Dynamically Feasible Formation Control for Shape Transitions for Teams of Aerial Robots

Arjav Ashesh Desai | Ellen Cappo | Nathan Michael

Goal

We want to use teams of quadrotors as characters in a theatrical piece to convey a story. The robot teams need to fly in formations and transition between different formations. Therefore our goal is to :

Plan dynamically feasible transitions between different formations.

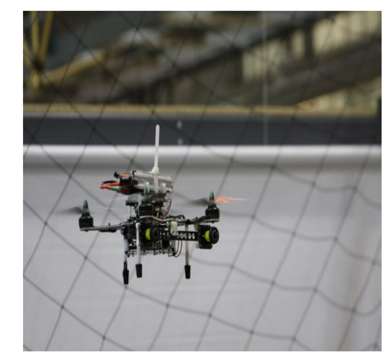


Fig 1: A quadrotor vehicle.

Fig 2 : A quadrotor team flying in the Vicon Motion Capture Arena.

Approach

Trajectory Generation

Goals:

- Generate optimal trajectories through the specified waypoints.
- Ensure continuity with respect to endpoint derivatives.
- Ensure feasibility with respect to quadrotor actuator limits.

Challenges:

- Limited time for computation during online use.
- Actuator constraints.
- Non trivial non-linear dynamics.

Approach:

1. We solve an unconstrained quadratic program to generate a piecewise polynomial trajectory through the specified waypoints. We seek to minimize an energy based cost function. This results in 9'th order minimum snap polynomial trajectories [1] (see Fig 3).
2. The generated trajectories are continuous across waypoints up to the 4'th derivative (see Fig 4).
3. We time scale the resulting trajectory to ensure feasibility with respect to the actuator limits (see Fig 5).
4. Our control loop runs at a frequency of 200 Hz. Determining trajectories which can pass through up to 4 waypoints (3 segments) can be done online in MATLAB (see Fig 6).

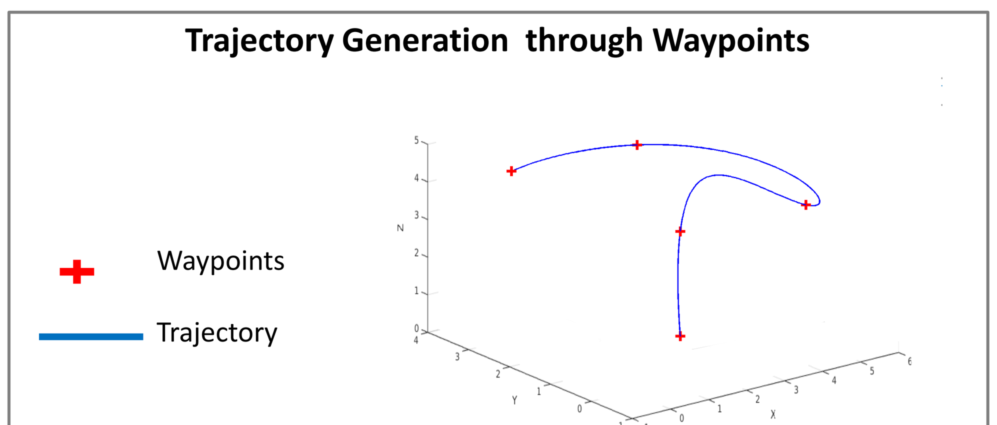


Fig 3: Polynomial trajectory passing through multiple waypoints.

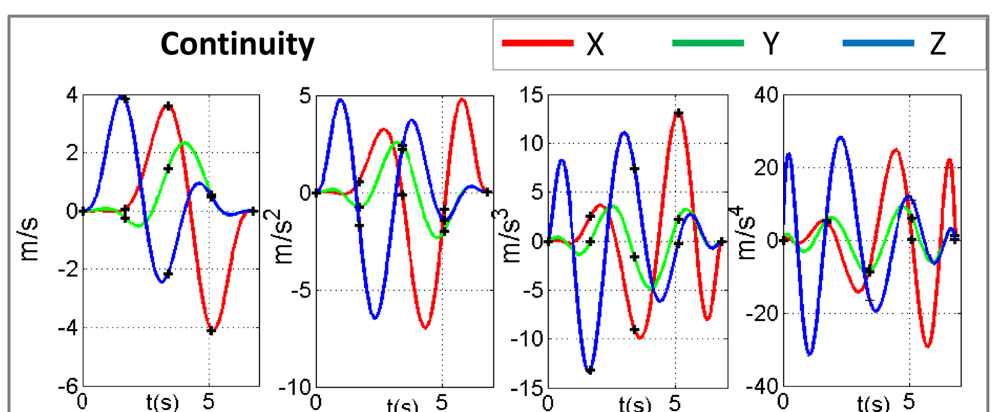


Fig 4: Continuity across higher order endpoint derivatives.

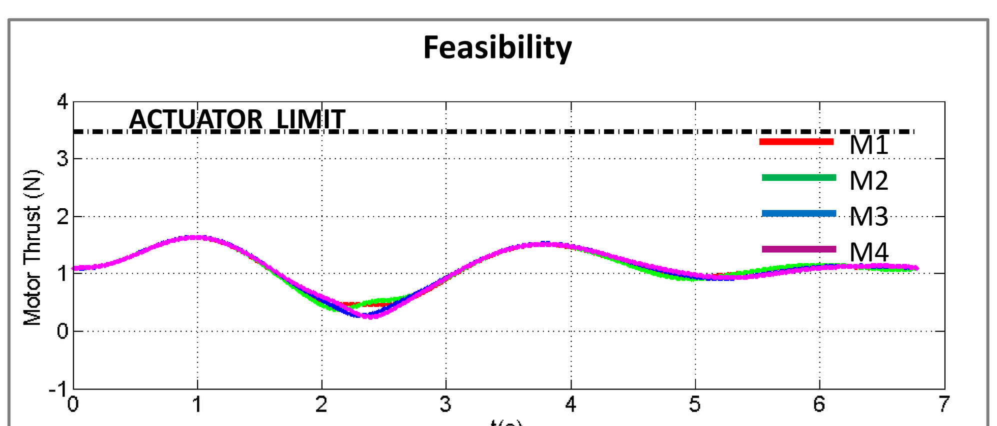


Fig 5: Trajectories generated are feasible

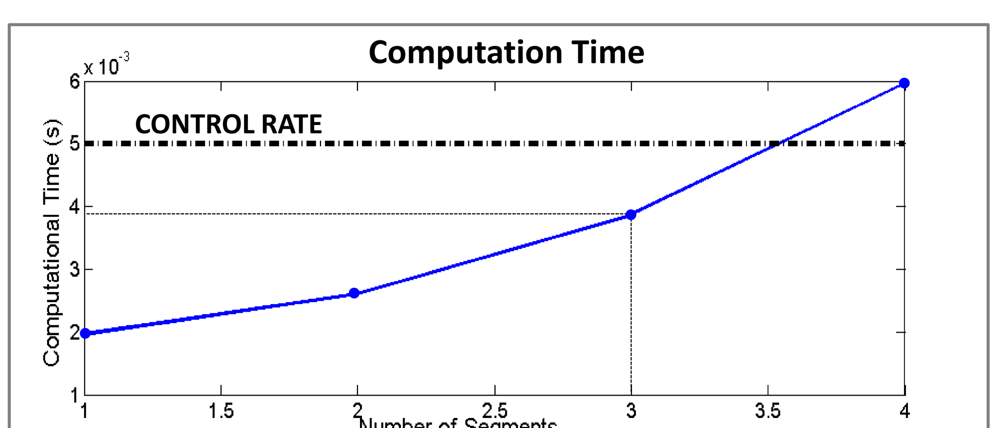


Fig 6 : 3 segment trajectory generation possible online

Formation Control

Goals:

- Define a multi-robot formation.
- Specify a group trajectory for a formation.
- Evaluate performance of shape tracking in aggressive flights.

Approach:

1. We define the N robot formation to consist of a leader and N-1 followers (see Fig 7).
2. We define shape error metric as in [2] for the i'th robot as shown in Equation 1. To estimate the robustness of our formation control, we take the sum of shape errors across all robots in the formation. Figure 8 shows the collective shape error for the formation over the length of the trajectory shown in Figure 7.
3. Given a leader trajectory, single segment follower trajectories are generated online such that they minimize the shape error for the i'th robot over a predefined time horizon (1 second). Figure 7 shows the leader and follower trajectories.
4. Shape error increases as the aggressiveness of the trajectory increases. [*Aggressiveness = Thrust/Max Thrust*] Figure 9 shows the baseline performance of our shape tracking for increasingly aggressive flights.

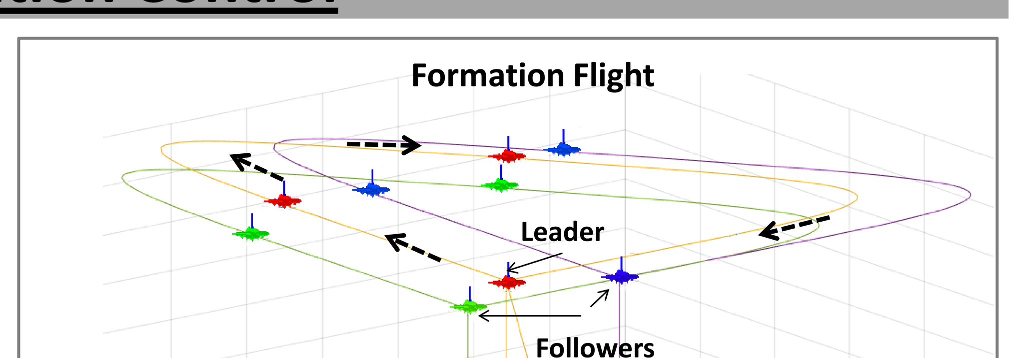


Fig 7: Snapshots of formation flight for three robots at different times.

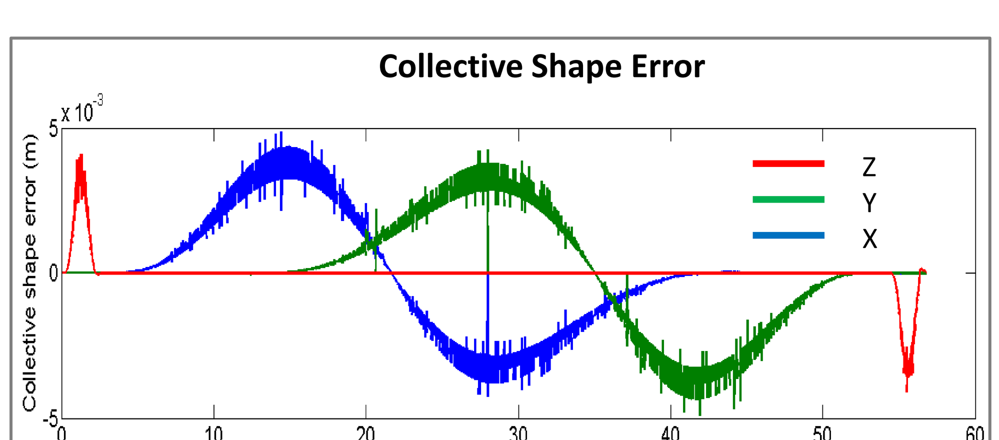


Fig 8: Collective shape error for a simulated formation flight.

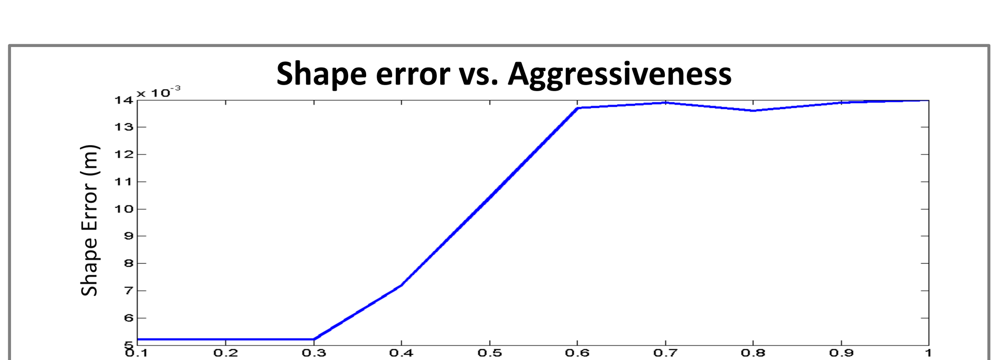


Fig 9: Maximum collective shape error vs. aggressiveness.

Equation 1

$$\mathbf{e}_i(t) = \sum_{j \in \mathcal{N}_i} c_{i,j} (\mathbf{x}_i(t) - \mathbf{x}_j(t) - \mathbf{s}_{i,j}(t))$$

$x_i(t)$: state of i'th robot
$x_j(t)$: state of the j'th robot
$s_{i,j}(t)$: shape vector for the robots i and j
$c_{i,j}$: i'th robot's confidence in the relative estimate of the state of the j'th robot

Shape Transitions

Goals:

- Generate a time varying shape vector to transition between two formations.
- Minimize overall transition time.
- Ensure feasibility with respect to actuator limits.
- Optimally assign robot positions within the new formation.

Approach:

1. We generate trajectories in the local frame of the leader from the initial shape, $S_{initial}$, to the final shape , S_{final} (see Fig 10).
2. The minimum feasible transition time is found using a line search method (Algorithm 1). Thrusts are calculated from the trajectory by using the simplified model presented in [3].
3. We use the Hungarian algorithm O(N^3) to ensure optimal assignment during transitions.

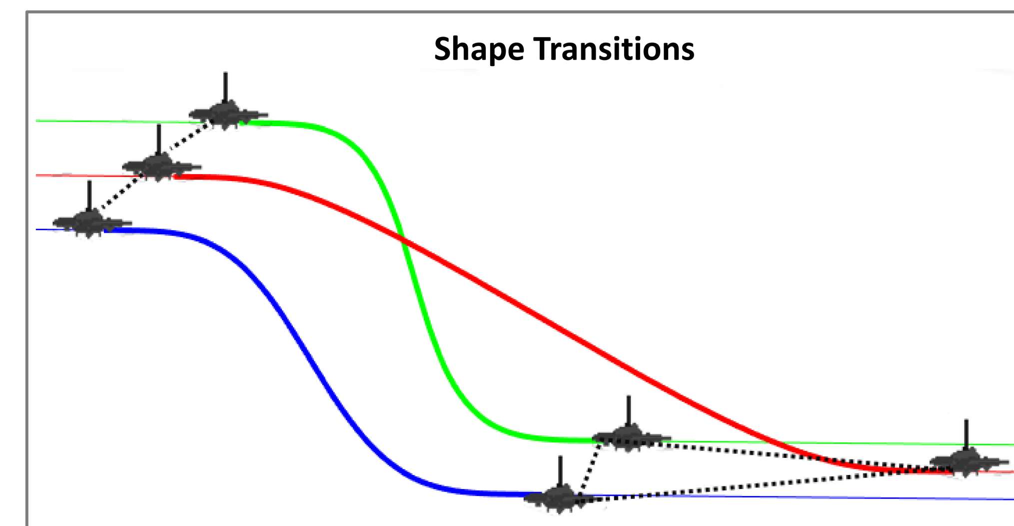


Fig 10: Shape transition from a line to a triangle.

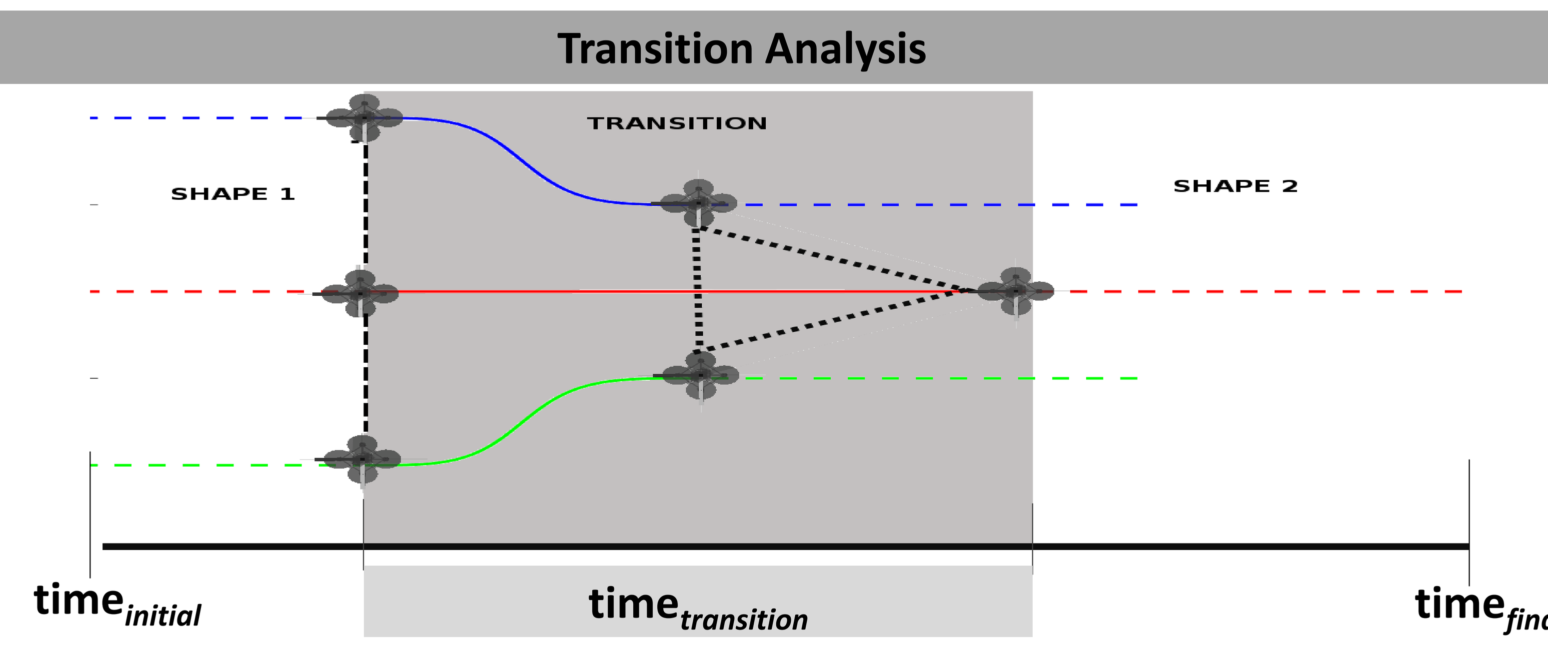
Algorithm 1: Line Search for calculating minimum feasible transition time

```
feasibility_flag = true
initialize time_final = distance / average_speed

while feasibility flag
    t = time_final - time_step
    [x(t), y(t), z(t)] = trajectory_generation(t)
    thrusts_per_motor = thrust_calc(x(t), y(t), z(t))
    if max(thrust_per_motor) <= thrust_limit
        time_final = t
    else
        feasibility_flag = false
    end
End

return time_final
```

Transition Analysis



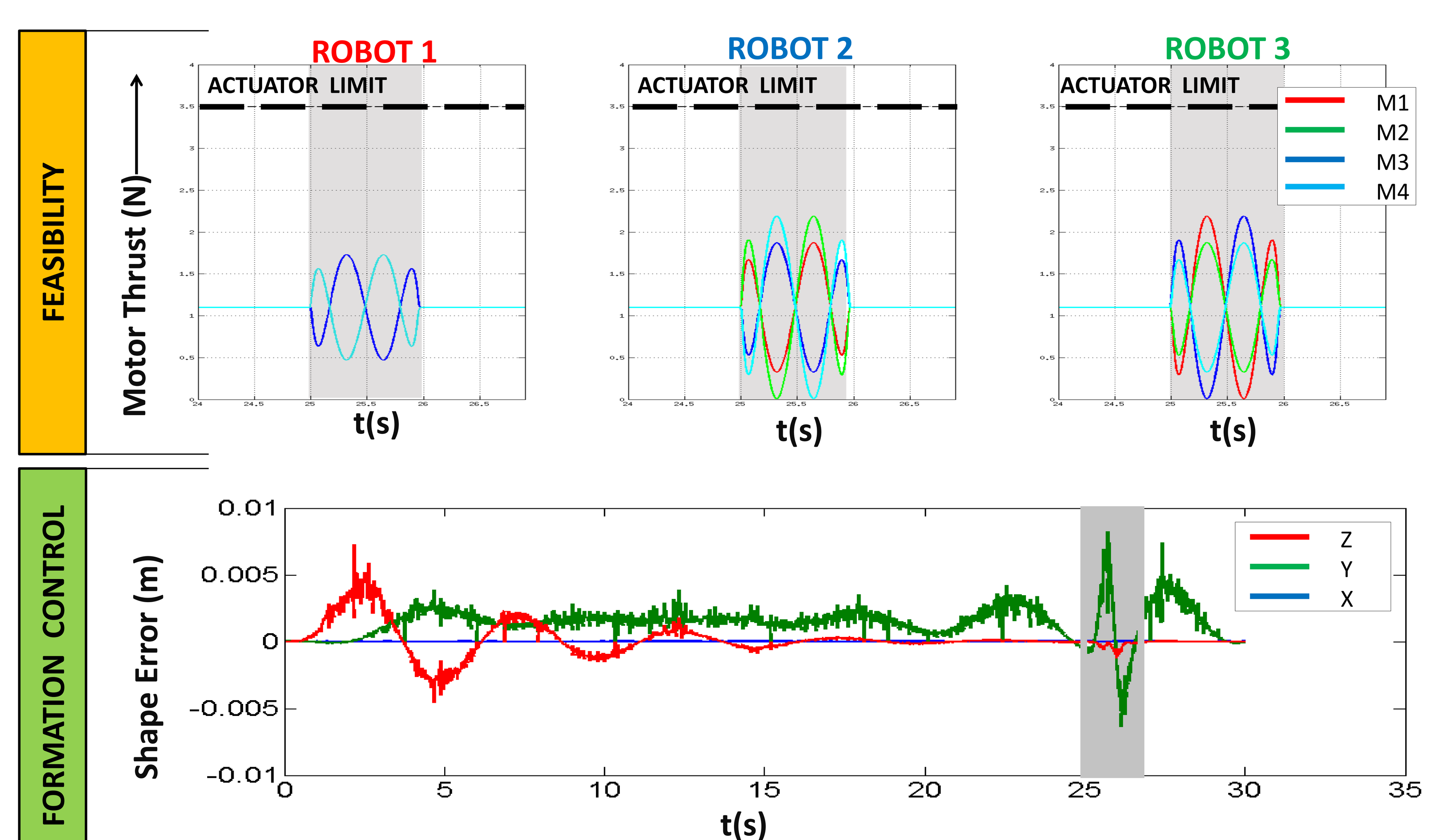


Fig 11: Feasible transition achieved in minimal time within the line search resolution.

Acknowledgments

Special thanks to Nate and Ellen for their support and guidance. Thanks to everyone in the Robust Adaptive Systems Lab and the RISS coordinators for making this research possible.

Conclusion and Future Work

Conclusion : Achieved a baseline performance for time optimal, feasible shape transitions in simulation.

Future Work : Extend implementation to hardware; add time scaling for collision avoidance.

References

1. Charles Richter, Adam Bry, and Nicholas Roy. *Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments*. ISSR 2013.
2. Matthew Turpin, Nathan Michael, and Vijay Kumar. *Decentralized Formation Control with Variable Shapes for Aerial Robots*. ICRA 2012.
3. Abbas Chamseddine, Youmin Zhang, Camille Alain Rabbath, Cedric Join, and Didier Theilliol. *Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle*. Aerospace and Electronic Systems, IEEE Transactions on, 2012.

Arjav Ashesh Desai | RISS Cohort - 2015 | Email : arjavd@andrew.cmu.edu