Abstract: As research into UAVs accelerates into the 21st century, alternatives to fixed wing vehicles such as the quadrotor are causing interest. The quadrotor is a small agile vehicle which could be suitable for search and rescue, surveillance and remote inspection. For autonomous operation a control system that incorporates both trajectory planning and trajectory following is required. Trajectory planning can be posed as a constrained optimization problem typically within the control space and with some constraints being placed in the output space. However, differential flatness enables the optimization to occur within the output space and therefore simplifies the problem. A parameterization of the output is required to reduce the problem to a finite dimensional problem, this can be done using any number of techniques. Trajectory following can be achieved using linear multi-variable control techniques such as LQR control.

Keywords: Quadrotor, Optimal trajectory planning, Laguerre polynomials, Differential flatness, Trajectory following, LQR.

1. INTRODUCTION

Unmanned air vehicles (UAVs) have attracted considerable interest in recent times. Fixed wing vehicles have had extensive applications for military and meteorological purposes. The quadrotor is a small agile vehicle controlled by the rotational speed of the four rotors. It benefits from having very few constraints on motion and an ability to carry a high payload. Furthermore, with the use of ducted fans instead of prop rotors it would be safe for internal flight. The low cost and simplicity mean the quadrotor provides an excellent testing ground for application of advanced control techniques on UAVs.

In order to achieve full autonomy a controller has to incorporate trajectory planning and trajectory following which is often regarded as a separate issue (Richards and How, 2002). The path planning problem can be simplified by exploiting the differential flatness of the vehicle dynamics. Differential flatness (Fleiss et al., 1992) enables the optimization to occur within the output space, as opposed to the control space. This technique has been considered for air vehicles (Martin et al., 1994) and applied to a helicopter (Koo and Sastry, 1999). Differential flatness has also been considered with respect to the quadrotor to achieve a convergent tracking controller (Driessen and Robin, 2004). Trajectory following can be achieved using a LQR controller to follow a time dependent reference trajectory $x_{ref}$. In this paper, non-linear optimization is used to determine the optimal trajectories for the quadrotor to execute some simple missions. An LQR controller is then used to follow the generated trajectory. By means of simulation, it is shown that the quadrotor follows the optimal trajectory, despite wind and other perturbations.

This paper is divided into 3 sections, the first discusses closed loop control and application of LQR to follow a reference trajectory. The second section derives the differential flatness and formulates the trajectory planner. Finally application of these techniques requires a output space parameterization which is considered
within the final section. The controller is then validated using a full simulation model of the quadrotor.

2. LQR TRAJECTORY FOLLOWING

In the standard LQR control problem, for a linear state-space model of the plant dynamics

\[ \dot{x}(t) = Ax(t) + Bu(t), \]

a control input

\[ u(t) = u_{\text{ref}} - K_c x(t) \]

is determined such that the closed loop system

\[ x = [A - BK_c]x(t) \]

is stable and a performance measure where

\[ J = \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) \, dt \]

is minimized where \( Q \) and \( R \) are weighting matrices.

Assuming a time dependent state reference trajectory \( x_{\text{ref}}(t) \), LQR control can be applied as a trajectory follower to minimize small errors between the full measured state \( x \) and the reference state \( x_{\text{ref}} \), such that the applied control is

\[ u(t) = u_{\text{ref}} - K_c (x(t) - x_{\text{ref}}(t)). \]

3. DYNAMIC MODEL

![Image of Quadrotor Schematic](image)

Fig. 1. Quadrotor schematic

The quadrotor is actuated only by independently varying the speed of the four rotors. A pitch moment is achieved by varying the ratio of the front and back rotor speeds, a roll by varying the left and right rotor speeds (see Figure 1). A yaw moment is obtained from the torque resulting from the ratio of the clockwise (left and right) and the anti-clockwise (front and back) speeds. The actuator outputs, \( \tilde{u}_i \), are hence functions of normalized individual thrusts and torques:

\[ \tilde{u}_1 = \frac{(t_1 + t_2 + t_3 + t_4)}{} \]

\[ \tilde{u}_2 = \frac{t_1 + t_3 + t_4}{l} \]

\[ \tilde{u}_3 = \frac{(t_1 - t_2)}{} \]

\[ \tilde{u}_4 = \frac{(u_1 + u_2 - u_3 + u_4)}{} \]

where \( u_i \) and \( t_i \) are respectively the normalized torques and normalized thrusts from the \( i \)th rotor, and \( l \) is the distance of the rotor from the center of mass.

The state variable vector, \( x \), is defined as

\[ x^T = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \phi \ \theta \ \psi] \]

where \( x, y \) and \( z \) are the translational positions (see Figure 1) and \( \phi, \theta, \psi \) are the roll, pitch and yaw respectively. In order to determine the differential flatness that will be discussed in section 4.2 it is necessary to define four outputs. These are chosen as the translational positions, \( x, y, z \), and the yaw angle, \( \psi \). The yaw angle is chosen because it can be dynamically decoupled from the other states (Castillo et al., 2004). The output vector, \( y \), is hence defined as

\[ y^T = [x \ y \ z \ \psi]. \]

The control vector, \( u \), is defined as

\[ u^T = [u_1, u_2, u_3, u_4] \]

where \( u_1 = \tilde{u}_1 \), \( u_2 = \tilde{u}_2 \), \( u_3 = \tilde{u}_3 \) and \( u_4 = \tilde{u}_4 \).

The control input \( u_1 \) can be expressed as

\[ u_1 = \sqrt{x^2 + y^2 + (g - z)^2} \]

The actuator outputs can be expressed as functions of the control and its integrals

\[ \begin{bmatrix} \dot{\tilde{u}}_2 \\ \dot{\tilde{u}}_3 \\ \dot{\tilde{u}}_4 \end{bmatrix} = \begin{bmatrix} c_w c_\theta c_\phi & s_w c_\theta & -s_\phi \\ -s_w & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \]

\[ + \begin{bmatrix} -s_w c_\phi c_\theta & c_\phi c_\theta & -c_\theta \\ -c_\phi & -s_\phi & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \]

where \( c_\theta = \cos(\theta), s_\theta = \sin(\theta) \), etc.

From (13) we get the translational equations of motion

\[ \dot{x} = u_1(s_\theta c_\phi) \]

\[ \dot{y} = u_1(-s_\theta) \]

\[ \dot{z} = g - u_1(c_\theta c_\phi) \]

Taking the state equation (10) and the translational equations of motion (15) – (17), a state space model can be defined as

\[ \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ u_1(s_\theta c_\phi) \\ u_1(-s_\theta) \end{bmatrix} \]

where \( c_\theta = \cos(\theta), s_\theta = \sin(\theta) \), etc.
include physical constraints, actuator constraints and output space and the state space. These constraints formed subject to some constraints placed within the control space is performed to allow optimizations to occur within the control space. This is typically a optimization within the control space is performed subject to some constraints placed within the output space and the state space. These constraints include physical constraints, actuator constraints and output, g.

3.1 Stability analysis

The control gains \( K_c \) were designed with the plant linearized at hover with

\[
Q = (1 \times 10^{-5})I \quad (19)
\]

\[
R = \text{diag}(1 \times 10^{-5}, 1 \times 10^9, 1 \times 10^8) \quad (20)
\]

The weighting matrices \( Q \) and \( R \) were chosen to ensure that the actuator constraints would be maintained. Clearly, to follow a trajectory, the system does not remain at hover. A simplified analysis is therefore performed to determine an envelope of operation where the vehicle will remain stable. The analysis is not rigorous and is hence only an indicator, however the analysis is simple and provides a convex bound on the state \( x \). Stability for a calculated trajectory can be subsequently checked by simulation. From (18), the linearized dynamics depend on three variables, \( \theta, \phi \) and \( u_1 \). We define the linearized stability set \( S \) to be

\[
S = \{ \theta, \phi : \alpha(A(\theta, \phi, u_1) - B(\theta, \phi)K_c) < 0, \ 0.5 < u_1 < u_1\text{max} \} \quad (21)
\]

where \( \alpha(\cdot) \) is the spectral abscissa (most positive real part of the eigenvalues). The set is plotted in Figure 2. By inspection, we can fit a disk inside the set, hence it is clear that \( S_c \subset S \) where

\[
S_c = \{ \theta, \phi : \theta^2 + \phi^2 \leq r^2 \} \quad (22)
\]

with \( r = 48^\circ \). \( S_c \) is also shown in Figure 2. An extra constraint can be inserted into the trajectory planner, which maintains the angles within this set and therefore ensures linearized time-invariant stability.

4. TRAJECTORY OPTIMIZATION

4.1 Problem formulation

To determine the optimal reference trajectory, typically a optimization within the control space is performed subject to some constraints placed within the output space and the state space. These constraints include physical constraints, actuator constraints and obstacle avoidance constraints. For example, the problem could be posed as

\[
\min \Phi \text{ for } t \in [0, T],
\]

s.t. \( c_u(\mathbf{u}) \leq 0 \)

s.t. \( x_0 - g_1(\mathbf{u}_0) = 0 \)

s.t. \( y_T - g_2(\mathbf{u}_T) = 0 \)

s.t. \( y = g_3(\mathbf{u}, x_0) \)

where \( \Phi \) is the cost function, \( c_u(\mathbf{u}) \) is a set of functions that express inequality constraints on the state and output, \( x_0 \) is the initial state at \( t = 0 \), the state is a function \( g_1 \) of the input, \( y_T \) represents the terminal output at \( t = T \), the output is some a function \( g_2 \) of the input and some dynamic constraints are applied so that the output \( y \) is a function \( g_3 \) of the input \( \mathbf{u} \) and the initial state \( x_0 \).

4.2 Differential flatness

Differential flatness is the expression of the state and control vectors in terms of the output vector (Fleiss et al., 1992). For a system to be differentially flat and therefore possessing a flat output it requires (Chelouah, 1997) a set of variables such that;

(1) the components of \( y \) are not differentially related over \( \mathbb{R} \);

(2) every system variable may be expressed as a function of the output \( y \);

(3) conversely, every component of \( y \) may be expressed as a function of the system variables and of a finite number of their time derivatives.

By manipulation of the equation of motion and recalling (13)-(14), the state vector and input vector can be expressed as a function of the output vector.

\[
\theta = \arctan \left( \frac{x}{g - z} \right) \quad (24)
\]

\[
\phi = \arcsin \left( \frac{y}{u_1} \right) \quad (25)
\]

\[
\dot{\theta} = \frac{x(g - z) + y\bar{z}}{(g - z)^2 + x^2} \quad (26)
\]

\[
\dot{\phi} = \frac{u_1\bar{y} - u_1\dot{y}}{u_1\sqrt{u_1^2 - y^2}} \quad (27)
\]

Singularity in this model only appear when \( g = \bar{z} \), in other words when the vehicle is in free fall. This can be avoided by constraining the input such that \( u_1 > 1 \) and the pitch and roll such that \( \theta < 90^\circ \) and \( \phi < 90^\circ \). These angles are outside the set \( S_c \) defined by (22).

4.3 Problem formulation

From the differentially flat equations the problem can be reposed to allow optimizations to occur within the output space as opposed to the control space. This is
beneficial because constraints such as obstacle avoidance occur in the output space, hence the computation time for constraint handling is reduced. The problem is posed as follows:

\[
\min_{y(t)} \Phi \quad \text{for } t \in [0, T]
\]

s.t. \( c_j(y) \leq 0, \quad x_0 - h_1(y(0)) = 0, \quad y_T - y(T) = 0 \) \hspace{1cm} (28)

where the inequality constraints are now expressed as a function of the output \( c_j(y) \) and the state is now a function of the output obtained from the differential flatness \( h_1(y) \). This problem can, with a suitable parameterization, be entered into MATLAB and solved using the optimization toolbox function \texttt{fmincon}.

4.4 Cost function

The objective function, \( \Phi \), is a quantitative measure of the optimality of the trajectory, which, can be approximated by a measure of the running costs. Assuming running costs are proportional to average velocity then the objective function can be defined as:

\[
\Phi = \frac{1}{T} \int_0^T \left( P_1 \dot{x}^2 + P_2 \dot{y}^2 + P_3 \dot{z}^2 \right) dt
\]

where \( P_1, P_2, P_3 \) are weighting factors.

4.5 Constraints

For efficient computation the number of constraints have to be kept small. From the differential flatness property, the state and input trajectories, \( x \) and \( u \) respectively, can be determined from the output trajectory. Inequality constraints on the output trajectory (for obstacle avoidance), and on the state and input trajectories (to avoid singularities and to provide constraints on the control signals) are expressed through the function \( c_j(y) \). Initial constraints are placed on the state at \( t = 0 \). Terminal constraints are also placed on the output trajectory to ensure the vehicle reaches its destination at \( t = T \), these are expressed by \( y(T) = y_f \).

5. PARAMETERIZATION

The optimization of the 4 outputs \((x,y,z,\psi)\) must be parameterized in order to reduce the dimensions of the problem to a finite amount. There are numerous alternatives for this task such as the polynomial (Nieuwstadt and Murray, 1995), Laguerre polynomials (Huzmezan et al., 2001) and Chebyshev polynomials (Vlassenbroeck and Van Dooren, 1988). Polynomials can also be conditioned using a basis scaling obtained from the Taylor series. Every technique, however, can be expressed as a product of a free variable \( \lambda \) and a basis function \( \Gamma_k \),

\[
f(t) = \sum_{k=0}^{M} a_k \Gamma_k(t) \quad \text{ (30)}
\]

where \( M \) is the order of the basis function. The search space hence becomes \( \mathbb{R}^{4(M+1)} \).

The polynomial basis function can be expressed as,

\[
\Gamma_k(t) = t^k. \quad \text{ (31)}
\]

Chebyshev polynomials can be defined as,

\[
\Gamma_k(x) = \cos(k \pi x) \text{ where } x = \cos(y). \quad \text{ (32)}
\]

Laguerre polynomials can be derived from recurrence relationship,

\[
\begin{align*}
\Gamma_0(t) &= 1 \\
\Gamma_1(t) &= 1 - t \\
(k + 1)\Gamma_{k+1}(t) &= (2K + 1 - t)\Gamma_k(t) - k\Gamma_{k-1}(t).
\end{align*}
\]

From the Taylor series the conditioning of the polynomials (31) can be improved using a basis scaling,

\[
\Gamma_k = (1/k!)t^k. \quad \text{ (34)}
\]

Polynomials, Laguerre polynomials, Chebyshev polynomials and Taylor series expansion polynomials have been tested for 3 missions defined in (6.1) and the results can be seen in Table (1).

<table>
<thead>
<tr>
<th>Iteration</th>
<th>The well</th>
<th>Vertical</th>
<th>Obstacle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial</td>
<td>30</td>
<td>85</td>
<td>17</td>
</tr>
<tr>
<td>Laguerre</td>
<td>21</td>
<td>33</td>
<td>18</td>
</tr>
<tr>
<td>Chebyshev</td>
<td>15</td>
<td>33</td>
<td>17</td>
</tr>
<tr>
<td>Taylor</td>
<td>30</td>
<td>30</td>
<td>20</td>
</tr>
</tbody>
</table>

Chebyshev polynomials are cylindrical in nature and therefore initial and final boundary conditions can be determined analytically and applied to reduce computation time although this proves complex for variable horizon times. Taylor series expansions and Laguerre polynomials both prove efficient parameterization techniques for this problem. Laguerre polynomials seem slightly better and are therefore chosen.

6. SIMULATION WITH FULL MODEL

To validate the control strategy a full simulation model of the quadrotor has been developed. The full SIMULINK model incorporates experimental data and theoretical analysis. This model is used to test the control algorithm to fly 3 missions. The model is described more fully in Cowling et al. (2006).

6.1 Missions

To test the control scheme, 3 missions have been created to test the robustness and performance of the
controller. All missions involve starting from hover at (0, 0, 0) and moving to a destination within the time horizon with a sampling rate of 10Hz. To test the robustness of the controller, disturbances have been added, these include translational drift and random gusting to simulate wind. To simulate inaccuracies within the rotor, random noise is also added to the voltage input into the model.

(1) The first mission involves a vertical flight of 5m in 7 seconds with a cross wind of 0.01 ms\(^{-1}\).

(2) The second mission involves navigation around an obstacle to reach a destination at (6,0,0) within 15 seconds. The obstacle is modelled as a sphere centered at (3,0,0) with a radius of 1m. A tail wind is modelled as a drift of 0.01 ms\(^{-1}\).

(3) The third mission is a horizontal flight to (10,0,0) followed by a descent down a mineshaft of radius 2m to the bottom(10,0,-5) within 25 seconds with a tail wind of 0.05 ms\(^{-1}\).

6.2 Results

The first mission consists of a 5m vertical flight within 7 seconds. To increase the chance of a feasible flight path when optimising the trajectory, a tolerance of 25cm is included in the terminal constraints. The flight path and reference trajectory for this mission are shown in Figure 3. The second mission involves the navigation around an obstacle to reach the destination within 15 seconds. As seen in Figure 4 the optimal path is very close to the obstacle, this is understandable as the optimal route is the shortest route. This does show the need for a factor of safety allowance into the path planning. The third mission involves a horizontal flight to the top of a mineshaft and then a vertical descent down to the bottom. The flight path and reference trajectory can be seen in Figure 5. The wind acts on the vehicle when the vehicle is above ground but not when the vehicle is in the mineshaft. As seen, the vehicle drifts from the trajectory above ground but still passes down the mineshaft without hitting the walls.

This paper presents an optimal trajectory planner with a linear control scheme to follow a reference trajectory. This scheme has been validated using a full dynamic model of the quadrotor. Such a scheme can be used to achieve real time autonomous behaviour of unmanned air vehicles.

The disadvantages of linear trajectory following occur when the trajectory becomes sub-optimal or infeasible due to environmental changes. This will be considered in future work by implementing a dual loop control scheme. An outer loop will be a trajectory optimization and the inner loop will be a stabilizing trajectory follower. Future work will also extend the scheme to a larger quadrotor (Draganflyer X-Pro) and include dynamic modelling and flight testing.

REFERENCES


Fig. 4. Obstacle Avoidance

Fig. 5. Mineshaft descent