REAL-TIME CONSTRAINED TRAJECTORY GENERATION APPLIED TO A FLIGHT CONTROL EXPERIMENT

Mark B. Milam [*,1] **Ryan Franz** [**] **Richard M. Murray** [*]

[*] *Division of Engineering and Applied Science*
Control and Dynamical Systems
California Institute of Technology
Pasadena,CA 91125

[**] *Electrical and Computer Engineering*
University of Colorado, Boulder, CO 80309

Abstract: A computational approach to generate real-time, optimal trajectories for a flight control experiment is presented. Minimum time trajectories are computed for hover-to-hover and forward flight maneuvers. Instantaneous changes in the trajectory constraints that model obstacles and threats are also investigated. Experimental results using the Nonlinear Trajectory Generation software package show good closed-loop performance for all maneuvers. Success of the algorithm demonstrates that high-confidence real-time trajectory generation is achievable in spite of the highly nonlinear and non-convex nature of the problem.

Keywords: Trajectory planning, real-time, optimal control, optimization, nonlinear control, guidance systems

1. INTRODUCTION

Real-time trajectory generation in the presence of constraints is an important problem in the control of mechanical systems. Typical applications are obstacle avoidance of a robotic vehicle, minimum time interception of a maneuvering target by a missile, and a rapid change of trajectory for an Uninhabited Combat Air Vehicle (UCAV) to address a dynamic threat. Advanced control approaches are needed to achieve such missions (*Uninhabited Air Vehicles: Enabling Science for Military Systems*, 2000), and (McCall and J.A. Corder, 1996). The objective of this paper is to present a computational approach for real-time trajectory generation for class of problems and validate the solution on an experiment that represents a real-world application.

For our experiment, real-time trajectory generation will be combined with a feedback controller to form the *two degree of freedom* control system shown in Figure 1. In this paradigm, the trajectory generator provides a feasible feed-forward control and reference trajectory in the presence of system and environment constraints. Given inherent modeling uncertainty, a feedback controller (tracker) is necessary to provide stability around the reference trajectory. An advantage of this approach is that a stabilizing controller is provided with the feasible trajectory, not just the trajectory itself. Furthermore, it is possible to make the reference trajectory as aggressive as is allowed by the model.

A prime example of a case where the *two degree of freedom* design would be used is in the control of an UCAV. The desired objective of the UCAV would be commanded by the operator or pre-programmed without operator intervention. The UCAV would be required to generate trajectories in real-time and stabilize about them to meet the objectives. The objective might be to engage a target in a dynamically changing environment or to determine a trajectory that minimizes the cross-section of the UCAV in order to evade radar in a reconnaissance mission.

The Caltech ducted fan (Milam and Murray, 1999) is a scale model of a highly maneuverable UCAV and will be used as a test-bed to validate our real-time trajectory generation methodology. Our desired objective, similar to that of a real UCAV, will be to track real-time commanded positions and velocities of the ducted fan in minimum time. This must be achieved while ensuring that the test vehicle is stabilized and operating within all flight and actuation constraints.

Real-time trajectory generation for a ducted fan has been considered in (Nieuwstadt and Murray, 1998). In this case, differentially flat systems (Fliess *et al.*, 1999) with no constraints are considered. Additionally, no optimization criterion was used when computing the trajectories. (Faiz *et al.*, 2001) studies a system similar to the ducted fan, where constraints are approximated with linear segments and the problem cast into a linear programming problem. Applying this technique to systems with input constraints may be difficult. The reason is that deriving input constraints using the differential flatness formulation is frequently highly nonlinear and non-convex.

The primary contribution of this paper is to implement and validate the Nonlinear Trajectory Generation (NTG) software package described in (Milam *et al.*, 2000) on a real-time flight control experiment. NTG is a software package used for trajectory generation that combines elements of geometric control, B-splines, and nonlinear programming. In addition, it will be demonstrated that real-time trajectory generation is possible in

---
[1] Author to whom correspondence should be addressed, email: `milam@cds.caltech.edu`.

a dynamically constrained and non-convex environment. Finally, some insights on handling the convergence issues associated with using nonlinear programming techniques on-line will be presented.

The organization of the paper is as follows. The Caltech ducted fan and experimental setup will be described in Section 2. In Section 3, the dynamic model of the ducted fan will be presented. The formulation of the optimization problem and desired objective is provided in Section 4. In Section 5, a brief overview of NTG will be provide as well as the proposed methodology for trajectory generation. The timing and trajectory management is given in Section 6 and experimental results are given in Section 7.
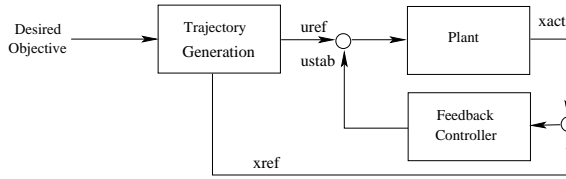

Fig. 1. Two Degree of Freedom Design.
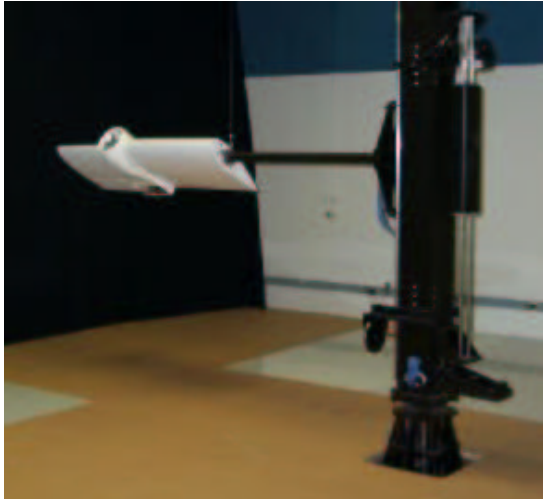
2. FLIGHT EXPERIMENT: THE CALTECH DUCTED FAN

Fig. 2. Caltech Ducted Fan Testbed.

The primary components of the Caltech ducted fan are shown in Figure 2. The aluminum stand and boom limit the operation of the ducted fan to a cylinder of radius $r_s = 2.35$ m and a height of 2.5 m. Revolute joints at the base of the stand and end of the boom provide two rotational degrees of freedom and a prismatic joint parallel to the stand provides a translational degree of freedom. All three of these directions are sensed with optical encoders.

A counterbalance system was required since the maximum thrust of the ducted fan engine is limited to 15 N while the weight of the boom and fan is $m = 12.5$ kg creating an "effective" gravity ($mg_{\text{eff}} = 7$ N). The unique feature of the counterbalance is the pulley system that provides a 4 to 1 gear ratio. That is, for every 1 m the boom and ducted fan move the counterweight moves 0.25 m. Gearing the system in this way allows us an

increase the maximum vertical acceleration of the ducted fan. The ducted fan's primary components are the wings and engine. In order that the system be able to fly in both directions, the wings and engine were designed to be symmetric. The ducted fan assembly houses a ducted fan and electric motor. Thrust vectoring of the ducted fan is provided by two aluminum paddles driven by PWM servos. Two joysticks provide the operator with a variety of interface options to the test-bed. All trajectory generation algorithms and control laws are hosted on four signal processors.

3. FLIGHT EXPERIMENT MATH MODEL

The configuration variables x and z represent the horizontal and vertical inertial translations, respectively, of the ducted fan while θ is the rotation of the ducted fan about the boom axis. The axis system located at the center of the ducted fan and rotating with it will be referred to as the body frame and be denoted by a b subscript.

$$
\begin{aligned}
m\ddot{x} + F_{X_a} - F_{X_b}\cos\theta - F_{Z_b}\sin\theta &= 0 \\
m\ddot{z} + F_{Z_a} + F_{X_b}\sin\theta - F_{Z_b}\cos\theta &= mg_{\text{eff}} \\
J\ddot{\theta} - M_a + \frac{1}{r_s}I_p\Omega\dot{x}\cos\theta - F_{Z_b}r_f &= 0,
\end{aligned} \quad (1)
$$

where $F_{X_a} = D\cos\gamma + L\sin\gamma$ and $F_{Z_a} = -D\sin\gamma + L\cos\gamma$ are the aerodynamic forces. See (Milam and Murray, 1999) for a complete derivation of the equations of motion. A spatial representation of the equations of motion is chosen in order that both hover and forward flight modes can be considered. F_{X_b} and F_{Z_b} are thrust vectoring body forces. I_p and Ω are the moment of inertia and angular velocity of the ducted fan propeller, respectively. $J = .25$ kgm^2 is the moment of inertia of the ducted fan and r_f is the distance from center of mass along the X_b axis to the effective application point of the thrust vectoring force. The angle of attack α can be derived from the pitch angle θ and the flight path angle γ by

$$\alpha = \theta - \gamma.$$

The flight path angle can be derived from the spatial velocities by

$$\gamma = \arctan\frac{-\dot{z}}{\dot{x}}.$$

The lift (L), drag (D), and moment (M) are given by

$$L = qSC_L(\alpha),\ D = qSC_D(\alpha),\ M = \bar{c}SC_M(\alpha),$$

respectively. The dynamic pressure is given by $q = \frac{1}{2}\rho V^2$. The norm of the velocity is denoted by V, S the surface area of the wings, and ρ is the atmospheric density. Figure 3 depicts the coefficients of lift ($C_L(\alpha)$) and drag ($C_D(\alpha)$) and the moment coefficient ($C_M(\alpha)$). These coefficients were determined from a combination of wind tunnel and flight testing.

4. OPTIMIZATION PROBLEM FORMULATION

A very aggressive optimization problem was chosen to be solved on-line: Minimize time (T) subject to the trajectory constraints $0 \le F_{X_b} \le F_{X_b}^{max}$, $F_{X_b}^{max}/2 \le F_{Z_b} \le -F_{X_b}^{max}/2$, and $z^{min} \le$
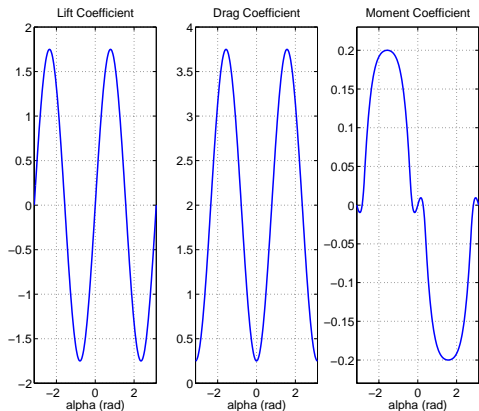
Fig. 3. B-spline curve fits wind tunnel and flight test results to the $C_L(\alpha)$, $C_D(\alpha)$, and $C_M(\alpha)$ aerodynamic coefficients).

$z \leq z^{max}$, the boundary constraints at the initial time

$$x(0),\ \dot{x}(0),\ \ddot{x}(0),\ z(0),\ \dot{z}(0),\ \ddot{z}(0),\ \theta(0), \dot{\theta}(0),\ \ddot{\theta}(0),$$

and boundary conditions at the final unknown time T

$$x(T), \dot{x}(T), z(T), \dot{z}(T), \theta(T), \dot{\theta}(T).$$

$F_{X_b}^{max} = 11$ N, $z^{min} = -1$ m, $z^{max} = 1$ m are the values of the constraints used for all the test results presented in this paper. In the *two degree of freedom* design, the force constraints were chosen in the trajectory generation to be conservative so that the stabilizing controller has some remaining control authority to track the reference trajectory. The reason that minimum time was chosen as the objective was to make the trajectories as aggressive as possible. The boundary constraints on the initial time accelerations provide us with smooth inputs in the case when a new trajectory is computed away from an equilibrium. Our final boundary condition will always be an equilibrium.

5. TRAJECTORY GENERATION METHODOLOGY

There are three components of the proposed trajectory generation methodology. The first is to determine a parameterization (output) such that Equation (1) can be mapped to a lower dimensional space (output space). (Fliess *et al.*, 1999) gives information on finding this mapping if the system if flat. The idea is to map dynamic constraints to algebraic ones. Once this is done the cost and constraints can also be mapped to the output space. The second is to parameterize each component of the output in terms of an appropriate B-spline polynomial. Finally, sequential quadratic programming is used to solve for the coefficients of the B-splines that minimize the cost subject to the constraints in output space. See (Milam *et al.*, 2000; Petit *et al.*, 2001) for more details on this approach. The NTG software package is an implementation of this concept. The user provides the cost and the constraints in terms of the outputs and their derivatives as well as the Jacobian of the cost and constraints with respect

to each output and the maximum derivative that occurs in each output.

By using this technique, a sufficient reduction can be made in the dimension of the nonlinear programming problem to make real-time computation possible. For the system under consideration, the outputs $z_1 = x(t)$, $z_2 = z(t)$, $z_3 = \theta(t)$, and $z_4 = T$ are chosen. Note that the system in Equation (1) is not necessarily differentially flat. An off-line comparison of computation times and probability of convergence from a random initial guess for several different outputs motivated our parameterization. By choosing this parameterization, one equality constraint will need to be satisfied over the entire trajectory. In the case of only forward flight, it would be possible to choose a parameterization that contains no equality constraints. See (Martin, 1996) for a complete discussion of this topic. In addition, a particular parameterization may also depend on the complexity of deriving the constraints from the outputs. However, generally it is recommended to use a parameterization that eliminates all equality constraints.

6. TIMING AND TRAJECTORY MANAGEMENT

Two different modes are considered in the experimental results: hover-to-hover and forward flight. These modes may also be combined. In the hover-to-hover mode the user commands a desired position x_d and z_d via the joystick positions. Every t_{sample} seconds a new minimum time trajectory is computed from the boundary conditions t_{sample} seconds into the future to the desired equilibrium position given by current position of the joysticks. Equilibrium is defined for the hover-to-hover mode as being the desired translational position, zero velocities, $\theta = \pi/2$, $F_{Z_b} = 0$, and $F_{X_b} = 7$ N. The forward flight mode is similar to the hover-to-hover mode except that the user commands the desired position in the vertical direction z_d and the desired spatial velocity $\dot{x}_d$. The equilibrium manifold if found by solving the resulting transcendental equations when $\dot{z} = 0$ in Equation (1) . A plot of the equilibrium manifold is shown in Figure 4. Figure 5 shows the timing
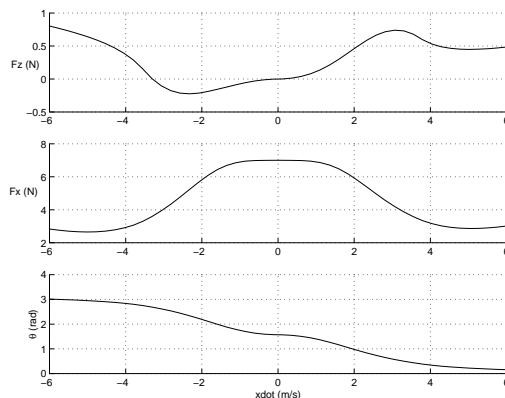

Fig. 4. Forward Flight Mode Equilibrium Manifold

scheme used in the experiment. A higher level management function controlling which trajectory to stabilize about is necessary since there is the possibility of the algorithm not converging as well

as excessive computation time in computing a trajectory. Before any optimizations have been
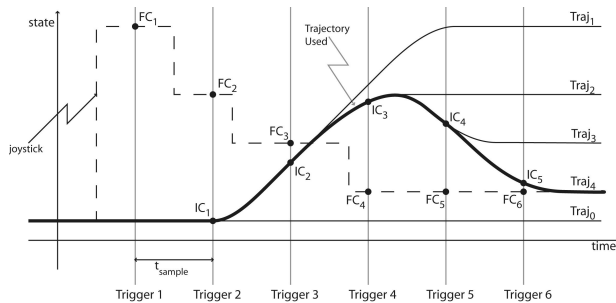

Fig. 5. Sample run showing joystick input and timing concepts. The initial conditions for each run are denoted with IC, and the final conditions with FC.

computed, a nominal equilibrium trajectory is used, denoted by Traj_0. The first optimization is provided with the state and inputs of this nominal trajectory t_{sample} seconds in the future as an initial boundary condition and the equilibrium condition indicated by the joystick positions as the final boundary condition. If the optimization has finished successfully before t_{sample} seconds, at $t = t_{\text{sample}}$ the resulting trajectory is used and another optimization is triggered in the same fashion. If the optimization takes longer than t_{sample} seconds, the trajectory is truncated in the first $t_{\text{runtime}} - t_{\text{sample}}$ seconds to attempt to maintain continuity in the trajectory, but a small discontinuity may occur. For this reason, t_{sample} should ideally be longer than the expected run times of NTG. An important point is that the value of t_{sample} is not a constant. The reason is as follows: the initial bound for the next optimization must lie on a point on the last accepted trajectory where the differential equations are exactly satisfied. This is enforced at 21 points along each trajectory, but because of the variable horizon length (due to minimum time), the spacing of the points in time varies. For this reason, t_{sample} is chosen to coincide with the closest enforcement point, within some nominal sample time.

Due to the nature of the trajectory generation methodology used in this experiment, the convergence to an optimal solution is not guaranteed. Because of this, higher-level management logic also has to decide whether to use a given trajectory computed by NTG. The most obvious criterion to accept a trajectory is an indication of convergence returned by NTG. Other criteria include an upper bound on the acceptable runtime. For example, if the runtime is more than 10 percent longer than t_{sample}, the current trajectory generation computation is aborted. If the decision is made to reject a trajectory, the last accepted trajectory continues to be used and another optimization is triggered as usual. If the existing trajectory is exhausted before another one is accepted, the final equilibrium condition is continued as long as necessary. In hover, this simply means that x and z are kept at the desired values and all velocities are zero; in forward flight, x is incremented with time according to the desired velocity and z is kept at the desired value.

7. EXPERIMENTAL RESULTS

In this section a description of the controllers and observers necessary for stabilization about the reference trajectory will be provided . Second, the NTG setup used for both the hover-to-hover and forward flight modes will be discussed. Finally, example trajectories using NTG for two flight modes on the Caltech Ducted Fan experiment are provided.

7.1 *Stabilization Around Reference Trajectory*

Although the reference trajectory is a feasible trajectory of the model, it is necessary to use a feedback controller to counteract model uncertainty. There are two primary sources of uncertainty in our model: aerodynamics and friction. Elements such as the ducted fan flying through its own wake, ground effects and thrust not modeled as a function of velocity and angle of attack contribute to the aerodynamic uncertainty. The friction in the vertical direction is also not considered in the model. The prismatic joint has an unbalanced load creating an effective moment on the bearings. The vertical frictional force of the ducted fan stand varies with the vertical acceleration of the ducted fan as well as the forward velocity. Actuation models are not used when generating the reference trajectory, resulting in another source of uncertainty.

The separation principle was kept in mind when designing the observer and stabilizing controller. Since only the position of the fan is measured, the choice was made to estimate the rates using an extended Kalman filter. The gains were scheduled on the forward velocity.

The stabilizing LQR controllers were gain scheduled on pitch angle (θ) and the forward velocity. The weights were chosen differently for the hover-to-hover and forward flight modes. For the forward flight mode, a smaller weight was placed on the horizontal (x) position of the fan compared to the hover-to-hover mode. Furthermore, the z weight was scheduled as a function of forward velocity in the forward flight mode. There was no scheduling on the weights for hover-to-hover. The elements of the gain matrices for both the controller and observer are linearly interpolated over 51 operating points.

7.2 *Nonlinear Trajectory Generation Parameters*

In Section 4, the optimal trajectory generation problem we intended to solve was outlined. The three outputs $z_1 = x$, $z_2 = z$, and $z_3 = \theta$ will each be parameterized with four (intervals) , sixth order, C^4 (multiplicity), piecewise polynomials over the time interval scaled by the minimum time. The last output ($z_4 = T$), representing the time horizon to be minimized, is parameterized by a scalar. Choosing the outputs to be parameterized in this way has the effect of controlling the frequency content of inputs. Since the actuators are not included in the model, it would be undesirable to have inputs with a bandwidth higher than the actuators. There are a total of 37 variables in this optimization problem. The trajectory constraints are enforced at 21 equidistant breakpoints over the scaled time interval.

There are many considerations in the choice of the parameterization of the outputs. Clearly there is a trade between the parameters (variables, initial values of the variables, and breakpoints) and measures of performance (convergence, runtime, and conservative constraints). Extensive simulations were run to determine the right combination of parameters to meet the performance goals of our system.

7.3 *Forward Flight*

To obtain the forward flight test data, the operator commanded a desired forward velocity and vertical position with the joysticks with $t_{\text{sample}} = 2.0$ sec. By rapidly changing the joystick positions, NTG produces high angle of attack maneuvers. Figure 6 depicts the reference trajectories and the actual θ and $\dot{x}$ over 60 sec. Figure 7 shows the commanded forces for the same time interval. The sequence of maneuvers in this plot are the following: First, the ducted fan transitions from near hover to forward flight. Second, the ducted fan is commanded from a large forward velocity to a large negative velocity. Finally, the ducted fan is commanded to go to hover. Figure 8 is an illustration of the ducted fan altitude and x position for these maneuvers. The air-foil in the figure depicts the pitch angle (θ). It is apparent from this figure that the stabilizing controller is not tracking well in the z direction. This is due to the fact that unmodeled frictional effects are significant in the vertical direction. It is believed that this could be corrected with an integrator in the stabilizing controller. Figure 9 shows the run times for the 30 trajectories computed in the 60 second window. The average computation time is less than one second. Each of the 30 trajectories converged to an optimal solution and was approximately between 4 and 12 seconds in length. A random initial guess was used for the first NTG trajectory computation. Subsequent NTG computations used the previous solution as an initial guess. Much improvement can be made in determining a "good" initial guess. Improvement in the initial guess will improve not only convergence but also computation times.
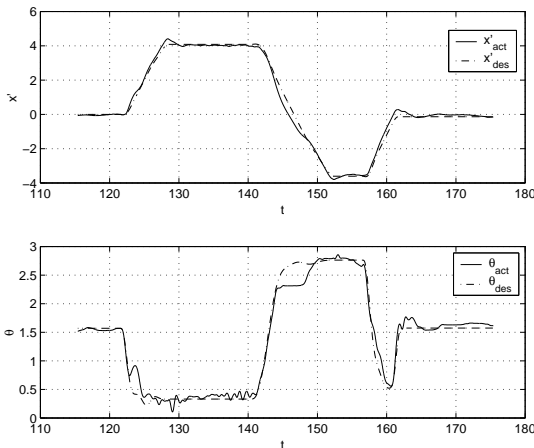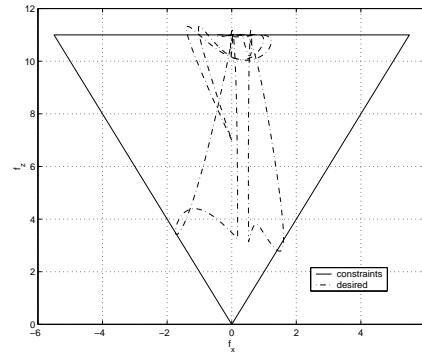
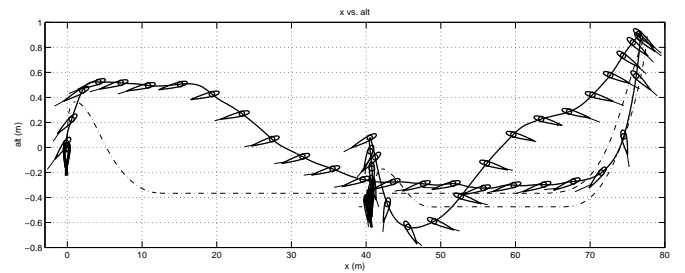Fig. 7. Forward Flight Test Case: Desired F_{X_b} and F_{Z_b} with bounds.

Fig. 8. Forward Flight Test Case: Altitude and x position (actual (solid) and desired(dashed)). Airfoil represents actual pitch angle (θ) of the ducted fan.
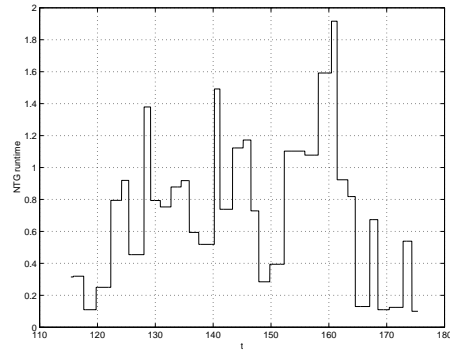
Fig. 9. Forward Flight Test Case: 60 second run, 30 computed trajectories, $t_{\text{sample}} = 2$ sec.

7.4 *Hover-to-Hover*

To obtain hover-to-hover test data, the operator commanded a desired horizontal and vertical position with the joysticks with $t_{\text{sample}} = 1.0$ sec. By rapidly changing the joystick positions, NTG can produce very aggressive trajectories. The top plot in Figure 10 shows z and x positions of the ducted fan and the trajectory constraints. The maneuvers are created by holding the commanded z constant and changing the commanded x by the following sequence: $0 \mapsto 7.5 \mapsto 15 \mapsto 7.5 \mapsto 0$. (Note: these commanded positions are approximate.) These maneuvers were done over a time period of 60 seconds with a computation time on average of .7 seconds for 4 to 8 seconds of trajectory. Each of the 60 trajectories converged to a locally optimal solution. The bottom plot in Figure 10 corresponds to approximately the same changes in x but with some terrain added in real-time. There was no new initial guess provided

Fig. 6. Forward Flight Test Case: θ and $\dot{x}$ desired and actual.

to NTG when it was required to solve this optimization problem. The new terrain profile was also not tested off-line. These results give a reasonable argument for computing the trajectories on-line. It would be difficult to store trajectories for unknown threats and changes in terrain. All but one of the 60 trajectories converged to a local optimal solution. The one that did not converge was a result of asking the ducted fan to move to a position in violation of the trajectory constraints. The average computation time of each trajectory was approximately .8 seconds for 4 to 9 seconds of trajectory.
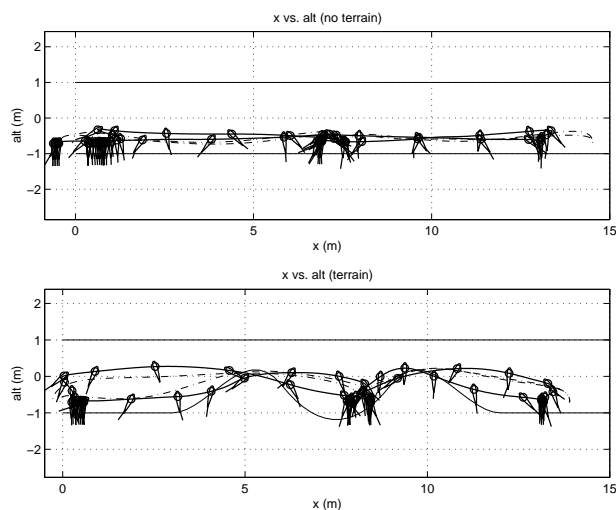

Fig. 10. Hover to Hover Test Case: Altitude and x position for two different vertical trajectory constraints (actual (solid) and desired (dashed)). Airfoil represents actual pitch angle (θ) of the ducted fan.

8. CONCLUSIONS AND FUTURE WORK

In this paper a methodology for real-time trajectory generation and validation of this approach with experimental results was presented. It was demonstrated that minimum time constrained trajectory generation is possible in real-time for two different flight modes on the Caltech ducted fan. In addtion, it was illustrated that dynamically changing trajectory constraints can be taken into account in real-time.

For both the forward flight and the hover-to-hover test cases it was always assumed that the ducted fan could track the reference trajectory. Recall that the initial state of the reference trajectory starts from a point on the previous reference trajectory, not from the actual position of the fan. There may be circumstances in which the ducted fan cannot track the reference trajectory. In this case, one may want to update the reference trajectory using the current state of the ducted fan. This motivates considering a model predictive control approach as in (Dunbar et al., 2002).

Developing a high confidence hierarchical control scheme is a direction of future research. In this paper, confidence is achieved in our trajectory generation routine by defining a set of logic to manage the output from NTG. Standard measures of convergence and confidence need to be developed for on-line systems hosting algorithms that are not guaranteed to converge.

Along the same lines of a hierarchical control scheme is to develop different levels of trajectory generation. In our tests, it was noticed that trajectories could be any length from 1 sec to 25 sec with the same number of variables for each trajectory. It may be useful to have NTG determine trajectories using a kinematic model of the ducted fan at a high level and then determine trajectories at a lower level using a dynamic model. By doing this, it could be possible to get a more consistent length of trajectory for each computation.

Another topic for future research would be further development of on-line trajectory generation tools such as NTG. Developing a sequential quadratic programming routine designed specifically to run in real-time is a research goal. A sequential quadratic programming technique that incorporates an analytical Hessian and/or is based on the Interior Point Method are potential candidates to get reduced run-times and improved rates of convergence. B-splines are only one possibility of basis functions to use to parameterize the outputs. There may be other basis functions, such as rational B-splines, that better span the trajectory space of a system than B-splines and are more suitable for real-time computations.

9. ACKNOWLEDGMENT

10. REFERENCES

Dunbar, W.B., M.B. Milam, R. Franz and R. M. Murray (2002). Model predictive control of a thrust-vectored flight control experiment. In: *15th IFAC World Congress.*

Faiz, N., S. Agrawal and R. M. Murray (2001). Trajectory planning of differentially flat systems with dynamics and inequalities. *Journal of Guidance, Control, and Navigation* **24**(2), 219–227.

Fliess, M., J. Lévine, P. Martin and P. Rouchon (1999). A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Trans. Auto. Cont.* **44**(5), 928–937.

Martin, Philippe (1996). Aircraft Control Using Flatness. In: *Computational Engineering in Systems Applications.* pp. 194–199.

McCall, G.H. and editors J.A. Corder (1996). *New World Vistas: Air and Space Power for the 21st Century.* United States Air Force.

Milam, M. B. and R. M. Murray (1999). A testbed for nonlinear control techniques: The Caltech Ducted Fan. In: *1999 IEEE Conference on Control Applications.*

Milam, M. B., K. Mushambi and R. M. Murray (2000). A new computational approach to real-time trajectory generation for constrained mechanical systems. In: *IEEE Conference on Decision and Control.*

Nieuwstadt, M. and R.M. Murray (1998). Real time trajectory generation for differentially flat systems. *Journal of Robust and Nonlinear Control* (11), 995–1020.

Petit, N., M. B. Milam and R. M. Murray (2001). Inversion based constrained trajectory optimization. In: *5th IFAC symposium on nonlinear control systems.*

Uninhabited Air Vehicles: Enabling Science for Military Systems (2000). National Academy Press.