Autonomous Navigation of Generic Quadrocopter with Minimum Time Trajectory Planning and Control

Kumar Bipin, Vishakh Duggal and K.Madhava Krishna

Abstract— **The challenges in generating minimum time trajectory and control for generic quadrocopter flying through sophisticated and unknown environment are explored in this paper. The proposed method uses *convex programming* technique to optimize polynomial splines, which are numerically stable for high-order including large number of segments and easily formulated for efficient computation. Moreover, exploiting the differential flatness of system, these polynomial trajectories encode the dynamics and constraints of the vehicle and decouple them from trajectory planning. The framework is fast enough to be performed in real time and results in solution which is close to *time optimal*. As control inputs are computed from the generated trajectory in each update, they are applicable to achieve closed-loop control similar to model predictive controller.**

I. INTRODUCTION

Recent years have seen considerable advances in small unmanned aircraft like quadrocopter. Due to their agility, mechanical simplicity and robustness, which enable these vehicles to fly through dense, cluttered spaces of real-world environment and within vicinity of humans.

Quadrocopter with commercial applications are entering in the market. Amazon made the news with its *Prime Air* [1] unmanned delivery service. By the same token, the La Fondation Bundi, which sponsors the *Flying Donkey Challenge* [2], a forum that works to attracts tech-entrepreneurs to develop commercial drone those can deliver packages through African sky.

These opportunities and advances in techniques , together motivate to address the challenges in autonomous navigation. Hence, universal framework which could efficiently generate trajectories and control for any commercial quadrocopter [3] flight through real-world environment is proposed.

However, a considerable number of contributions exists on modeling [4], design [5], control [6], [7] and trajectory generation [8]–[10] for quadrocopter. The research by Daniel Mellinger [11] shows the trajectory generation with minimum jerk in tightly constrained indoor environment. Similarly, another work by Mark Cutler [5] addresses an algorithm which fits a time-parametrized polynomial through a number of way-points and results in closed-form solution, if the corresponding way-point arrival *time are known a prior*. In both of these works trajectory generation is tightly coupled with complex vehicle dynamics and constraints. Additionally, experiments had been carried out in laboratory with motion capture system. Moving ahead in the direction of autonomous navigation, a remarkable paper by Charles Richter [8] explores the challenges of planing trajectory through complex environment, his derivations also requires an *a prior selection of the time* to traverse between one way-point and the next. Moreover, the framework proposed in our current work addresses these issues of "*a prior knowledge of the time*" with help of an objective function formulated as *convex optimization problem* and provides solution which is close to time optimal.

The authors are with Robotics Research Lab, IIIT Hyderabad-500032, AP, India, mkrishna@iiit.ac.in,{kumar.bipin vishakh.duggal}@research.iiit.ac.in.

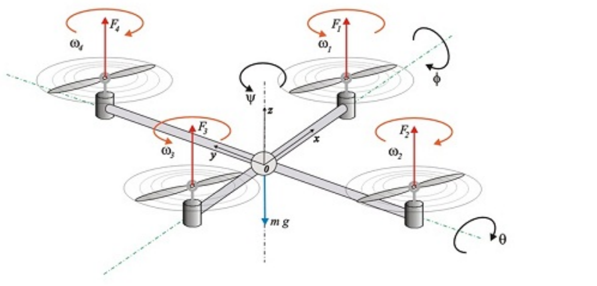

Fig. 1: Dynamic model of a quadrocopter, acted upon by gravity g, a thrust force F_i pointing along the body-fixed axis z; and rotating with angular rate ($\omega = \omega_1, \omega_2, \omega_3$), with its position in inertial space given as (x, y, z).

The major contribution of this paper would be in development of a generic trajectory planning control framework that facilitates planning from any initial state-position, velocity and acceleration to a target state while ensuring the following objectives to be accomplished:

- The trajectories must be feasible under the dynamics and input constrain of the quadrocopter described in Section II (C^2 continuity with bounded jerk).
- The generated trajectories and control should drive the vehicle to target state in minimum time. In Section III, the formulation of *convex objective function* with respective constraints would be discussed.
- The proposed framework does not require any numerical integration at any stage and also avoids iterative optimization approach.
- The computation cycle of framework must be fast enough to be applied in real-time at each control update similar to model predictive control. The experiment and results discussed in Section IV demonstrate its practicality.

II. QUADROCOPTOR DYNAMICS AND CONTROL ALGORITHM

In this section high level overview of proposed framework would be presented. The planning problem (details in: Section III) has been split into two parts. First, trajectory($s(u)$)

containing no time information is calculated using class of motion primitives-B-Spline. Following that, the resulted trajectory is parametrized in time by maximizing its velocity profile subject to dynamic feasibility constraints-velocity, acceleration and jerk are met.

The planning is carried out using generic control parameters-velocity, acceleration etc, which may not be the actual control inputs for the vehicle. Therefore, specific control inputs complaint with vehicle under consideration must be derived from the generic control parameters. For example, according to data sheet and user manual available for commercially available quadrocopter *Ardrone 2.0* [3], which has been used for experimental setup, the control inputs are roll (θ), pitch (ϕ) and $\ddot{z}$ for motion along y, x and z axis receptively and ψ for rotation along vertical axis as shown in ”Fig. 1”. A differential flat representation of quadrocopter [6] provides an analytical mapping from generated trajectory and its derivatives to the vehicle’s states and control inputs which are required to follow a planned trajectory.

Moreover, a quadrocopter is described by six degree of freedom of the rigid body as $q = [x, y, z, \theta, \phi, \psi]^T$, where the triple (x, y, z) represents the position of the center of mass of quadrocopter (in Earth inertial reference frame) and ”roll-pitch-yaw” (θ, ϕ, ψ), set of Euler angles which represents the orientation in the same reference frame. The dynamic model [6] can be derived by following the Lagrangian approach:

$$\ddot{x} = U_1 \frac{(\cos\psi \cos\phi \sin\theta + \sin\psi sin\phi)}{m} \tag{1}$$

$$\ddot{y} = U_1 \frac{(\sin\psi \cos\phi \sin\theta - \sin\phi cos\psi)}{m} \tag{2}$$

$$\ddot{z} = U_1 \frac{\cos\theta cos\phi}{m} - g \tag{3}$$

After simplification and eliminating $\frac{U_1}{m}$ from equations (1),(2) and (3), control commands for system describe above could be derived as follows:

$$\phi = \frac{\arctan(\ddot{x}\cos\psi + \ddot{y}sin\psi)}{\ddot{z}+g} \tag{4}$$

$$\theta = \frac{\arctan(\ddot{x}\sin\psi - \ddot{y}cos\psi)}{\ddot{z}+g} \cos\phi \tag{5}$$

for motion along y and x axis respectively where $\ddot{z}$ is another control command for height. The proposed Navigation Framework works with the assumption of fixed heading angle $\psi = 0$.

The trajectory planning procedure is fast enough to be performed for every control update. Therefore, a feedback loop is closed by re-planning the entire trajectory on each control update after dispatching the control commands of previous control interval to the quadrocopter as shown in ”Fig. 2”. The method is similar to model predictive control where an optimum trajectory is generated at each time step. Since actual commands to the underlying inner control loops are generated in discrete time interval, a numerical average of the control inputs over a typically small interval(experimentally determined) is computed and this is commanded to the underlying inner control loops. As stated in Section IV, this method has shown to work well in experiments.

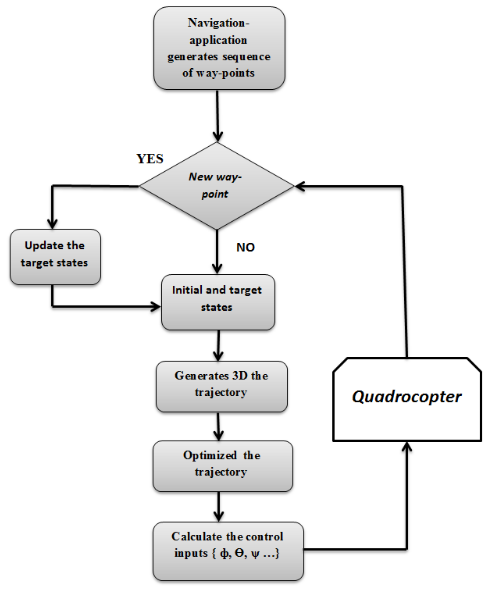
Navigation-application generates sequence of way-points
New way-point
YES
NO
Update the target states
Initial and target states
Generates 3D the trajectory
Optimized the trajectory
Calculate the control inputs { ϕ, Θ, ψ ...}
Quadrocopter

Fig. 2: Closed-loop control for quadrocopter using tarjectory planning

”Fig. 2” delineates the flow chart of closed-loop control using the trajectory planning. At every controller update the initial states of planning problem are updated whereas update of target states depends upon arrival of new way-point from sequence of way-points provided by Navigation Application.

III. TRAJECTORY GENERATION AND OPTIMIZATION

In this section, the trajectory planning problem has been formulated as convex optimization problem where an objective function which has to be minimized is defined in parametric form, containing no time information. The proposed method could be used to generate trajectory and controls for any arbitrary type of vehicle but explicitly modified for quadrocopter as describe in Section II. However, the vehicle dynamics define the consideration and applicability of various constraints which would be discussed at various point in the text.

A. Synthesis of the Trajectory

As describe in previous section, proposed framework assume that initial and target states are provide by any Navigation Application,where current state corresponds to *initial state* $X_0 = [x_0, \dot{x}_0, \ddot{x}_0]$ and sequence of way-points define the *target or final states* $X_f = [x_f, \dot{x}_f, \ddot{x}_f]$. The generation of trajectory begins with Bezier curve which interpolates $n+1$ points $(q_0, \ldots, q_n)$ from initial to target position while ensuring end conditions are met.

$$q_i(\tau) = \sum_{i=0}^{n} \binom{n}{i} \tau^i (1-\tau)^{n-i} p_i, \forall \tau = 0, \ldots, 1 \tag{6}$$

where $\binom{n}{i}$ are binomial coefficient and $\binom{n}{i}\tau^i(1-\tau)^{n-i}$ are the *Bezier/Bernstein* basis polynomials, and p_i are scalar

Preprint submitted to 2014 IEEE International Conference on Vehicular Electronics and Safety. Received October 15, 2014.

coefficient called *control points* calculated by solving the equation (6) with boundary conditions (i.e. initial and target states).

B. Spline Representation of Motion Primitive

Furthermore, $n+1$ points $(q_0,\ldots,q_n)$ along the generated trajectory, are interpolated by B-Spline polynomial with $m+1$ de Boor control points:

$$s(u)=\sum_{j=0}^{m} P_j B_j^p(u), \quad u_{min}\le u\le u_{max} \tag{7}$$

where P_j are the scalar de Boor control points, B_j^p are the B-spline basis functions of degree p. Bezier polynomial with strict relation between curve degree and number of control points made it inapt for our formulation. B-Spline polynomial a generalization of Bezier polynomials overcomes this limitation and with strong convex hull and inherent continuity (C^{p-1}) properties, made it an optimum choice for our formulation. To avoid high force and moments on quadrotor during motion, jerk continuous quintic B-spline($p=5$) are used. The knot vector initially chosen to define B-Spline are calculated using "cord length distribution" [12]. To generalize the analysis, a non dimensional time variable, u, is defined as

$$t_i=\lambda_i\cdot u_i, \qquad \forall i=0,\ldots,n \tag{8}$$

where $u=[u_0,\ldots,u_n]$ and λ_i is a scalar time scaling variable. The B-Spline function "s" is then defined as:

$$s(t_i)=s(\lambda_i\cdot u_i), \qquad \forall i=0,\ldots,n \tag{9}$$

After, differentiating the equation (9), velocity, acceleration and jerk profile of motion are obtain:

$$v(t_i)=\dot{s}(t_i)=\frac{\dot{s}(u_i)}{\lambda_i}=\sqrt{\dot{s}_x(t_i)^2+\dot{s}_y(t_i)^2+\dot{s}_z(t_i)^2} \tag{10}$$

$$a(t_i)=\ddot{s}(t_i)=\frac{\ddot{s}(u_i)}{\lambda_i^2}=\sqrt{\ddot{s}_x(t_i)^2+\ddot{s}_y(t_i)^2+\ddot{s}_z(t_i)^2} \tag{11}$$

$$j(t_i)=\dddot{s}(t_i)=\frac{\dddot{s}(u_i)}{\lambda_i^3}=\sqrt{\dddot{s}_x(t_i)^2+\dddot{s}_y(t_i)^2+\dddot{s}_z(t_i)^2} \tag{12}$$

where $v(t_i)$, $a(t_i)$ and $j(t_i)$ are the velocity, acceleration and jerk of quad-rotor at time t_i. $\dot{s}_x(t_i)$, $\ddot{s}_x(t_i)$, $\dddot{s}_x(t_i)$ and similarly others represent individual velocity, acceleration and jerk components along each axis.

The following section describes the problem formulation of minimum time trajectory generation as a convex optimization problem where position, velocity , acceleration and jerk of motion constitutes the respective constraints.

C. Objective Function

Following the philosophy described in [10], for calculation of a minimum time motion profile, the proposed method must bring velocity profile $v(t)$ of the system as close possible to maximum velocity limit v_{max} subject to maximum acceleration a_{max} and maximum jerk j_{max} constraints provided by physical limit of vehicle dynamics. Following, resulting objective function needs to be minimized is given by :

$$\Omega=\sum_{i=0}^{n}\left[\frac{\dot{s}(u_i)}{\lambda_i}-v_{max}\right]^2 \tag{13}$$

Fig. 3: Ardrone [3] performing 3D collision avoidance with obstacle (orange cone) by generating trajectories on-line through a sequence of collision free way-points. ArTags [18] were placed at strategic(predefined) location in flying area of the lab.

Furthermore, the equation (13) can be expressed as a standard form of quadratic function:

$$f_o(\chi)=\chi^T P\chi+q^T\chi+r \tag{14}$$

where $\chi=\left[\frac{1}{\lambda_0},\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_n},\right]$ is vector of optimization variables.

D. Convex Constraints

Given initial and target states, introduce equality constraints derived from both initial and final position, velocity and acceleration. In addition, for ensuring the feasibility of generated trajectory, it is necessary to introduced constraints from physical limit of vehicle dynamics and also from operating environment (i.e. maximum velocity limit for indoor and outdoor operation). Specially for quadrocopter applying constraints on acceleration and jerk would be necessary for smooth motion. The constraints mentioned above can be expressed by affine functions of χ in the form of equality and inequality constraints:

$$A_{eq}\chi=b_{eq} \tag{15}$$

$$A_{in}\chi\preceq b_{in} \tag{16}$$

In following paragraph these introduced constraints would be described in detail and emphasized those specific to quadrocopter.

1) Initial and Final States: As stated above, position, velocity and acceleration are known for initial and final state provided by Navigation Application ,which introduces 14 equality constraints. Those are included in the matrix A_{eq} and b_{eq}.

2) Velocity: Similarly, velocity level constraints are evaluated at each points $(q_1,\ldots,q_{n-1})$, which are normally specified according to safety norm of operating environment-indoor or outdoor, usually kept lower than the maximum limit. These introduce $n-1$ constraints are included in matrix A_{in} and b_{in}.This condition reads as :

$$v(t_i)=\frac{\dot{s}(u_i)}{\lambda_i}\le V_{max}, \quad \forall\, i=0,\ldots,n. \tag{17}$$

3) Acceleration: For a quadrocopter, the collective thrust as shown in "Fig. 1" $(F_1+F_2+F_3+F_4)$ is limited by minimum and a maximum thrust valued, denoted by f_{min} and f_{max} respectively. These conditions could be represented as:

$$f_{min}\le\sqrt{(\ddot{x})^2+(\ddot{y})^2+(\ddot{z}+g)^2}\le f_{max} \tag{18}$$

which produces a quadratic constrain. For seek of simplicity and computational efficiency of the problem, this constrain is decoupled by allocating a constant maximum allowable acceleration magnitude to each co-ordinate separately ($a_{max,x}$, $a_{max,y}$, $a_{max,z}$) [7]. These maximum acceleration are selected such that they fulfill the following constraints:

$$\sqrt{(a_{max,x})^2 + (a_{max,y})^2 + (a_{max,z})^2} \leq a_{max} \tag{19}$$

$$g + a_{max,z} \geq a_{min} \tag{20}$$

These are compliant to the affine formulation presented above in equation (16) and contributes $(3 \times n - 1)$ additional inequality constrain to matrix A_{in} and b_{in}.These inequalities are represented as:

$$\ddot{s}_l(t_i) = \frac{\ddot{s}_l(u_i)}{\lambda_i^2} \leq a_{max,l} \quad l \in \{x,y,z\}, \forall i = 0, \ldots, n. \tag{21}$$

4) Jerk: Following the approach describe in [7], upper bound on the allowable jerk per axis ($j_{max,x}$, $j_{max,y}$, $j_{max,z}$) is derived as:

$$j_{max,l} = \frac{1}{\sqrt{3}} f_{min} \omega_{max}, \quad l \in \{x,y,z\} \tag{22}$$

where $j_{max,l}$ is upper bound on the allowable jerk per axis, f_{min} the minimum collective thrust of quadcopter and maximum angular rate of rotation ω_{max} as described in [7]. In the worst case, all three axes produced the maximum allowable jerk j_{max} at the same time as the minimum thrust f_{min} is achieved. These add another $(3 \times n - 1)$ constrain to the problem and are included in matrix A_{in} and b_{in}.Inequalities for jerk are represented as:

$$\dddot{s}_l(t_i) = \frac{\dddot{s}_l(u_i)}{\lambda_i^3} \leq j_{max,l} \quad l \in \{x,y,z\}, \forall i = 0, \ldots, n. \tag{23}$$

In current section formulation of optimization problem with respective constraints was discussed which facilitates the generation of minimum time trajectory. The overall optimization problem is represented as :

$$\begin{aligned} &\text{minimize} && \Omega \\ &\text{subject to :} && A_{eq}\chi = b_{eq} \\ & && A_{in}\chi \preceq b_{in} \end{aligned}$$

In continuation a noble method for generating trajectory close to time optimal is described in next section.

E. Solving Optimization Problem for Minimum Time Trajectory

Optimization problem (13) with equality and inequality constraints (15), (16) is quadratic and convex hence it converges to global optima extremely fast. Which results in a vector of *optimal* time scaling parameters $\chi = \left[\frac{1}{\lambda_0}, \frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_n},\right]$. These are substituted in equation (8) to obtain minimum scaled time $t_0, \ldots, t_n$ for the planned trajectory. Thus, total time of trajectory:

$$T = \sum_{i=0}^{n} \alpha_i$$

where $\alpha_i = t_{i+1} - t_i$ and T is feasible [19] *minimum trajectory time*,calculated from trajectory optimization whereas similar work [8], [11] require estimate of T as priori for trajectory generation. There is no inherent relation between $\frac{1}{\lambda_i}, \frac{1}{\lambda_j}; \forall i \neq j$, hence may cause uneven stretching of time interval α_i. Therefore dynamic time scaling may cause velocity, acceleration or jerk discontinuity at joints of B-Spline segments [13], [14]. To remove the incidentally introduced discontinuities B-spline trajectory parameters have to be recomputed (i.e. de Boor control points with time parameters $t_0, \ldots, t_n$). The proposed framework solved this by using *Squared Distance Minimization* approach where

- The end points are exactly interpolated i.e. $q_0 = s(t_0)$ and $q_n = s(t_n)$.
- The internal points q_i, $\forall i = 1, \ldots, n-1$ as calculated in Section (III-A),are approximate in the least square sense, by minimizing the optimization function

$$Q = \sum_{i=0}^{n} [q_i - s(t_i)]^2 \tag{24}$$

The equation (24) can be expressed as a standard form of quadratic function:

$$f_o(X) = X^T P X + q^T X + r \tag{25}$$

where $X = [p_0, p_1, \ldots, p_m,]$ is vector of optimization variables (corresponds to de Boor control points). The constraints described in previous sections (v_{max}, a_{max} and j_{max}) can be expressed as affine function of X in form of equality and inequality constrain as:

$$A_{eq} X = b_{eq} \tag{26}$$

$$A_{in} X \preceq b_{in} \tag{27}$$

Finally, the solution of optimization problem- quadratic and convex-with respective equality and inequality constraints described above provides a *minimum time* trajectory which is close to *time optimum*. Time optimal motions are characterized by saturation of either the velocity or acceleration components at any given instant. As can be seen from the Fig. 4, that the acceleration components are saturated for significant amount of time. The performance can be improved by increasing the resolution of discretization of the optimization problem, although at the cost of increased computation time. Thus, a trade-off needs to be achieved between real time performance and time optimality.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

For all experiments, state estimation and purposed framework computations were carried out on a conventional desktop computer running Ubuntu 12.04 LTS and equipped with an Intel Core i3 processor @ 2.4 GHz and 4GB of RAM.Experiments were performed within indoor lab having dimensions of 10mX10mX6m. ArTags were placed at strategic (predefined) location in flying area of the lab and works as a alternative solution for Motion Capture System. Communication between the desktop and quadcopter was over Wi-Fi 2.4 GHz link. The quadcopter transfers on-Board IMU data (velocity, acceleration and orientation) and video feed of front primary camera over Wi-Fi link to state estimation software and control framework @20Hz. Using ARToolKit library, ArTags are detected within video feed

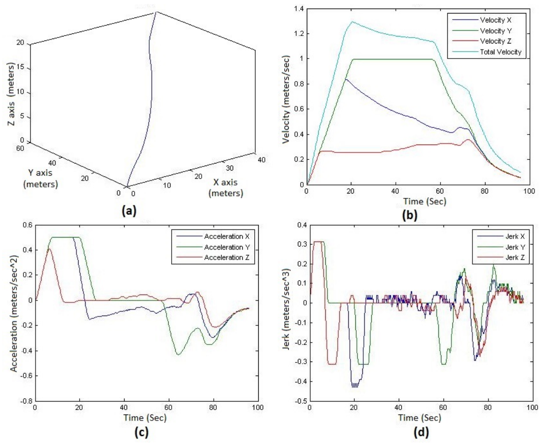

Fig. 4: The simulation result of minimum time trajectory with velocity and acceleration profile is obtained by solving (24). It can be seen that the velocity and acceleration level constrained ($v_{max} = 1.30m/s$ and $a_{max,l} = 0.5m/s^2, l \in \{x,y,z\}$) are perfectly satisfied whereas jerk profile remains bounded ($j_{max,l} = 0.5m/s^3, l \in \{x,y,z\}$). Similar to the usual time optimal planning with acceleration input, at least one of the acceleration component is near saturation, when the velocity profile is not the limit curve. The acceleration profile is zero when the velocity profile is at the limit curve.

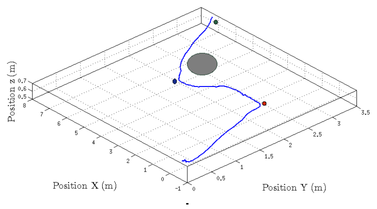

Fig. 5: 2D obstacle avoidance experiment. Quadrocopter flies through series of 2D way-points shown in dots to avoid the obstacle in X-Y plane.

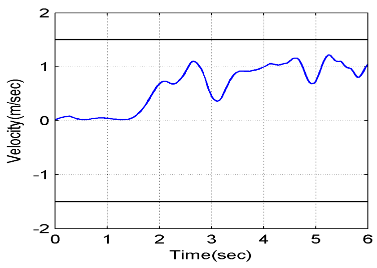

Fig. 6: Velocity profile, maximum achieved value of $1.28m/sec$ bounded to $v_{max} = 1.5m/sec$

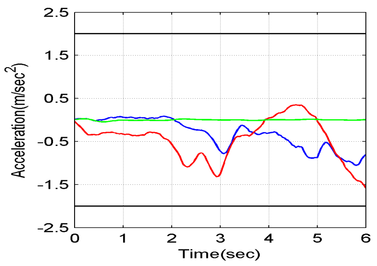

Fig. 7: The resulting continuous acceleration profile, where blue, red and green line corresponds to a_x, a_y and a_z respectively with $a_{max} = 2m/sec^2$

and corresponding state of quadcopter is calculated using simple vector geometry. *The state information received from ArTags and on-Board IMU are susceptible to erroneous data.* The IMU data and state estimated using ArToolkit are fused

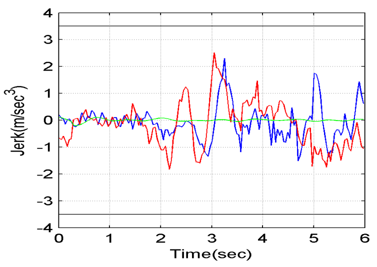

Fig. 8: The bounded jerk profile,where blue,red and green line corresponds to j_x, j_y and j_z respectively with $j_{max} = 3.8m/sec^3$

using kalman filter to minimize the errors in state estimation. The proposed framework utilizes above estimated state as input and generates control commands which are transferred to quadcopter over Wi-Fi.

B. Experiments

The minimum time trajectory generated for quadrocopter corresponds to maximum limit on allowable velocity, acceleration and jerk. Although data sheet and user manual available for *Parrot*^{TM} Ardrone claims maximum achievable velocity up to $11.11m/s$ but due to space and safety constraints, the maximum velocity was bounded at $v_{max} = 2.8 - 1.5m/s$. Similarly, maximum acceleration and jerk limit was kept around $a_{max} = 2 - 3.8m/s^2$ and $j_{max} = 6 - 3.8m/s^3$ respectively for current experimental setup. However, theoretical maximum allowable jerk $j_{max} = 25m/s^3$ with $f_{min} = 9.8m/s^2$ (i.e. g) and $\omega_{max} = 4rad/sec$, centrum of recommended operational range as per the ArDrone data sheet.

1) Collision avoidance with planar (X-Y plane) way points navigation: In current experiment an obstacle was placed at a known location between initial and final state of the quadrocopter flight. The experiment start from quadrocopter at *hover state* and target state is updated according to sequence of way-points provided by the user application, which are pre-computed for planar motion to avoid obstacle shown in "Fig. 5". At each control update, a new trajectory to the next immediate way-point is computed. Hence execution of each trajectory segment is accompanied with several re-plannings. Velocity, acceleration and jerk profile of quadrocopter for the experiment is shown in "Fig. 6", "Fig. 7" and "Fig. 8". Maximum velocity reached during the experiment was around 1.28 m/s, which is less than v_{max} and acceleration profile are continuous with bounded jerk. Due to limited indoor space operations the quadrocopter may never reach its v_{max}, a_{max} or j_{max} constraints. All state estimation and proposed control framework computations performed online during the flight where average computation time was 21 milliseconds.

2) Collision avoidance with Non-Planar (X-Y-Z) way point navigation: Similarly, the quadrocopter starts it motion from *hover state* and target state is updated with new way-points provided by user application, which are pre-computed for Non-planar motion to avoid obstacle shown in "Fig. 9". The velocity, acceleration and jerk profile of quadcopter for the experiment is shown in "Fig. 10", "Fig. 11" and

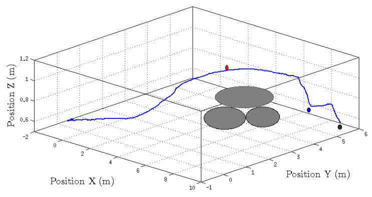

Fig. 9: 3D obstacle avoidance experiment. Quadrotor flies through series of 3D way-points shown in dots to avoid the obstacle in X-Y-Z plane.

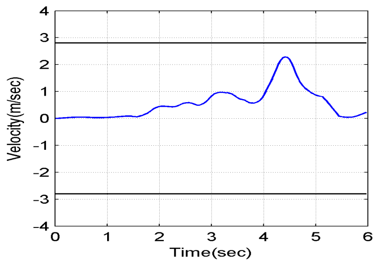

Fig. 10: Velocity profile, maximum achieved value of $2.32 m/sec$ bounded to $v_{max} = 2.8 m/sec$

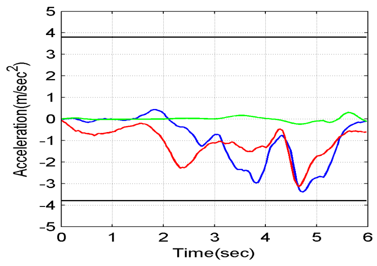

Fig. 11: The resulting continuous acceleration profile, where blue, red and green line corresponds to a_x, a_y and a_z respectively with $a_{max} = 3.8 m/sec^2$

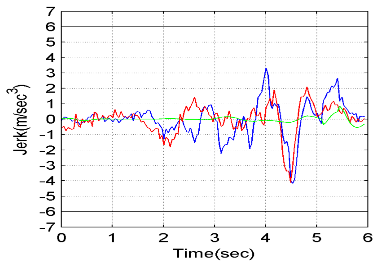

Fig. 12: The bounded jerk profile, where blue,red and green line corresponds to j_x, j_y and j_z respectively with $j_{max} = 6 m/sec^3$

"Fig. 12". The average computation time is observed as 20 milliseconds.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, minimum time trajectory generation and control framework has been presented that adheres to the dynamic and input constraints of quadrocopter vehicle. The framework is fast enough to be evaluated at every update, and therefore allows feedback control based on planned trajectories. This planning and feedback control method seems promising, as validated by experiments on *Parrot*^{TM} Ardrone 2.0, within indoor lab environments.

The algorithm is generally independent of the vehicle's type. However, information about the vehicle influences the selection of the appropriate inequality constraints which leads to more reliable results. Finally, this paper demonstrates the effectiveness of this approach, when applied to commercial quadrocopter.

The current framework works with assumption of fixed heading angle ($\psi = 0$). Moving forward, it must incorporate the variable heading angle-yaw ($\psi \neq 0$) handling along the tangent direction of path, so that for outdoor navigation, quadrocopter could use front camera to monitor video to assist autopilot.

REFERENCES

[1] Amazon.com. (2013) Amazon prime air. [Online]. Available: http://www.amazon.com/b?node=8037720011
[2] S. Johnson. (2014) The flying donkey challenge lifting africa [la fondation bundi]. [Online]. Available: http://www.flyingdonkey.org/
[3] P.-J. Bristeau, F. Callou, D. Vissière, N. Petit *et al.*, "The navigation and control technology inside the ar. drone micro uav," in *18th IFAC World Congress*, vol. 18, no. 1, 2011, pp. 1477–1484.
[4] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. citatione, "A prototype of an autonomous controller for a quadrotor uav," in *European Control Conference*, 2007, pp. 1–8.
[5] M. Cutler and J. P. How, "Actuator constrained trajectory generation and control for variable-pitch quadrotors," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2012.
[6] S. Formentin and M. Lovera, "Flatness-based control of a quadrotor helicopter via feedforward linearization." in *CDC-ECE*, 2011, pp. 6171–6176.
[7] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 1383–1389.
[8] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for quadrotor flight," in *International Conference on Robotics and Automation*, 2013.
[9] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3248–3253.
[10] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3. IEEE, 2002, pp. 1936–1941.
[11] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.
[12] H. Park, "Choosing nodes and knots in closed b-spline curve interpolation to point data," *Computer-Aided Design*, vol. 33, no. 13, pp. 967–974, 2001.
[13] H. Yang, W. Wang, and J. Sun, "Control point adjustment for b-spline curve approximation," *Computer-Aided Design*, vol. 36, no. 7, pp. 639–652, 2004.
[14] W. Rackl, R. Lampariello, and G. Hirzinger, "Robot excitation trajectories for dynamic parameter estimation using optimized b-splines," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2042–2047.
[15] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1917–1922.
[16] W. Hoffmann and T. Sauer, "A spline optimization problem from robotics," *Rediconti di Mathematica*, vol. 26, pp. 221–230, 2006.
[17] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 1, pp. 102–106, 1984.
[18] H. Kato, "Artoolkit: library for vision-based augmented reality," *IE-ICE, PRMU*, pp. 79–86, 2002.
[19] L. Biagiotti and C. Melchiorri, "Trajectory planning for automatic machines and robots," pp. 188–194, 2008.