2011-12

Minimum Time Control of Helicopter UAVs using Computational Dynamic Optimization

Tang, K.; Wang, B.; Kang, W.; Chen, B.

2011 9th IEEE International Conference on Control and Automation (ICCA), Santiago, Chile, December 19-21, 2011
http://hdl.handle.net/10945/42202

2011 9th IEEE International Conference on
Control and Automation (ICCA)
Santiago, Chile, December 19-21, 2011

Minimum Time Control of Helicopter UAVs using Computational Dynamic Optimization

K.Tang, B. Wang, W. Kang, and B. Chen

***Abstract*— This paper focuses on the problem of minimum time trajectory planning for helicopter UAVs. It is formulated as a nonlinear optimal control subject to the dynamics and limitations of helicopter UAVs. The dynamical system is defined by a set of fifteen states nonlinear differential equations developed for HeLion, a UAV helicopter constructed in National University of Singapore (NUS). The problem is then solved numerically using pseudospectral method for dynamic optimization. The results show that minimum time trajectories are highly nonlinear that require complicated maneuvering.**

I. INTRODUCTION

In this paper, we explore computational approaches and methods for optimal UAV (Unmanned Aerial Vehicle) trajectory planning. It is a significant challenge for autonomous UAVs to fly in an optimal manner with respect to expenditure of time, use of fuel, or target tracking errors. On the other hand, the rapid development of computational optimal control has resulted in enabling new methodologies with successful real-life applications. In this paper, we explore the application of a pseudospectral method for dynamic optimization to the problem of minimum time trajectory planning for helicopter UAVs.

It is important here to distinguish between path planning and trajectory planning. Path planning finds a feasible path from start to goal. When the complete trajectory including all states and control inputs as functions of time for the computed path is determined, it becomes trajectory planning [1], [2]. If the trajectory planning requires the optimization of certain cost, then the problem is called *optimal trajectory planning*. A vast volume of work can be found in the literature of path planning. Using a planned or desired path, the problem of trajectory planning can be solved by inverse the dynamics or an inner-loop feedback that tracks the desired path. However, these methods cannot solve the problem of optimal trajectory planning. Existing approaches of path planning emphasizes stability and robustness. The goal of research in this paper is different from path planning in the sense that we emphasize the planning of optimal trajectories, more specifically minimum time trajectories, taking into the consideration of the full nonlinear dynamics of the system. Due to the complexity of helicopter aerodynamics, it is impossible to find analytic solutions for optimal trajectories. Therefore, finding numerical solutions is the more appropriate method.

In the next section, the problem definition and the differential equations for the dynamics of helicopter UAVs are introduced. The helicopter model is developed by the National University of Singapore and is outlined further in [3]. Initial work was done in [6] where the author used a preliminary helicopter model. We used an updated model because results using the preliminary model were chattery and not realistic. A pseudospectral computational method is outlined in Section III. Some numerical results and several minimum time trajectories are shown in IV.

II. PROBLEM DEFINITION

To compute a minimum time 3-D UAV trajectory, we formulate the following problem of optimal control

$$\begin{array}{l} \min J = \int_{t_0}^{t_f} 1dt \\ \text{subject to} \\ \dot{x} = f(x,u) \\ x_{\min} \le x \le x_{\max}, \quad u_{\min} \le u \le u_{\max} \\ h(x) \ge 0 \ \text{(obstacles)} \\ x(t_0) = x_0, x(t_f) = x_f, t_f \ \text{is unspecified} \end{array} \qquad (1)$$

In this formulation, the state and control constraints defined by $x_{\max}$, $x_{\min}$ $u_{\max}$, and $u_{\min}$ represent the limitations of the variables. The obstacle constraint is for future research which is not included in this paper. The differential equation of $f(x,u)$ represents the helicopter model. HeLion is a helicopter UAV built in NUS. Its mathematical model is a system of fifteen nonlinear differential equations [3]. The model is based on two coordinate frames, i.e., the body frame and the north-east-down (NED) frame. The state consists of the following variables

$$\begin{array}{rcl} x & = & [p_x \quad p_y \quad p_z \quad \phi \quad \theta \quad \psi \quad \cdots \\ & & \quad u \quad v \quad w \quad p \quad q \quad r \quad a_s \quad b_s \quad \delta_{ped,int}]^T \end{array}$$

The control input consists of the following variables

$$u = \begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{col} & \delta_{ped} \end{bmatrix}^T$$

These variables are explained in Table I - II.

The helicopter model consists of four key components which lead to a total of 15 differential equations [3]: 1)

This work was supported in part by TDSI and AFOSR

K. Tang is with the internship program, Naval Postgraduate School, Monterey, CA 93943, USA ktang200@yahoo.com

B. Wang is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 elewb@nus.edu.sg

W. Kang is with the Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943, USA wkang@nps.edu

B.M. Chen is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 bmchen@nus.edu.sg

Variable	Physical meaning (unit)
p_x, p_y, p_z	Position vector in NED-frame (m)
u, v, w	Velocity vector in body-frame (m/s)
ϕ, θ, ψ	Roll, pitch, and yaw angles (rad/s)
p, q, r	Roll, pitch, and yaw angular rate in body-frame (rad/s)
a_s, b_s	Longitudinal and lateral tip-path-plane (TPP) flapping angle
$\delta_{ped,int}$	Intermediate state in yaw rate gyro dynamics

TABLE I

PHYSICAL MEANINGS OF THE STATE VARIABLES

δ_{lat}	Normalized aileron servo input (-1, 1)
δ_{lon}	Normalized elevator servo input (-1,1)
δ_{col}	Normalized collective pitch servo input (-1, 1)
δ_{ped}	Normalized rudder servo input (-1, 1)

TABLE II

PHYSICAL MEANINGS OF THE CONTROL VARIABLES

kinematics, 2) six degree-of-freedom (DOF) rigid-body dynamics, 3) main rotor flapping dynamics, and 4) yaw rate gyro dynamics. They are summarized as follows.

$$\begin{aligned}
\dot{P}_n &= B_B V_b \\
\dot{\Omega}_n &= S_B \cdot \Omega_b \\
\dot{V}_b &= -\Omega_b \times V_b + \frac{F_b}{m} + \frac{F_g}{m} \\
\dot{\Omega}_b &= I^{-1}(M_b - \Omega_b \times I \cdot \Omega_b) \\
\dot{a}_s &= -q - \frac{a_s}{\tau} + \frac{A_{long}}{\tau}\delta_{lon} \\
&\quad + \frac{1}{\tau}\left[\frac{2K_u}{\Omega R}\left(\frac{8C_T}{\sigma C_{l\alpha,mr}} + \sqrt{\frac{C_T}{2}}\right)u_a \right. \\
&\quad \left. + \frac{K_u}{\Omega R}\frac{16\mu^2}{8|\mu| + \sigma C_{l\alpha,mr}}\mathrm{sgn}(\mu) w_a\right] \\
\dot{b}_s &= -p - \frac{b_s}{\tau} \\
&\quad + \frac{2K_\mu}{\tau \Omega R}\left(\frac{8C_T}{\sigma C_{l\alpha,mr}} + \sqrt{\frac{C_T}{2}}\right)v_a + \frac{B_{lat}}{\tau}\delta_{lat} \\
\dot{\delta}_{ped,int} &= K_a \delta_{\text{ped}} - r
\end{aligned} \tag{2}$$

In the equations, $P_n = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$ is the position vector in the NED frame; $V_b = \begin{bmatrix} u & v & w \end{bmatrix}^T$ is the velocity vector in the body frame; B_B is the transformation matrix defined by

$$B_B = \begin{bmatrix} \mathbf{c}_\theta \mathbf{c}_\psi & \mathbf{s}_\phi \mathbf{s}_\theta \mathbf{c}_\psi - \mathbf{c}_\phi \mathbf{s}_\psi & \mathbf{c}_\phi \mathbf{s}_\theta \mathbf{c}_\psi + \mathbf{s}_\phi \mathbf{s}_\psi \\ \mathbf{c}_\theta \mathbf{s}_\psi & \mathbf{s}_\phi \mathbf{s}_\theta \mathbf{s}_\psi + \mathbf{c}_\phi \mathbf{c}_\psi & \mathbf{c}_\phi \mathbf{s}_\theta \mathbf{s}_\psi - \mathbf{s}_\phi \mathbf{c}_\psi \\ -\mathbf{s}_\theta & \mathbf{s}_\phi \mathbf{c}_\theta & \mathbf{c}_\phi \mathbf{c}_\theta \end{bmatrix} \tag{3}$$

with $\mathbf{c}_* = \cos(*)$ and $\mathbf{s}_* = \sin(*)$; $\Omega_n = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is the Euler angle vector; $\Omega_b = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is the angular rate vector in the body frame; and S_B is the corresponding transformation matrix defined by

$$S_B = \begin{bmatrix} 1 & \mathbf{t}_\theta \mathbf{s}_\phi & \mathbf{t}_\theta \mathbf{c}_\phi \\ 0 & \mathbf{c}_\phi & -\mathbf{s}_\phi \\ 0 & \mathbf{s}_\phi/\mathbf{c}_\theta & \mathbf{c}_\phi/\mathbf{c}_\theta \end{bmatrix}$$

where $\mathbf{t}_* = \tan *$. In (2), m is the total mass of the UAV helicopter, i.e., HeLion; F_g is the gravitational force vector; $I = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$ is the moment of inertia matrix measured via pendulum experiment; F_b and M_b are the aerodynamic force and moment vectors, which consist of several components including the main rotor force and moment, the tail rotor force and moment, fuselage forces, vertical fin forces and moments, and horizontal fin forces and moments. The lengthy definitions and mathematical models of the forces and moments are omitted. Readers are referred to [3] for details. The equations of $\dot{a}_s$ and $\dot{b}_s$ represent the main rotor flapping dynamics. Since the yaw channel of hobby-based helicopters is highly sensitive, the yaw rate gyro dynamics is included in the model using an intermediate variable $\delta_{ped,int}$.

III. COMPUTATIONAL METHOD

Pseudospectral (PS) methods of computational dynamic optimization have been actively developed for the last twenty years with many real-life applications. The methods lead to numerical algorithms of solving nonlinear optimal control problems. In a PS approximation based on Legendre-Gauss-Lobatto (LGL) quadrature nodes, a function $f(t)$ is approximated by N-th order Lagrange polynomials using the interpolation at these nodes. The LGL nodes, $t_0 = -1 < t_1 < \cdots < t_N = 1$, are defined by

$$\begin{aligned} &t_0 = -1, \;\; t_N = 1, \text{ and} \\ &\text{for } k = 1, 2, \ldots, N-1, t_k \text{ are the roots of } \dot{L}_N(t) \end{aligned}$$

where $\dot{L}_N(t)$ is the derivative of the N-th order Legendre polynomial $L_N(t)$. The discretization works in the interval of $[-1, 1]$. It is proved in approximation theory that the polynomial interpolation at the LGL nodes converges to $f(t)$ under L^2 norm at the rate of $1/N^m$, where m is the smoothness of $f(t)$ [4]. If $f(t)$ is C^∞, then the polynomial interpolation at the LGL nodes converges at a spectral rate, i.e. it is faster than any given polynomial rate. In a PS method, the state trajectory, $x(t)$, is approximated by the vector

$$\bar{x}^{Nk} \approx x(t_k) \in \Re^n, \;\; k = 1, 2, \cdots, N$$

Similarly, $\bar{u}^{Nk}$ is the approximation of $u(t_k)$. PS method is a good all-around method for the approximation of smooth functions, integrations, and differentiations, all critical to optimal control problems. For differentiation, the derivative of a function $h(t)$ at the LGL nodes is easily approximated by the following matrix multiplication

$$\begin{aligned} &\begin{bmatrix} \dot{h}(t_0) & \dot{h}(t_1) & \cdots & \dot{h}(t_N) \end{bmatrix}^T \\ &= D \begin{bmatrix} h(t_0) & h(t_1) & \cdots & h(t_N) \end{bmatrix}^T \end{aligned} \tag{4}$$

where the $(N+1) \times (N+1)$ differentiation matrix D is defined by

$$D_{ik} = \begin{cases} \frac{L_N(t_i)}{L_N(t_k)} \frac{1}{t_i - t_k}, & \text{if } i \neq k; \\ -\frac{N(N+1)}{4}, & \text{if } i = k = 0; \\ \frac{N(N+1)}{4}, & \text{if } i = k = N; \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The cost functional $J[x(\cdot), u(\cdot)]$ is approximated by the Gauss-Lobatto integration rule,

$$J[x(\cdot), u(\cdot)] \approx \sum_{k=0}^{N} g(\bar{x}^{Nk}, \bar{u}^{Nk}) w_k \tag{6}$$

where w_k is the LGL weight defined by

$$w_k = \frac{2}{N(N+1)} \frac{1}{[L_N(t_k)]^2}$$

The approximation is so accurate that it has zero error if the integrand function is a polynomial of degree less than or equal to $2N - 1$, a degree that almost doubles the number of nodes.

To summarize, a finite dimensional approximation of the state and control trajectories exist at the LGL nodes. The system of differential equations can be approximated using the discrete differentiation (4). The cost function can be approximated by the Gauss-Lobatto integration rule (6). Integrating these discretization elements together yelds a finite dimensional nonlinear programming, which can be solved numerically. In our computations, sequential quadratic programming is used. Details are referred to [5] and references therein.

IV. Minimum time trajectories

A MATLAB code of the HeLion model is integrated with the PS method of dynamic optimization. The program was, at first, applied to a simplified 14-dimensional helicopter model [6]. Some preliminary results were found that lead us to work on a more accurate 15-dimensional model. As a first testing problem, we only fly the UAV for a distance of 10 meters without any obstacle.

A search region must be clearly defined for the computational algorithm. Our first step was to configure the bounds for the variables and to set the number of nodes. The computations were based on 30 LGL nodes. Similar to that in [6], the range of the position of the helicopter in NED-frame is from -50 to 50 meters for the x and y coordinates and -50 to 0 in the z coordinate. We gave the range $(-1.5, 1.5)$ for pitch and roll in the body-frame as the helicopter should not flip upside down, and $2\pi\ rads$ for yaw to give complete freedom of direction. In the body-frame, we bounded the x velocity from -10 to 10 m/s and the y and z velocities from -5 to $5\ m/s$ and the rotational velocity from -1.0 to $1.0\ rad/s$. For the 13th and 14th variables we used -0.02 to $0.02\ rad$. The 15th variable was restricted in the range of -1 to 1. For the controls, we used the ranges in the manual of the model. These upper and lower bounds are summarized in Table III. The program was first applied to the 14-variable model [6]. However, the results are choppy with erratic controls. We upgraded the program to use the helicopter model of fifteen variables. The results we achieved are shown in Figures 1 through 4. To validate this trajectory of state and control variables, we ran a 4th order Runge-Kutta solver using the system model based on a spline interpolation of the optimal control at the nodes. The dots in the figures represent the results from the optimization program, while the lines represent the trajectory from Runge-Kutta method. The plots in Figure 1 show the trajectories of the position in NED-frame and the attitude in Euler angles. The velocity of the helicopter and its angular velocity are shown in Figure 2. The plots in Figure 3 show the controls of the UAV. The total time of the flight is

$$T = 2.883 \text{ sec.}$$

P_x: (-50,50)	P_y: (-50,50)	P_z: (-50,0)
u: (-10,10)	v: (-5,5)	w: (-5,5)
ϕ: (-1.5,1.5)	θ: (-1.5,1.5)	ψ: (-3.14,3.14)
p: (-1,1)	q: (-1,1)	r: (-1,1)
a_s: (-0.02,0.02)	b_s: (-0.02,0.02)	$\delta_{ped,int}$: (-1,1)
δ_{lat}: (-1,1)	δ_{lon}: (-1,1)	δ_{col}: (-1,0)
δ_{ped}: (-1,1)		

TABLE III

THE BOUNDS FOR THE STATE AND CONTROL VARIABLES

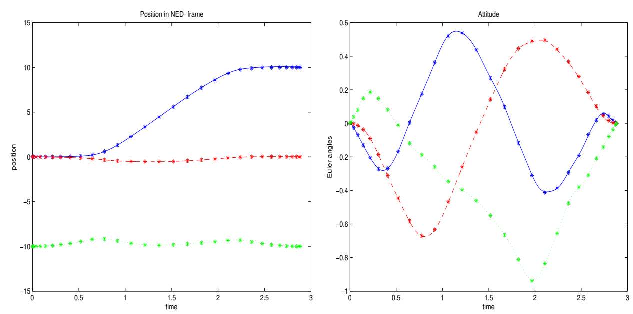

Fig. 1. x, ϕ -solid; y, θ - dash; z, ψ - dot.

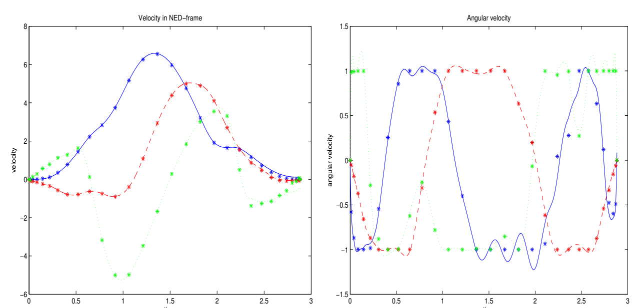

Fig. 2. u, p -solid; v, q - dash; w, r - dot.

The UAV flies a nonlinear path towards its destination with small movement in the y and z coordinates (in NED frame, $z < 0$). The UAV curves slightly to the right while it drops and rises twice back to its original height (10 meters above ground). The results for the controls are much smoother than the results we reached using the 14 variable model although control 3 still seems rather erratic. The variables a_s, b_s and $\delta_{ped,int}$ are shown in Figure 4. The trajectory is aggressive with complicated maneuvers. Although the shortest path between two points is a line, what we found is

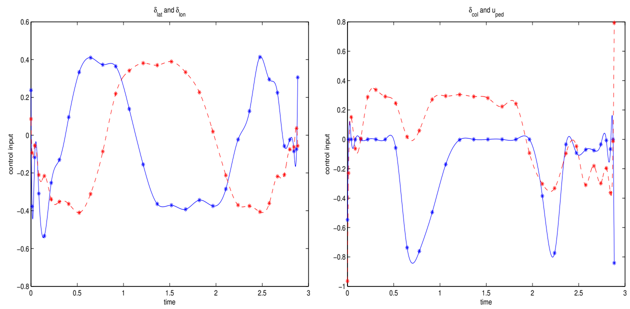

Fig. 3. δ_{lat}, δ_{col} -solid; δ_{lon}, δ_{ped} - dash.

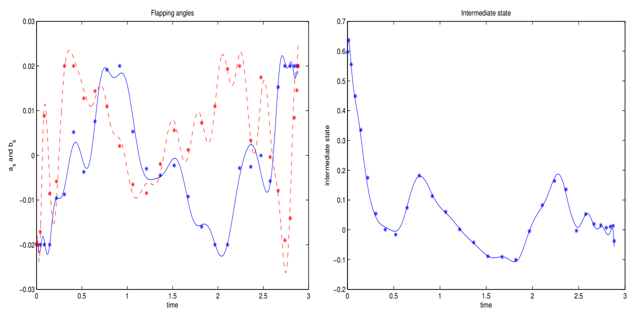

Fig. 4. a_s -solid; b_s - dash; $\delta_{ped,int}$ - solid.

that the minimum-time trajectory is anything but a straight line. This fact may sound counter intuitive. However, it is not surprising given the complex helicopter dynamics and its uneven distribution of forces around the body.

For the purpose of comparison, we apply additional constraints to force a path of straight line. In this case, we try to find the minimum-time controls subject to the constraints of constant $y = 0$ and $z = -10$. We used the following bounds for a straight line path. The results are shown in Figures 5-8. The total time for the flight is

$$T = 3.2538 \text{ sec}$$

which is about 13% more flying time than that of the optimal trajectory.

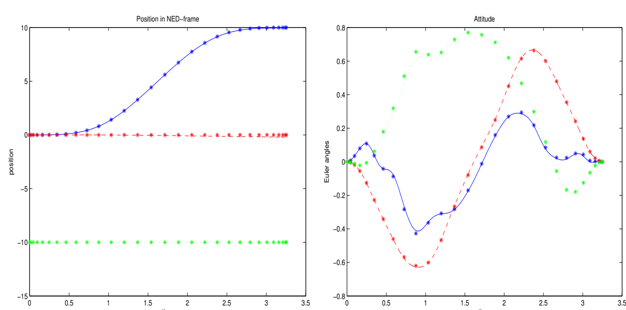

Fig. 5. x, ϕ -solid; y, θ - dash; z, ψ - dot.

We noticed that the optimal controls for a straight line path require erratic changes in the attitude of the UAV. In fact, the UAV is flying with a nonzero yaw angle for most of the

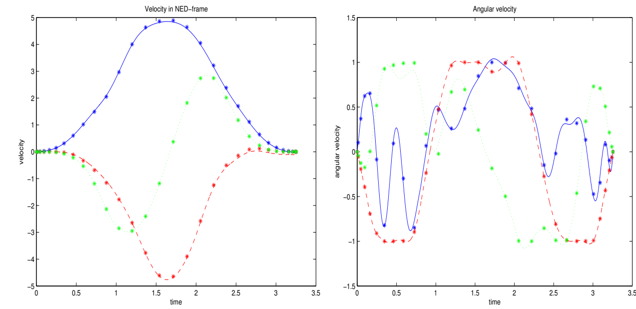

Fig. 6. u, p -solid; v, q - dash; w, r - dot.

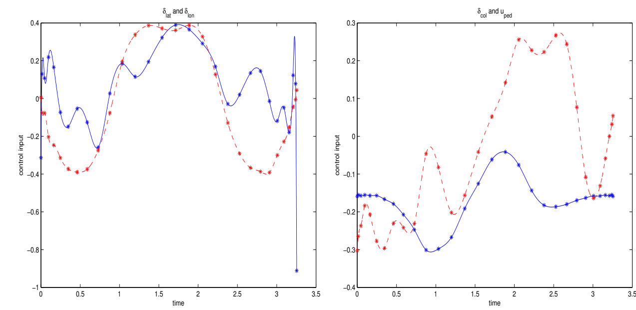

Fig. 7. δ_{lat}, δ_{col} -solid; δ_{lon}, δ_{ped} - dash.

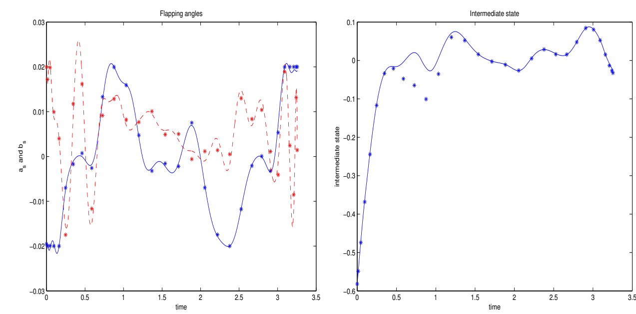

Fig. 8. a_s -solid; b_s - dash; $\delta_{ped,int}$ - solid.

time. We did another test to see how the helicopter would fly with zero yaw velocity, or simply, always try to point in the forward direction. In this case, the variable bounds are the same as the first simulation except that $r = 0$. The results are shown in Figures 9-12. In this case, the total flight time is

$$T = 3.1146 \text{ sec.}$$

Therefore, flying with zero yaw velocity yields a slightly faster time than flying in a straight line. Both took more than 10% of more time to complete than the optimal path. The validation of this path was very close, with some differences at the end but very minor.

V. Conclusions

In this research, a PS computational algorithm is applied to the model of HeLion, a Helicopter UAV. Various preliminary examples are used to test and verify the computation

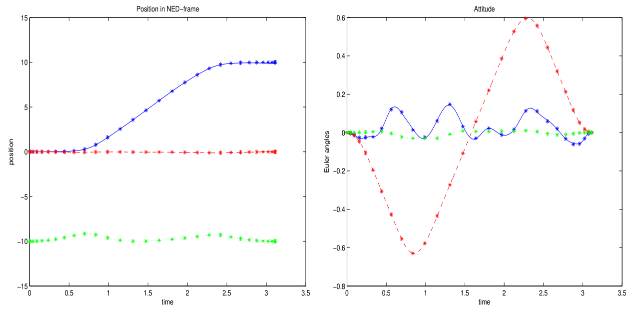

Fig. 9. x, ϕ -solid; y, θ - dash; z, ψ - dot.

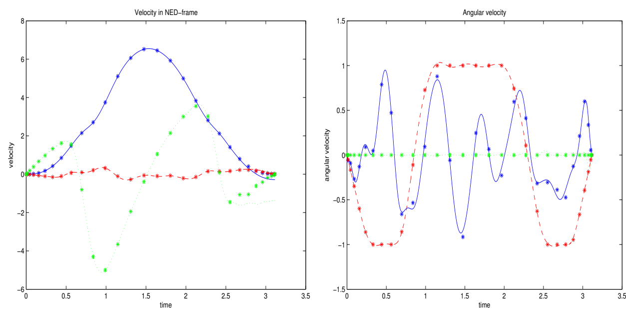

Fig. 10. u, p -solid; v, q - dash; w, r - dot.

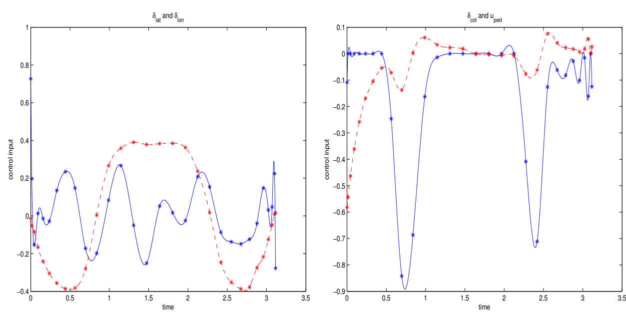

Fig. 11. δ_{lat}, δ_{col} -solid; δ_{lon}, δ_{ped} - dash.

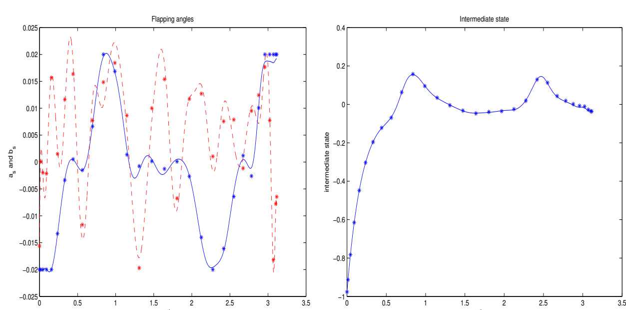

Fig. 12. a_s -solid; b_s - dash; $\delta_{ped,int}$ - solid.

program. These simulations show some new findings as well as questions for further research. First off, a minimum time trajectory requires highly nonlinear control maneuvering. The set of controls that generates the quickest path 10 meters forward is not a straight line, nor is it with zero yaw. The straight line type of paths require more than 10% of extra time. For future research, minimum time trajectories for a longer distance should be developed for meaningful real-life applications. Testing fight of the trajectories is planned. Paths with obstacles will also be considered in the next step of research. Real time application of a library of optimal trajectories is a major long term research goal.

REFERENCES

[1] R.Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, The MIT Press, 2004.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion; Theory, Algorithms, and Implementation*, The MIT Press, 2005.

[3] G. Cai, B. M. Chen and T. H. Lee, Unmanned Rotorcraft Systems, Springer, New York, 2011.

[4] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang. *Spectral Method in Fluid Dynamics.* New York: Springer-Verlag, 1988.

[5] W. Kang, Q. Gong, and I. M. Ross. On the convergence of nonlinear optimal control using pseudospectral methods for feedback linearizable systems. *International Journal of Robust and Nonlinear Control*, 2007, 17: 1251 – 1277.

[6] B. T. Gatzke, Trajectory Optimization for Helicopter Unmanned Aerial Vehicles (UAVs), NPS Thesis, 2010.