



Faculty Publications

2011-9

Differential Flatness Based Control of a Rotorcraft For Aggressive Maneuvers

Timothy McLain

Mechanical Engineering Department, Brigham Young University, mclain@byu.edu

Randal W. Beard

Department of Electrical and Computer Engineering, Brigham Young University, beard@ee.byu.edu

Robert C. Leishman

US AFRL Sensors Directorate, Wright-Patterson AFB

Jeffrey L. Ferrin

Brigham Young University

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Mechanical Engineering Commons](#)

Original Publication Citation

Ferrin, J., Leishman, R., Beard, R., and McLain, T. Differential Flatness Based Control of a Rotorcraft for Aggressive Maneuvers, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2688-2693, September 2011, San Francisco, California.

BYU ScholarsArchive Citation

McLain, Timothy; Beard, Randal W.; Leishman, Robert C.; and Ferrin, Jeffrey L., "Differential Flatness Based Control of a Rotorcraft For Aggressive Maneuvers" (2011). *Faculty Publications*. 1949.
<https://scholarsarchive.byu.edu/facpub/1949>

This Conference Paper is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

Differential Flatness Based Control of a Rotorcraft For Aggressive Maneuvers

Jeff Ferrin, Robert Leishman, Randy Beard, and Tim McLain

Abstract—We propose a new method to control a multi-rotor aerial vehicle. We show that the system dynamics are differentially flat. We utilize the differential flatness of the system to provide a feed forward input. The system model derived allows for arbitrary changes in yaw and is not limited to small roll and pitch angles. We demonstrate in hardware the ability to follow a highly maneuverable path while tracking a time-varying heading command.

I. INTRODUCTION

Simple rotorcraft are becoming increasingly utilized for aerial robotics research [15], [8], [16], [1], [7], [3]. These aircraft are well adapted for research as they are relatively inexpensive, easily repairable, carry a substantial payload for their size, and provide hover capability.

Path planning and control for rotorcraft has been studied extensively as well. For example, Hoffmann, et al. [4] implemented a path planning and control method that specifically avoided utilizing feed-forward control. They report indoor and outdoor hardware flight results. Indoor flight errors were at or below 0.1 m, in any one dimension, at a max speed of 0.5 m/s.

Recently, researchers have been able to complete complex maneuvers using these aircraft and motion capture systems. Researchers with the GRASP Lab have developed control schemes for flips, landing on walls, and flying through narrow windows [11]. Researchers with the ETH in Zurich have completed algorithms for moving to music using quadrotors [17]. Both of these approaches describe the need of feed forward terms to perform such maneuvers, but do not elaborate further.

Cowling, et al. [5] created a path planner and path follower based on the principle of differential flatness. Essentially, a feed-forward input is obtained from the desired path and the dynamics of the system. The authors implemented an optimization scheme for online path planning and show simulation results. However, the approach is limited to a zero-degree yaw angle and to small roll and pitch angles.

We propose a new method of control utilizing the differential flatness of a rotorcraft platform that allows for arbitrary changes in yaw and is not limited to small roll and pitch angles. This scheme is shown to be adequate for aggressive maneuvers.

The paper is organized as follows. In Section II we outline the dynamic model of the hexacopter system. We derive the differentially flat outputs of the rotorcraft and present the system architecture in Section III. Section IV discusses the

LQR controller that closes the loop. Implementation notes and the results that were obtained are presented in Section V, and finally, conclusions and future work are in Section VI.

II. DYNAMIC MODEL

We first set up the notation and coordinate systems that will be used in the derivation of the equations of motion for the hexacopter. A vector v in frame a will be denoted as v^a and a rotation matrix that rotates a vector from frame a to frame b is R_a^b . We use the rotation matrices as derived in [2].

There are 12 state variables that will be used in deriving the equations of motion. These states are

$$\mathbf{x} = [p_n \ p_e \ p_d \ \dot{p}_n \ \dot{p}_e \ \dot{p}_d \ \phi \ \theta \ \psi \ p \ q \ r]^T,$$

where p_n , p_e and p_d are the north, east and down positions, respectively, in the inertial coordinate frame. The inertial velocities are \dot{p}_n , \dot{p}_e , \dot{p}_d . The Euler angles ϕ , θ and ψ describe the orientation and p , q and r are the body-frame angular rates.

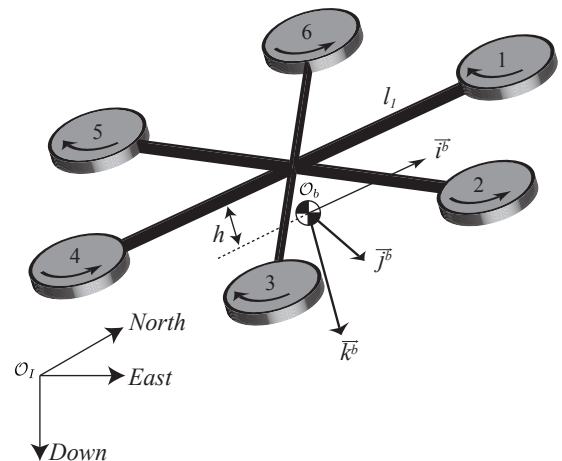


Fig. 1. Hexacopter layout and coordinate systems.

The dynamic equations of motion for the translation of the hexacopter in the inertial frame are

$$\begin{bmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{bmatrix} = R_b^i(\psi, \theta, \phi) \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \left(\frac{1}{m_h} \right) + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \quad (1)$$

where m_h is the mass of the hexacopter, T is the total thrust acting on the hexacopter in the upward direction, and g is gravity.

Applying the rotation in Equation (1) gives

$$\begin{bmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{bmatrix} = \frac{-T}{m_h} \begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}. \quad (2)$$

The relationship between the angular orientations of the hexacopter and the angular rates as shown in [2] is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad (3)$$

where $c_\theta \triangleq \cos \theta$, $s_\theta \triangleq \sin \theta$, and $t_\theta \triangleq \tan \theta$, etc.

The dynamic equations for rotational motion as also shown in [2] are

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 p q - \Gamma_2 q r \\ \Gamma_5 p r - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 p q - \Gamma_1 q r \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{J_y} m \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix}. \quad (4)$$

The Γ_i terms are functions of the mass moments of inertia and l , m , and n are the moments about the body frame axes.

The complete set of equations of motion for the hexacopter are comprised of the simple kinematic relationship between position and velocity and Equations (2), (3) and (4). It is important to note that, up to this point, no assumptions have been made regarding the number of rotors on the vehicle in the derivation. In our case, we define the thrust T with six inputs. This can simply be changed to the number of rotors on the rotorcraft.

A. State-Space Model

The hexacopter we are using comes equipped with an on-board attitude controller. The attitude controller accepts as inputs the desired thrust T^d , roll angle ϕ^d , pitch angle θ^d and body yaw rate r^d . We define the vector of inputs to the hexacopter as

$$\boldsymbol{\nu} = \begin{bmatrix} T^d \\ \phi^d \\ \theta^d \\ r^d \end{bmatrix}. \quad (5)$$

The on-board attitude controller closes the loop on the desired input commands which eliminates the need to control the moments l , m , and n ; accordingly we simplify the state vector as

$$\boldsymbol{x} = [p_n \ p_e \ p_d \ \dot{p}_n \ \dot{p}_e \ \dot{p}_d \ \psi]^T. \quad (6)$$

We now demonstrate how we can describe this system using linear equations through a change of variables. We define a new input vector as a function of the hexacopter inputs T^d , ϕ^d , θ^d and r^d and the state ψ , which eliminates the need to linearize. For this derivation we assume that the hexacopter attitude controller will achieve the commanded inputs with zero error, i.e., $T = T^d$, $\phi = \phi^d$, $\theta = \theta^d$ and

$r = r^d$. Our new input vector, a non-linear map of the inputs $\boldsymbol{\nu}$ and state \boldsymbol{x} , is defined as follows:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_{p(3x1)} \\ \boldsymbol{u}_{\psi(1x1)} \end{bmatrix} \triangleq \begin{bmatrix} \hat{f}_p(\boldsymbol{x}, \boldsymbol{\nu}) \\ \hat{f}_\psi(\boldsymbol{\nu}, q) \end{bmatrix}, \quad (7)$$

where

$$\hat{f}_p(\boldsymbol{x}, \boldsymbol{\nu}) = R_b^i(\psi, \theta, \phi) \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} \left(\frac{1}{m_h} \right). \quad (8)$$

and

$$\hat{f}_\psi(\boldsymbol{\nu}, q) = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta}. \quad (9)$$

The carets above the functions denote that these are models that are estimating the physical system. The vector \boldsymbol{u}_p includes the three inputs that affect \ddot{p}_n , \ddot{p}_e and \ddot{p}_d from the thrust rotated into the inertial frame using Equation (1). The mapping \boldsymbol{u}_ψ affects the heading of the vehicle.

Using the change of variables defined by Equations (8) and (9) we can now write the dynamics in state-space form as

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} + b\boldsymbol{g}, \quad (10)$$

where

$$A = \begin{bmatrix} 0_{3x3} & I_{3x3} & 0_{3x1} \\ 0_{3x3} & 0_{3x3} & 0_{3x1} \\ 0_{1x3} & 0_{1x3} & 0_{1x1} \end{bmatrix},$$

$$B = \begin{bmatrix} 0_{3x3} & 0_{3x1} \\ I_{3x3} & 0_{3x1} \\ 0_{1x3} & 1 \end{bmatrix},$$

and

$$b = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T.$$

B. Inverse Function for Control Inputs

Equation (10) shows how the system states evolve as functions of the input \boldsymbol{u} . The input \boldsymbol{u} is in units of acceleration while the hexacopter inputs $\boldsymbol{\nu}$, Equation (5), are in units of force, angles and angular velocity respectively. Therefore, the hexacopter inputs $\boldsymbol{\nu}$ must be converted from the input \boldsymbol{u} . By using the inverse functions \hat{f}_p^{-1} and \hat{f}_ψ^{-1} we are able to compute the inputs $\boldsymbol{\nu}$ that will give us the desired accelerations based on the desired trajectory.

Using Equation (8), we can solve for the first three input commands T , ϕ and θ . We write Equation (8) as

$$\begin{bmatrix} u_{p1} \\ u_{p2} \\ u_{p3} \end{bmatrix} m_h = R^T(\psi) R^T(\theta) R^T(\phi) \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} T.$$

Taking the norm of both sides gives

$$T = m_h \sqrt{u_{p1}^2 + u_{p2}^2 + u_{p3}^2}. \quad (11)$$

Now solving for the input commands ϕ and θ , we further manipulate Equation (8) as

$$R(\psi) \begin{bmatrix} u_{p1} \\ u_{p2} \\ u_{p3} \end{bmatrix} \frac{m_h}{-T} = R^T(\theta) R^T(\phi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (12)$$

For clarity we define

$$\mathbf{z} = R(\psi) \begin{bmatrix} u_{p1} \\ u_{p2} \\ u_{p3} \end{bmatrix} \begin{bmatrix} m_h \\ -T \end{bmatrix}. \quad (13)$$

Using Equations (12) and (13) we get that

$$\phi = \sin^{-1}(-z_2) \quad (14)$$

and

$$\theta = \tan^{-1} \left(\frac{z_1}{z_3} \right), \quad (15)$$

where the value z_i is the i^{th} element of \mathbf{z} . Equations (11), (14) and (15) make up the inverse function \hat{f}_p^{-1} . The last input r , is equal to \hat{f}_ψ^{-1} and is solved from Equation (9) as

$$r = \hat{f}_\psi^{-1} = \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi, \quad (16)$$

where $\dot{\theta}$ is calculated from the time derivative of Equation (15).

III. DIFFERENTIAL FLATNESS

A differentially flat system is one in which the state and control inputs can be expressed as functions of the output and its time derivatives [6], [14], [10]. In other words,

$$\mathbf{y} = h(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}, \dots, \mathbf{u}^{(k)}) \quad (17)$$

is a flat output if there exists smooth functions g_x and g_u such that

$$\mathbf{x} = g_x(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots, \mathbf{y}^{(k)}) \quad (18)$$

and

$$\mathbf{u} = g_u(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots, \mathbf{y}^{(k)}). \quad (19)$$

This implies that by specifying the output and the output derivatives, both the input and the state equations can be uniquely expressed as functions of the specified output equations.

For the system we define the flat output vector corresponding to Equation (17) as a function of only the reference states \mathbf{x}^r as

$$\mathbf{y}^{traj} \triangleq h(\mathbf{x}^r) = \begin{bmatrix} p_n^r \\ p_e^r \\ p_d^r \\ \psi^r \end{bmatrix}.$$

We define the state and control inputs calculated from the differential flatness as the reference states

$$\mathbf{x}^r \triangleq g_x(\mathbf{y}^{traj}, \dot{\mathbf{y}}^{traj}) \quad (20)$$

and the reference inputs

$$\mathbf{u}^r \triangleq g_u(\mathbf{y}^{traj}, \dot{\mathbf{y}}^{traj}, \ddot{\mathbf{y}}^{traj}). \quad (21)$$

The reference states \mathbf{x}^r are defined in Equation (6). The reference inputs \mathbf{u}^r are defined as follows:

$$\mathbf{u}^r = \begin{bmatrix} \mathbf{u}_p^r \\ \mathbf{u}_\psi^r \end{bmatrix} \quad (22)$$

where

$$\mathbf{u}_p^r = \begin{bmatrix} \ddot{p}_n^r \\ \ddot{p}_e^r \\ \ddot{p}_d^r \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (23)$$

and

$$\mathbf{u}_\psi^r = \dot{\psi}. \quad (24)$$

The input \mathbf{u}^r gives us the required inputs in terms of the system in Equation (10), these are then converted to the hexacopter units using the inverse functions \hat{f}_p^{-1} and \hat{f}_ψ^{-1} .

From Equations (11), (15), (14) and (16) we now can express desired states and inputs as functions of any twice-differentiable desired trajectory. The limits for this approach are when the quadrotor is in free-fall and when $\phi = \pm \frac{\pi}{2}$. Both of these scenarios can be avoided by judicious path selection.

A. System Architecture

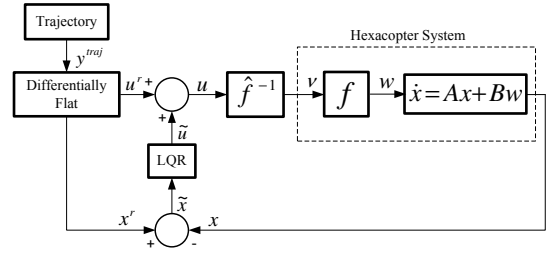


Fig. 2. System block diagram.

Figure 2 shows the block diagram of the system. The trajectory block generates a vector \mathbf{y}^{traj} that consists of a desired position. The Differentially Flat block takes these trajectory parameters and calculates a reference input \mathbf{u}^r from Equations (22) and (24) and also calculates the reference states \mathbf{x}^r from Equation (6). A Linear Quadratic Controller (LQR) is used for the feedback control. The input vector \mathbf{u} is fed into the inverse function \hat{f}^{-1} which calculates the actual commands $\boldsymbol{\nu}$ which are sent to the hexacopter on-board attitude controller. The hexacopter system is broken up into two blocks. The function f is the actual behavior of the physical system to the input vector $\boldsymbol{\nu}$ which produces actual accelerations \mathbf{w} . The second block is the evolution of the states from the input \mathbf{w} .

IV. CLOSED-LOOP CONTROL

The state-space model for the system is given in Equation (10). The state equations are dependent on the input vector \mathbf{u} . The hexacopter system as shown in Figure 2 contains the function f which produces the actual accelerations \mathbf{w} . We assume here that our function \hat{f} that models the physical behavior of the hexacopter is a perfect model, i.e., $\hat{f} = f$ which then implies that $\mathbf{u} = \mathbf{w}$.

With $\hat{f} = f$ the model in Equation (10) is used to design an LQR controller. We define $\tilde{\mathbf{x}}$ as the deviation of the state vector from the reference state \mathbf{x}^r as

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^r,$$

resulting in the error equations

$$\dot{\tilde{\mathbf{x}}} = A\tilde{\mathbf{x}} + B\tilde{\mathbf{u}}. \quad (25)$$

We desire to control the system using full state feedback where $\tilde{\mathbf{u}} = -K\tilde{\mathbf{x}}$. Our goal is to design a LQR controller for the hexacopter. The LQR controller solves for the optimal state feedback matrix K that minimizes [9]

$$J = \int_0^{\infty} \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u} dt, \quad (26)$$

where the Q and R are symmetric positive-definite weighting matrices. The gains used on the hardware implementation were calculated using Bryson's rule [9]. This creates a controller that will produce the maximum input into the system when the error is at the maximum desired value.

V. IMPLEMENTATION AND RESULTS

The control algorithm was tested on the Mikrokoopter Hexacopter [12]. The state information, Equation (6), is obtained using a motion capture system from Motion Analysis Corporation [13], which tracks the position of the reflective spheres on the hexacopter to give inertial position and orientation. The flyable volume seen by the motion capture system is 4 m by 3 m by 1.6 m, which is fairly small for the speeds achieved. The control commands are calculated on a ground station computer that receives state information from the motion capture system and sends the commands via a wireless connection using an XBee modem.

A. Path Example

Any twice differentiable function can be used as a trajectory for the hexacopter. Four time-dependent equations are required, one each for $p_n(t)$, $p_e(t)$, $p_d(t)$ and $\psi(t)$. These equations are differentiated to find $\dot{p}_n(t)$, $\dot{p}_e(t)$, $\dot{p}_d(t)$ and $\dot{\psi}(t)$, then differentiated again to find $\ddot{p}_n(t)$, $\ddot{p}_e(t)$, $\ddot{p}_d(t)$ and $\ddot{\psi}(t)$. These trajectory equations make up the \mathbf{y}^{traj} signal in Figure 2.

We now show an example figure eight path. The position trajectory of the hexacopter is given as

$$\begin{bmatrix} p_n(t) \\ p_e(t) \\ p_d(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} \alpha \cos(\frac{\omega}{2}t) \\ \beta \sin(\omega t) \\ \eta \\ 0 \end{bmatrix}, \quad (27)$$

where α and β are amplitude gains in the north and east directions respectively and η is the height of the flight. The values for this example path are $\alpha = 1.5$, $\beta = 0.75$ and $\eta = -0.75$.

The first and second derivatives of Equation (27) are used to compute u^r , which are then added to the feedback control commands, $\tilde{\mathbf{u}}$, as shown in Figure 2. The value of Δt is the time for one complete lap, and is chosen to be 5 s for this example. In Figure 3 we show the desired roll and pitch angles and the predicted speed of the aircraft for this example trajectory. For this path the max roll angle is 28° , the max pitch angle is 63° and the max velocity is 4.25 m/s.

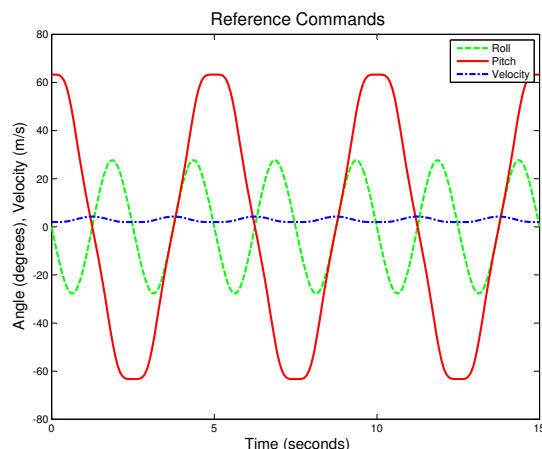


Fig. 3. Roll, pitch and velocity commands for the figure eight path.

B. Successive Loop Closure

During the experimentation of this control scheme, some difficulty was experienced with the interface into the inner loop on the Mikrokoopter autopilot, resulting in poor initial performance. Ultimately, we were able to use successive loop closure [2] with ϕ and θ to solve the problem. The values for ϕ and θ in ν become desired angles and are used with the actual angles from the motion capture system in a proportional and derivative (PD) control feedback loop. With this change, we were able to generate the results below.

C. Results

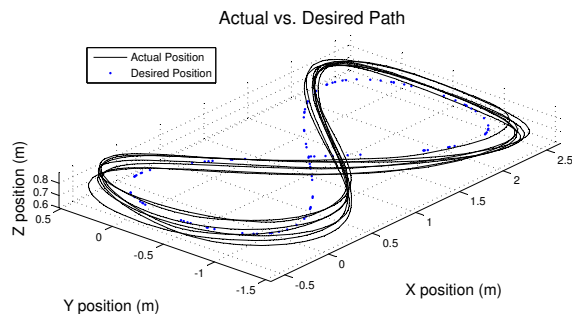


Fig. 4. Figure eight path with a 5-second lap time.

Figure 4 shows the results of a figure eight flight of the hexacopter platform. The path trajectory is given in Equation (27) with $\alpha = 1.5$, $\beta = 0.75$ and $\eta = -0.75$. The full loop was completed in 5 s with a maximum speed of 4 m/s. The path errors for this path are shown in Figure 5. Figure 6 shows the velocities achieved by the hexacopter while flying this path. The actual hexacopter roll and pitch angles are shown in Figure 7. Note the differences between these actual angles and the predicted angles shown in Figure 3. This discrepancy is due to the fact that the attitude controller does not arrive at the commanded angles.

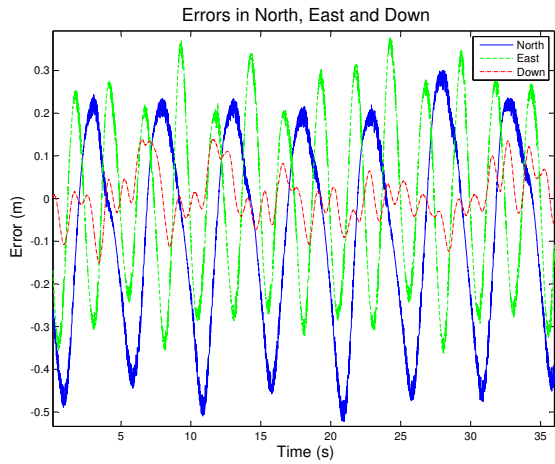


Fig. 5. Off-path error for a figure eight path with a 5-second lap time.

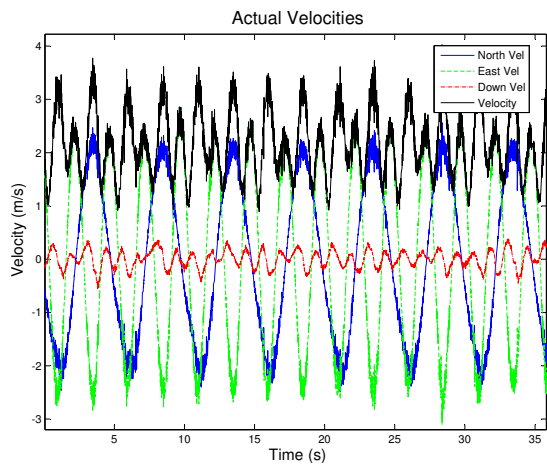


Fig. 6. Velocities while flying the figure eight with a 5-second lap time.

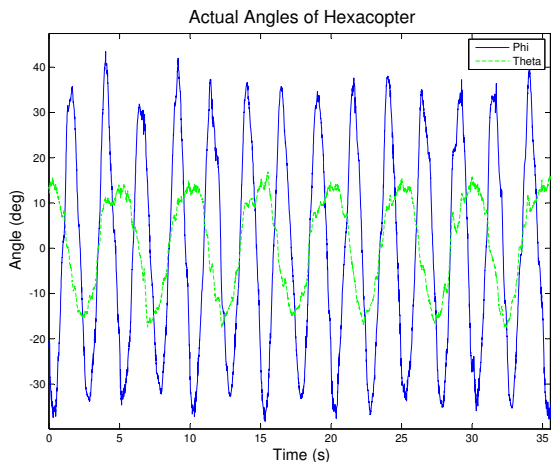


Fig. 7. Actual hexacopter angles while flying a figure eight path with a 5-second lap time.

This figure eight path was also used to fly the hexacopter without the feed forward term so that a comparison could be made between the differentially flat method proposed in this paper and a standard LQR controller without the feed

forward term. The hexacopter was unable to successfully fly the figure eight at a lap time of 5 s without the feed forward terms. Figure 8 shows the path of the hexacopter flying a figure eight path with a lap time of 6 s, without feed forward terms. The performance of the vehicle degraded without the use of the feed forward terms in a path with such aggressive angle commands. Figure 9 shows the path error for the 6-second figure eight path. The mean error for the 5-second figure eight path flown with the feed forward control is 0.305 m while the mean error for the 6-second figure eight path flown without the feed forward term is 0.573 m.

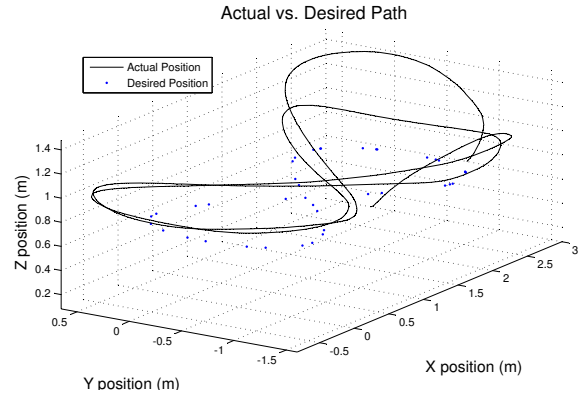


Fig. 8. Figure eight path with a 6-second lap time and no feed forward control.

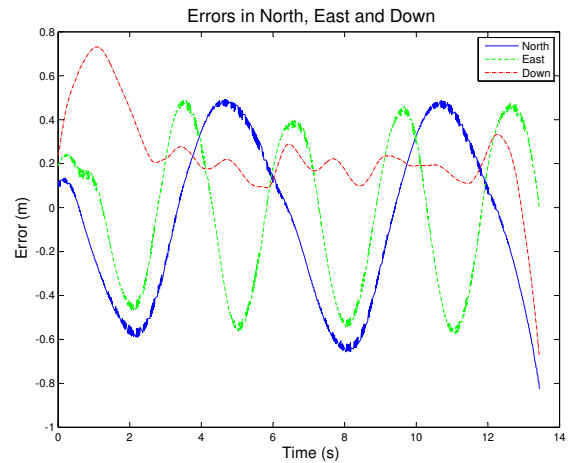


Fig. 9. Off-path error for a figure eight path with a 6-second lap time and no feed forward control.

Figure 10 shows an inclined circular path. The radius of the path is 0.75 m, with a 0.5 m change in height, and a constant desired yaw velocity of 0.23 rad/s. The hexacopter follows this path at 1.4 m/s. The errors in the North, East, and Down directions are shown in Figure 11. The desired yaw compared to the actual is shown in Figure 12. These results demonstrate that the control scheme can handle large changes in heading while performing aggressive maneuvers.

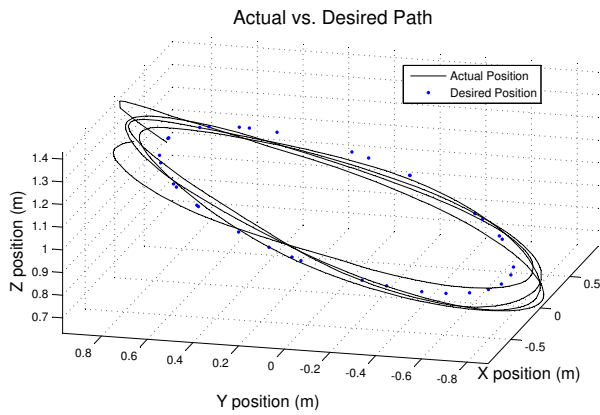


Fig. 10. A circular path with changing height and heading at 1.4 m/s.

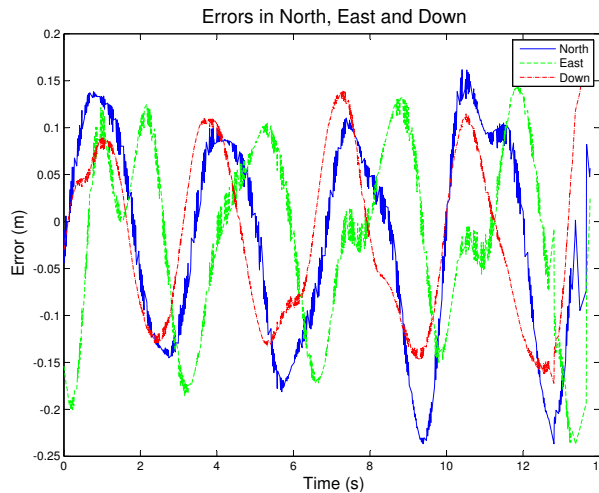


Fig. 11. The error in the North, East, and Down directions for the circular path at 1.4 m/s.

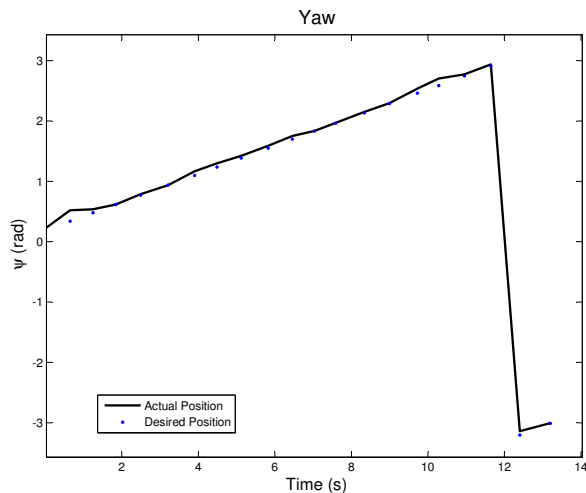


Fig. 12. Comparison of yaw to yaw-desired for the circular path.

VI. CONCLUSION AND FUTURE WORK

We have presented a method of control for rotorcraft that utilizes the differentially flat dynamics to generate feed forward commands. This method improves the path following

control scheme. The principle and accompanying equations are simple and easy to implement. The trajectories are also easy to generate. It allows for aggressive maneuvers along any smooth path with heading change. We have demonstrated the performance of this control scheme with hardware using a Mikrokopter hexacopter. Future work will include development of a robustness analysis given that the function \hat{f} is not a perfect model. We will investigate how this affects the performance of the system.

ACKNOWLEDGMENT

This research was supported by ONR contract number N00014-10-M-0345 through a subcontract to Scientific Systems Company, Inc. and through the DoD SMART Scholarship program.

REFERENCES

- [1] S. Ahrens, D. Levine, G. Andrews, and J. P. How, "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," in *ICRA '09*, 2009, pp. 2643–2648.
- [2] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft*. Princeton University Press, 2011.
- [3] M. Bloesch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf.*, 2010, pp. 21–28.
- [4] H. G. W. S. T. C.J., "Quadrotor helicopter trajectory tracking control," in *Proc. AIAA GNC, Honolulu, HI*, 2008.
- [5] I. Cowling, O. Yakimenko, J. Whidborne, and A. Cooke, "A prototype of an autonomous controller for a quadrotor UAV," in *Proceedings of the European Control Conference*, 2007.
- [6] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," CAS, Tech. Rep. A-284, January 1994.
- [7] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *ICRA '09*, 2009, pp. 2878–2883.
- [8] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 khz," in *Proc. IEEE Int Robotics and Automation Conf.*, 2007, pp. 361–366.
- [9] J. P. Hespanha, *Linear Systems Theory*. Princeton University Press, 2009.
- [10] P. Martin, R. Murray, and P. Rouchon, "Flat systems," 1997.
- [11] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
- [12] Mikrokopter, "http://www.mikrokopter.de/"
- [13] Motion Analysis, "http://www.motionanalysis.com/"
- [14] R. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *Int'l Mech Eng Congress and Expo.* ASME, November 1995.
- [15] P. C. Paul Pounds, Robert Mahony, "Modelling and control of a quadrotor robot," in *Australasian Conference on Robotics and Automation*, 2006.
- [16] H. Peng, J. Wu, and Q. Chen, "Modeling and control approach to a quadrotor helicopter," in *Seventh China-Japan Intl. Workshop Info Tech and Control.* Hunan, China: Central South University China, December 2009, pp. 46–55.
- [17] A. Schllig, F. Augugliaro, and R. D'Andrea, "A platform for dance performances with multiple quadcopters," in *IROS Wkshop Robots and Musical Expressions*, 2010, pp. 1 – 8.