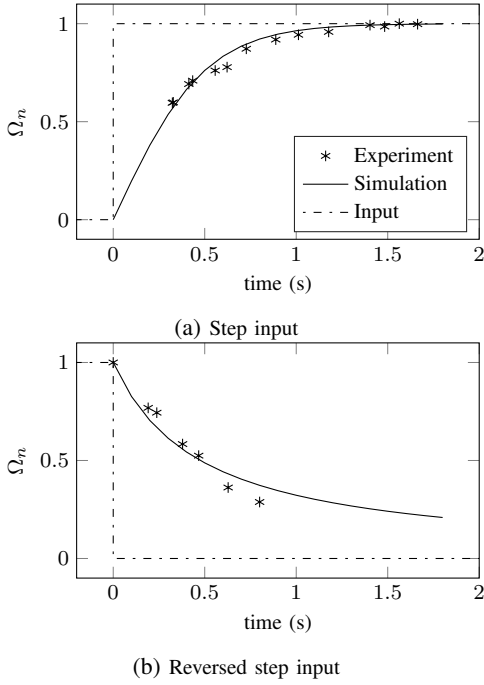


Fig. 2: Experimentally normalized response of a thruster

where the function Φ is the cost associated to the initial and final starting points, and L is the cost associated to an infinitesimal section of the path. The state vectors $\mathbf{s}(t)$ and the control vector $\mathbf{u}(t)$ have the following form:

$$\mathbf{u}(t) = [u_1(t) \ u_2(t) \ \dots]^T = [\tau_R \ \tau_L]^T \quad (14)$$

$$\begin{aligned} \mathbf{s}(t) &= [s_1(t) \ s_2(t) \ \dots]^T \\ &= [x^i(t) \ y^i(t) \ z^i(t) \ \dot{x}_x^b(t) \ \dot{x}_y^b(t) \ \dot{x}_z^b(t) \ \dots \\ &\quad \dots \ p(t) \ q(t) \ r(t) \ \phi(t) \ \theta(t) \ \psi(t) \ \dots \\ &\quad \dots \ \Omega_{nL}(t) \ \Omega_{nR}(t)]^T \end{aligned} \quad (15)$$

The components of the control vector $\mathbf{u}(t)$ are the controls sent to the two motors. The states $\mathbf{s}(t)$ of the airship are position, velocity, Euler angles, and normalized motor velocity. The states have dynamic constraints due to Newton's equations therefore the derivatives of the vehicle's states are functions of the vehicle's states, control inputs, and time, as expressed in the following equation:

$$\dot{\mathbf{s}}(t) = f(\mathbf{s}(t), \mathbf{u}(t), t), t \in [t_o, t_f] \quad (16)$$

Typically, there are maximum and minimum values associated to the control inputs, the states, the parameters, the start time and the end time. These bound constraints are given by (17) to (21).

$$\text{Control:} \quad \mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u, t \in [t_o, t_f] \quad (17)$$

$$\text{States:} \quad \mathbf{s}_l \leq \mathbf{s}(t) \leq \mathbf{s}_u, t \in [t_o, t_f] \quad (18)$$

$$\text{Start time:} \quad t_{0l} \leq t_0 \leq t_{0u} \quad (19)$$

$$\text{End time:} \quad t_{fl} \leq t_f \leq t_{fu} \quad (20)$$

$$\text{Total time:} \quad t_f - t_0 \geq 0 \quad (21)$$

The subscript u is for the upper limit and the subscript l is for the lower limit. The initial and final values of the problem can also be specified. For instance, the vehicle may have to start in a certain configuration, and the starting and ending values may be specified.

A. Description of the solver used and its methods

The process of transforming an optimal control problem into a non-linear optimization problem is well described in [18] and is briefly summarized here. The first part of this section describes how to approximate smooth functions using polynomials. The second part expresses the optimal control problem as a non-linear optimization problem.

If the states and the control inputs of the system, which are functions of time, are represented by polynomials, the problem can then be discretized in N points Π_k , and the value of the states and controls can be interpolated between those points using a special case of Lagrange polynomials: the Legendre polynomials. Legendre polynomials are orthogonal to each other and they are the solutions to the Sturm-Liouville, which is a differential equation problem. The n th degree Legendre polynomial is defined explicitly as:

$$P_n(\Pi) = \frac{1}{2^n n!} \frac{d^n}{d\Pi^n} [(\Pi^2 - 1)^n] \quad (22)$$

The Legendre polynomials have the property of being very fast to integrate in the $[-1, 1]$ interval using the Legendre-Gauss quadrature which states:

$$\int_{-1}^1 P_n(\Pi) \approx \sum_{i=1}^N w_i P_n(\Pi_i) \quad (23)$$

In summary, the integral of a function can be approximated by a weighted sum of this function evaluated at the nodes Π_i , which are the roots of the Legendre polynomial. The weights w_i are calculated only once using the Gauss-Legendre quadrature, which makes the integral very fast to perform. Note that for the rest of the problem, the functions of time t will be mapped to Π , where the start time t_0 and the end time t_f will become -1 and 1, respectively. The transformation used is the following:

$$\Pi \leftarrow \frac{2}{t_f - t_0} t - \frac{t_f + t_0}{t_f - t_0} \quad (24)$$

Any node Π could be chosen to represent the discretized function. However, for numerical stability reasons explained in [18], the Legendre-Gauss-Labatto (LGL) nodes will be used. The LGL nodes are more precise than equidistant nodes. In general, the coefficient of the spectral decomposition of a function $P_n(\Pi)$ over a certain interval is performed by integrating $P_n(\Pi)$ multiplied by the spectral functions. It is also possible to differentiate quickly the function approximated by a Legendre polynomial using a differentiation matrix as shown in (25) [18].

$$\dot{P}_n(\Pi_k) \approx \sum_{i=0}^N D_{ki} P_n(\Pi_i) \quad (25)$$

Consequently, the function to be optimized can be rewritten as

$$G = \phi(\mathbf{s}(-1), \mathbf{s}(1), t_0, t_f) + \frac{t_f - t_0}{2} \int_{-1}^1 L(\mathbf{s}(\Pi), \mathbf{u}(\Pi), \Pi) dt \quad (235)$$

$$\approx \phi(\mathbf{s}^N(-1), \mathbf{s}^N(1), t_0, t_f) + \dots$$

$$\frac{t_f - t_0}{2} \sum_{k=0}^N [L(\mathbf{s}^N(\Pi_k), \mathbf{u}^N(\Pi_k), \Pi_k)] \omega_k dt \quad (26) \quad (240)$$

In the previous equation, \mathbf{s}^N and \mathbf{u}^N are the states and the control inputs approximating the exact solution with Legendre polynomials of order N . For smooth functions, the integral can be approximated with very few points accurately. In the case of an airship, the performance index G will typically be the time, the power required, or a combination of both, such as in [19].

The solver will minimize the performance index while respecting the differential constraints imposed on \mathbf{s}^N , $\dot{\mathbf{s}}^N$ and \mathbf{u}^N .

The vector representing the problem is \mathbf{y} :

$$\mathbf{y} = [\mathbf{u} \quad \mathbf{s} \quad t_0 \quad t_f]^T \quad (27)$$

Finally, the minimization problem is solved by the non-linear optimizer IPOPT [20]. The non-linear optimizer transforms the problem into a linear algebra problem.

$$\min_{\mathbf{y}} G(\mathbf{y}) \quad (28)$$

subject to:

$$H_l < H(\mathbf{y}) < H_u \quad (29)$$

$$\mathbf{y}_l < \mathbf{y} < \mathbf{y}_u \quad (30)$$

The vector $H(\mathbf{y})$ is composed of the events, the path constraints, and the time $t_f - t_0$. The results of the optimization are the state vector \mathbf{y} , which contains the optimized control inputs and states, expressed as Legendre polynomials, the start time t_o and the end time t_f . The solution is not guaranteed to be the absolute minimum, but it is a local minimum that respects the constraints set on the system and the specified tolerances. The process of solving the problem is shown in the diagram of Figure 3.

IV. EXPERIMENTAL PLATFORM

For purpose of evaluating the use of optimal control for the trajectory planning of the micro airship, the performance index is set to time only ($G = t_f$) with $t_o = 0$. To demonstrate the feasibility of the trajectories generated by the solver, experiments are conducted on a real airship (see Figure 1a) by sending the commands computed by the solver directly and comparing the resultant trajectories. The airship's path is tracked using the ARUCO software from the Aplicaciones de la Visión Artificial research group at the University of Córdoba, Spain. The software tracks a 25 cm by 25 cm marker using a Microsoft Livecam running in 640x480 pixel mode.

To evaluate the effectiveness of the pseudo-spectral optimal control method, the micro airship was subjected to the control vector computed by the solver for two types of trajectories: linear, and rotational.

A. Linear trajectories

For the linear trajectory tests, the constraints imposed on the solver are that the vehicle starts immobile and must move forward by 1 m. The final roll and pitch are not considered important for this test and are considered "free" and can take any value. Their upper and lower limits were set a large value (± 10 rad) to relax their constraints in the solver.

Without the propeller inertia, the simulated solution would be very close to a simple bang-bang control, with a full forward thrust followed by a full reverse thrust. However, the simulated trajectory generated by the solver, shown in Figure 4, illustrates that the propeller inertia is non-negligible. The circle and cross on the solid line represents the position of the airship at $t_{f/2}$ and t_f , where t_f is the total time of the maneuver, while the arrows represent the relative magnitude of the thrust commands along the trajectory at which the thrust is applied. The length of the arrows represents the magnitude of the thrust, while the orientation of the arrow represents the forward or reverse direction of the thrust. An arrow is drawn at the location of the thruster at time $N \times t_f/10$, with $N = 0, 1, \dots, 10$. According to the optimizer, the solution is a full thrust forward command followed by a full thrust reverse command, and, finally, a short full forward thrust command near $x^i = 1$ to arrive at the goal with zero propeller velocity.

The commands generated by the solver were sent to the vehicle and the trajectory was tracked. Figure 5a shows the position of the vehicle at 50% and 100% of the experiment time. On average, the position error at 50 % time, after 1.9 seconds, is 3 % in the x direction and 6 % in the y direction, where the position error is expressed as a percentage of the total distance traveled at each time. At the end of the 3.8 seconds experiment, the error is 7 % in the x direction and 15 % in the y direction. The heading of the vehicle is shown in Figure 5b. The final orientation of the vehicle was positive for all tests indicating that the right motor was stronger than the left motor. Unfortunately, due to the non-linear behavior of the commands, calibrating the motors proved difficult. However, the results demonstrate consistent trajectories.

B. Rotational trajectories

For the rotational trajectory tests, the constraints imposed on the solver are that the vehicle starts immobile and must rotate by 180° . Once again, the final roll and pitch are considered "free" as their final values have little effect on the intended maneuver. The simulated thrust and position of the airship are shown in Figure 6. In this case, the two thrusters rotate the vehicle without inducing lateral motion. Flight tests were subsequently performed and the results, shown in Figure 7a, demonstrated excellent repeatability in the final yaw orientation. The standard deviation at the end of the test was 3.7° , while the average error was 18° . The vehicle

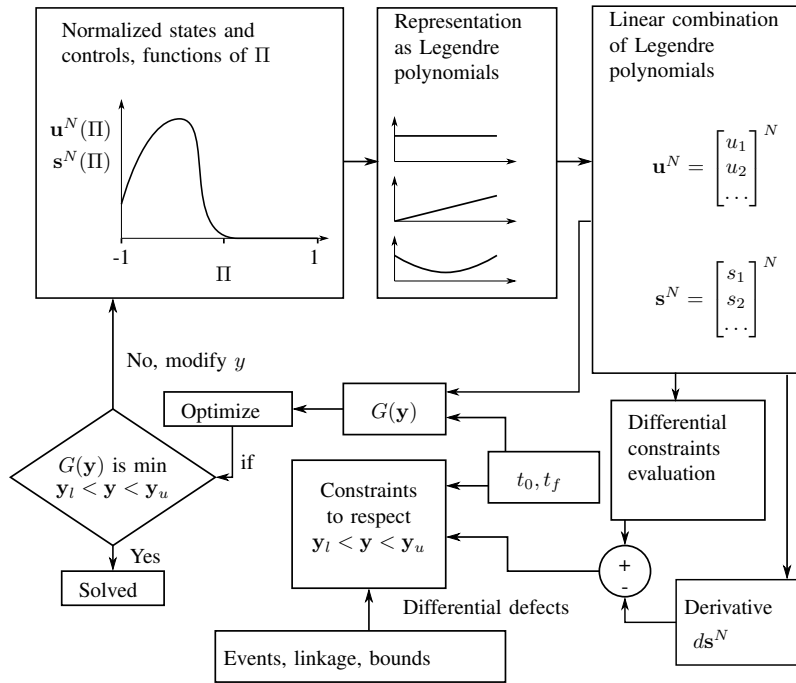


Fig. 3: Optimal control problem simplified flowchart

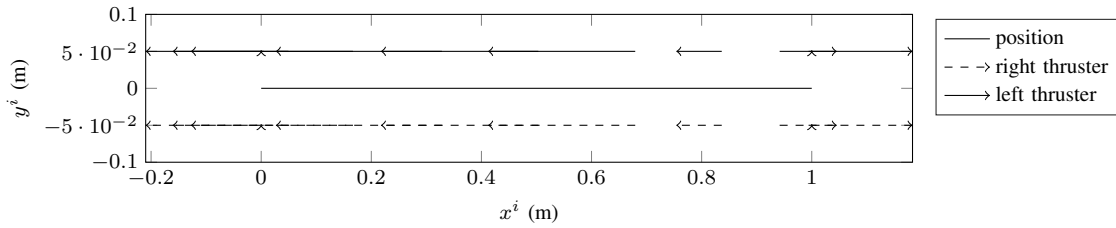


Fig. 4: Simulated thrust for a linear trajectory. The 11 arrows for the right and left thrust illustrate the relative thrust levels at equidistant times during the trajectory

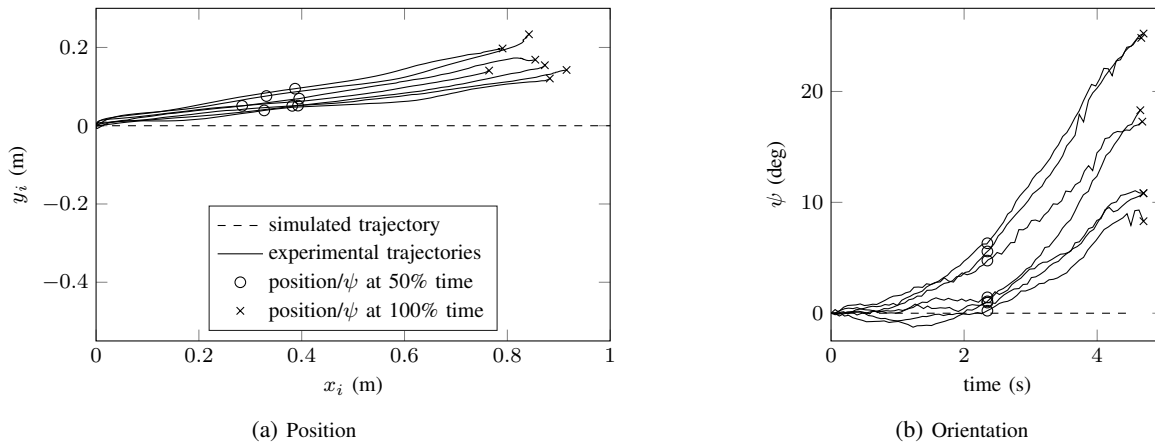


Fig. 5: Experimental linear trajectories

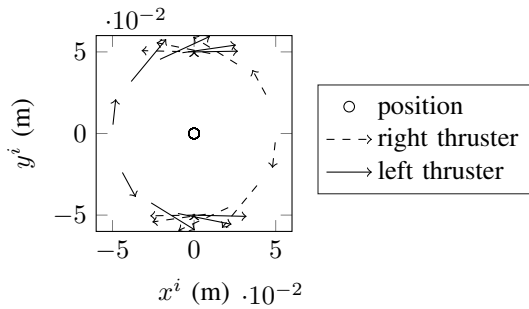


Fig. 6: Simulated thrust and trajectory for the rotational trajectory

consistently overshoots, which implies that the inertia or the rotational drag coefficient or both were overestimated. As a consequence, the position of the airship generally drifted to the lower left corner of its initial position over time as shown in Figure 7b. Since the command sent to each motor was the same, but inverted, the drift in position shows that each motor had a different response to the input. Additionally, the positional error may be due to the initial velocity being slightly nonzero and unmodeled aerodynamic effects such as wind.

V. DISCUSSION

A. Sources of error

The airship and thruster models are not perfectly accurate; consequently, the vehicle exhibits drift in open-loop. Errors in the modeling will cause differences in the acceleration time, the time of arrival, the terminal velocity and other characteristics of the airship. In addition, some of the model assumptions were simplified. For instance, with real propellers, the thrust is a function of the airspeed, which was neglected in the model. The drag is based on a very simple model, not taking into account the angle of attack or the change in Reynold number when the velocity increases, which means that the drag is not perfectly proportional to the square of the velocity. Since the airship is not a rigid hull, there are also deformations in the helium envelop, particularly at the location of the gondola mount.

Errors were also induced by the measurement system. The low camera resolution limited the precision of depth measurements. To mitigate this limitation, the camera was placed directly above the vehicle to reduce the noise and vehicle altitude, computed from the size of the marker in the field of view, was not measured. Errors were also induced by atmospheric disturbances due to ventilation, convection, and movement was observed in the test area.

B. Limitations

The convergence of optimal control solvers are a known issue due to the very wide search space and the quality of the initial guess. To ensure convergence, the two experiments were resolved in multiple steps. The solver initially interpolates between the initial and final states to create an initial guess for every state, with the controls set to

zero. The solver then finds a solution with a low number of points, which corresponds to a low order of Legendre polynomials. The low resolution solution is then used as a guess for a higher order of polynomial approximation. The process is repeated until the desired resolution is achieved. In the case of the two experiments presented in this work, the solver started with 8 nodes, followed by 9, 15, 19 and 25 nodes. This proved sufficient, however a refined guess may be necessary in more complex cases, especially if the solution diverges significantly from the guess. Part of the convergence issues were due to the sharp discontinuities in the solutions. Attempts to solve directly with 25 nodes failed. Consequently, to use a pseudo-spectral solver on a vehicle, a method for generating an initial guess would be required and, if the optimal solver does not converge, this guess could be used by the path follower as a less optimal solution. An example would be to use rapidly exploring random trees to generate a rough possible path and initial guess for the optimal control solver [9].

Polynomials, while very accurate for continuous solutions, will show artifacts when approximating discontinuous changes, such as a the bang-bang control seen in the linear trajectory. It has been proven in [21] and shown here that convergence is possible even in the presence of discontinuities. For a more accurate solution, several phases could be used as described in [18]. Since each phase does not require continuity, the solution could be approximated with the discontinuities properly modeled.

C. Comparison of simulation and experimental results

Open loop control was adopted to test the accuracy of the model and evaluate the performance of the path planner. All experimental trajectories drifted to the left due to a difference in the right and left motor thrust and due to the slight aerodynamic asymmetry of the helium envelope. The motor commands could be weighed to match the thrust of each motor or a feedback controller could directly correct for the error in the model. Additionally, a more detailed propeller model taking into account the inertia and the drag of the propeller could be added without changing the problem formulation.

D. Solution time

The solver required between 1 s and 30 s on an Intel i7 2600 processor to generate the trajectories. The time to solve the problems varied significantly as a function of the initial guess, the number of nodes, the number of mesh refinement, the initial and final conditions and tolerance accepted. It can be affected due to technical details of implementation as many code libraries are used by the solver. In the experiments performed, it was found that the number of nodes was the most influential parameter affecting the time to solve the problem. A poor initial guess also affected the time of convergence for the first step of solving. It is possible to change the tolerance accepted by the solver, which may improve the convergence time. However, the tolerance accepted has

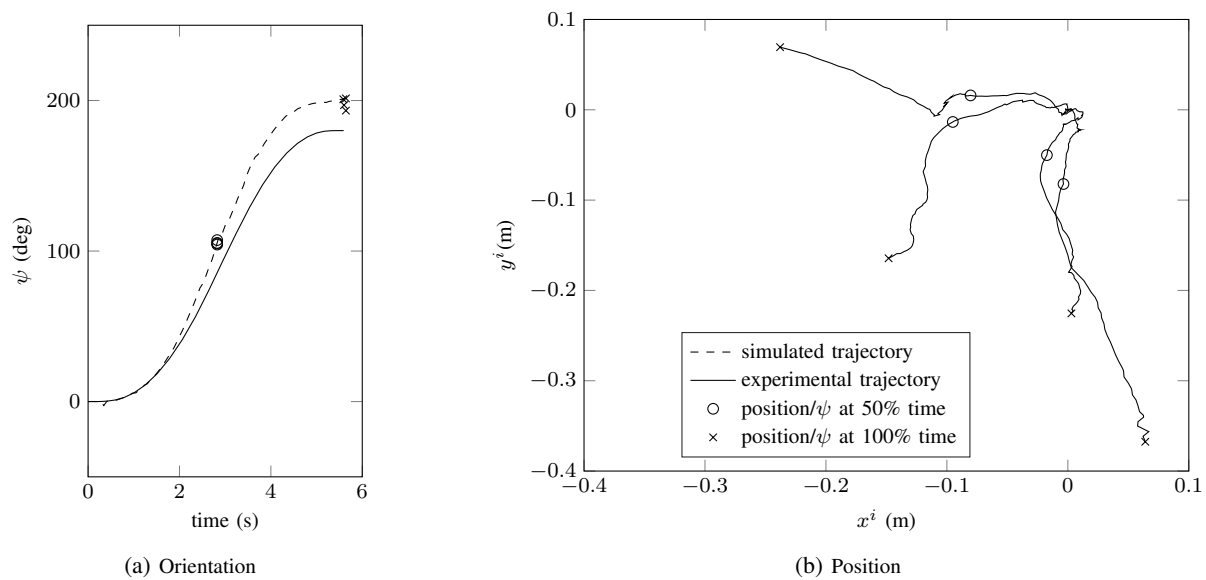


Fig. 7: Experimental rotational trajectories

a relatively small impact due to the exponential rate of convergence of the pseudo-spectral method [18].

VI. CONCLUSIONS

Optimal control solvers can be used to generate very complex locally optimal paths while respecting the differential equations of the system. While trajectory optimization using optimal control was presented previously, the result presented in this paper are the first instance illustrating the application of this method to a real LTA with limited maneuverability doing complex maneuvers. Given a more robust model, the solver could be used to pre-calculate more complex open-loop maneuvers such as parts of the landing or take-off procedures.

REFERENCES

- [1] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: A survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 1, pp. 36–44, 2010.
- [2] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 57, no. 1-4, pp. 65–100, 2010.
- [3] W. H. Al-sabban, L. F. Gonzalez, R. N. Smith, and G. F. Wyeth, "Wind-Energy based Path Planning For Electric Unmanned Aerial Vehicles Using Markov Decision Processes," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012, pp. 1–6.
- [4] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristic-based path planning," in *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems*, 2005, pp. 1–10.
- [5] S. John, S. Morello, and H. Erzberger, "Application of trajectory optimization principles to minimize aircraft operating costs," *18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, vol. 18, pp. 415–421, 1979.
- [6] B. R. Geiger, J. F. Horn, A. M. Delullo, and L. N. Long, "Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming," in *American Institute of Aeronautics and Astronautics GNC conference.*, vol. 2, 2006, pp. 1–13.
- [7] O. von Stryk and R. Burlirsch, "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, vol. 37, pp. 357–373, 1992.
- [8] T. C. Flanzer, G. C. Bower, and I. M. Kroo, "Robust Trajectory Optimization for Dynamic Soaring," in *AIAA Guidance, Navigation, and Control Conference*, 2012, pp. 1–22.
- [9] K. Mohan, M. Patterson, and A. V. Rao, "Optimal Trajectory and Control Generation for Landing of Multiple Aircraft in the Presence of Obstacles," *Guidance, Navigation, and Control Conference*, pp. 1–16, August 2012.
- [10] S. Lee and H. Bang, "Three-Dimensional Ascent Trajectory Optimization for Stratospheric Airship Platforms in the Jet Stream," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1341–1351, sep 2007.
- [11] E. Lanteigne, A. Alsayed, D. Robillard, and S. Recoskie, "Modeling and control of an unmanned airship with sliding ballast," *Journal of Intelligent Robotic Systems*, 2017.
- [12] C. Blouin, W. Gueaieb, and E. Lanteigne, "Trajectory optimization of a small airship in a moving fluid," in *CCToMM Symposium*, 2015, pp. 1–12.
- [13] Y. Liu, Y. Zhang, and Y. Hu, "Optimal path planning for autonomous airship based on clonal selection and direct collocation algorithm," in *Proceedings of 2008 IEEE International Conference on Networking, Sensing and Control, ICNSC*, 2008, pp. 1828–1832.
- [14] S. B. V. Gomes and J. J. R. Ramos, "Airship dynamic modeling for autonomous operation," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, vol. 4, May 1998, pp. 3462–3467.
- [15] D. T. Liesk and P. M. Nahon, "An investigation of the equations of motion of an airship in a wind field," *Control*, pp. 24–29, July 2011.
- [16] D. R. Yoerger, J. G. Cooke, and J.-J. E. Slotine, "The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 3, pp. 167–178, 1990.
- [17] V. M. Becerra, "PSOPT Optimal Control Solver User Manual (Release 3)," p. 421, 2011.
- [18] S. Recoskie, A. Fahim, W. Gueaieb, and E. Lanteigne, "Hybrid power plant design for a long-range dirigible UAV," *IEEE/ASME Transactions on Mechatronics*, vol. 19, pp. 606–614, 2014.
- [19] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [20] W. Kang, Q. Gong, I. M. Ross, and F. Fahroo, "On the convergence of nonlinear optimal control using pseudospectral methods for feedback linearizable systems," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 14, pp. 1251–1277, 2007.