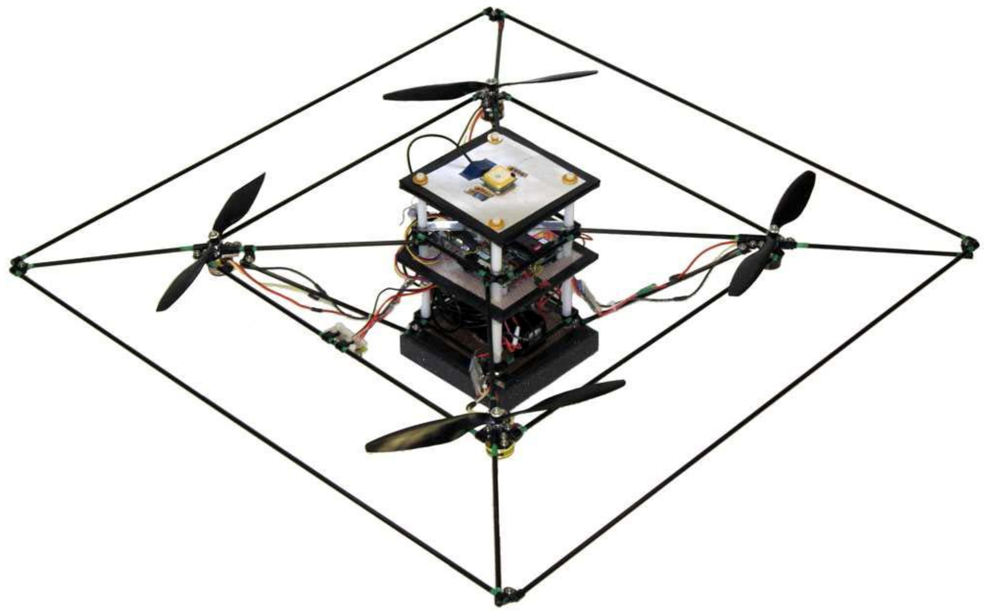


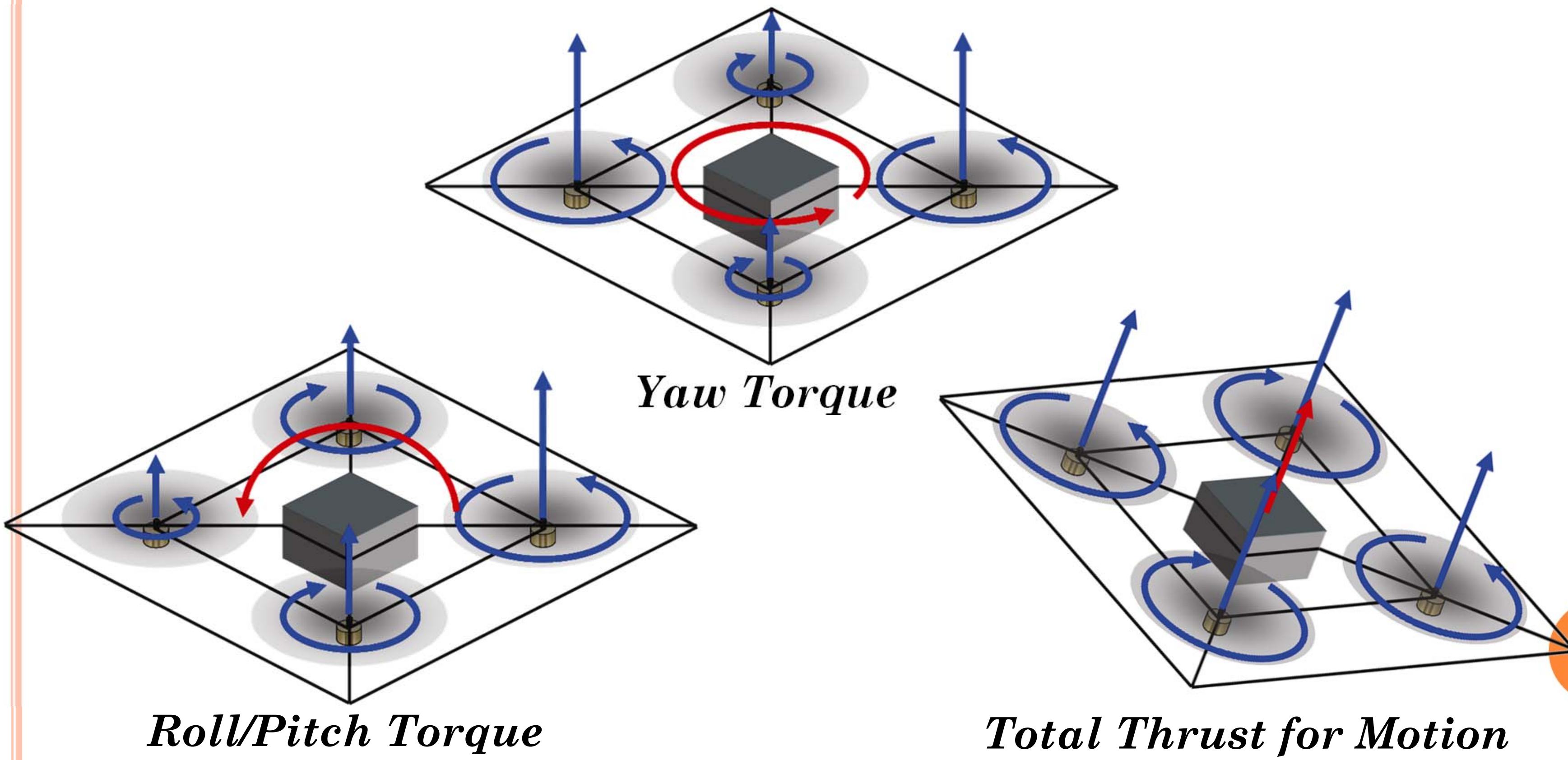
ME 597: Autonomous Mobile Robotics
Section 9 – Quadrotors

Prof. Steven Waslander

APPLICATIONS

How Do Quadrotors Work?

- Two pairs of counter-rotating blades allow for fixed pitch rotors and independent actuation of roll, pitch, yaw and altitude



Why Quadrotors

- Easy to build
- Easy to maintain
- Can hover in place
- Simple dynamics
- Relatively safe

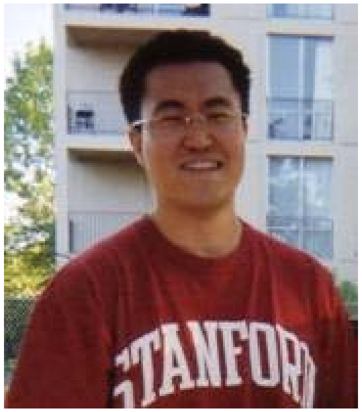
STARMAC PROJECT GOALS

- To develop a reliable, easy to use aerial multi-vehicle platform
- To investigate quadrotor dynamics and control
- To investigate path planning in challenging environments
- To investigate coordination of multiple vehicles

STARMAC Team 03-08

Dave Dostal

Prof. Claire Tomlin



Jung Soon Jang

Dev Rajnarayan

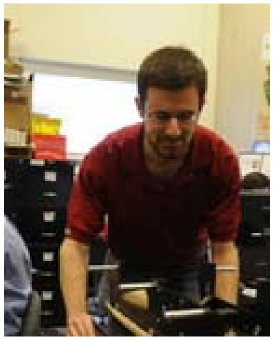
Gabe Hoffmann, Steven Waslander, Mike Vitus, Haomiao Huang, Vijay Pradeep

Jeremy Gillula

WAVELAB GOALS

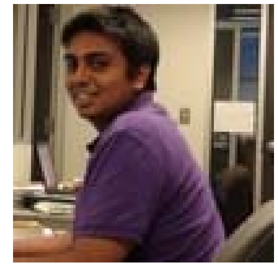
- Robust, precision control
  - GPS/IMU in winds
  - Vision to support/replace GPS
- Onboard mapping
  - Lidar based occupancy grid mapping
  - Vision (mono/stereo/multi-cam)
- Onboard planning
  - Real time 3D planning
    - Known environments – PRM/Motion primitives
    - Unknown environments – combine with mapping

WAVELAB QUADROTOR TEAM 09-14


Ryan Gariepy


Carlos Wang


Abdel El Bably


Prasenjit Mukherjee


Peiyi Chen

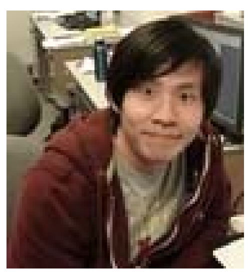

Nima Mohajerin


Arun Das


Adeel Akhtar


Me


Derek Chow

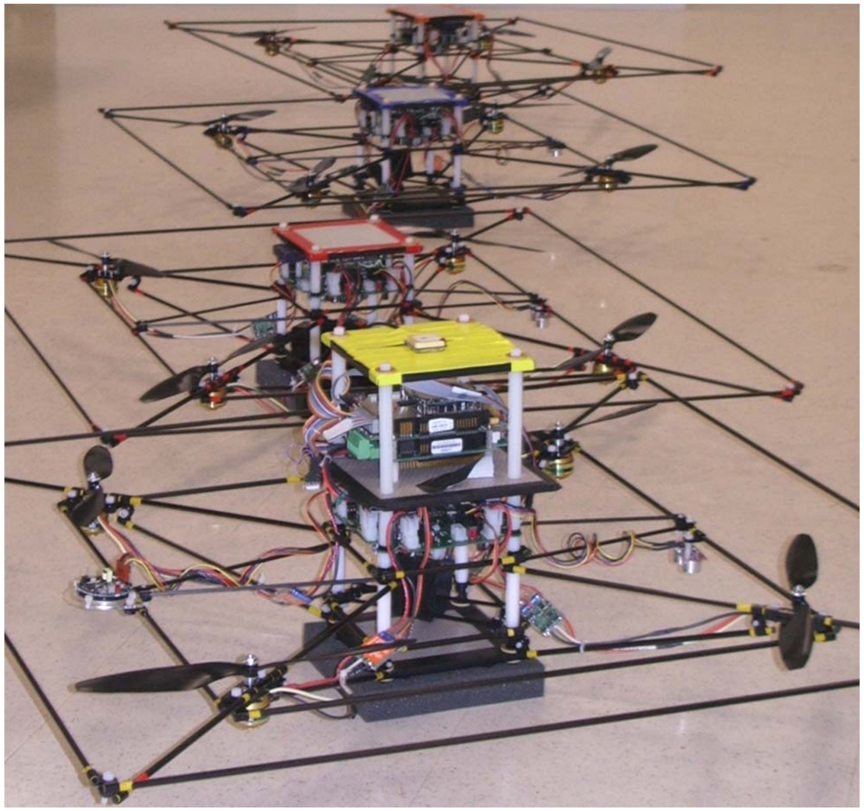

Yan Ma


Mike Tribou


Kevin Ling

Outline

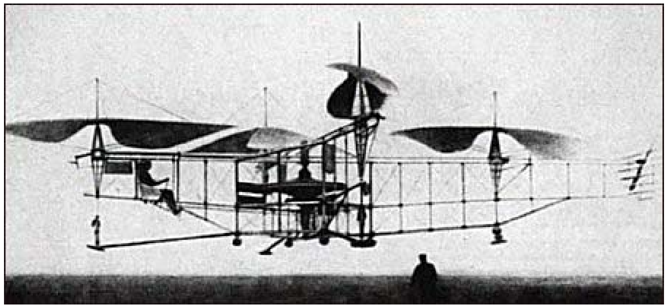
- Introduction
- Platform Development
- Vehicle modeling
- Estimation and Control
- Mapping
- Motion Planning
  - Tunnel-MILP
  - PRM/NLP
  - Reachable Sets
- Multi-Vehicle Coordination



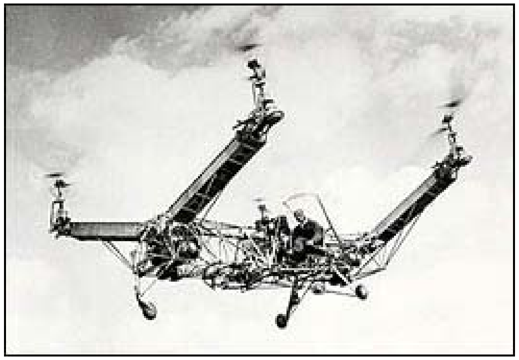
QUADROTOR HISTORY

- Historical attempts
  - Earliest attempts failed to leave ground effect, too complex for pilots
  - Later attempt worked well but interest waned with success of helicopter

De Bothezat (1922)



Convertawings Model "A" (1956)



Curtiss-Wright X-19A (1960)

Commercial Quadrotors

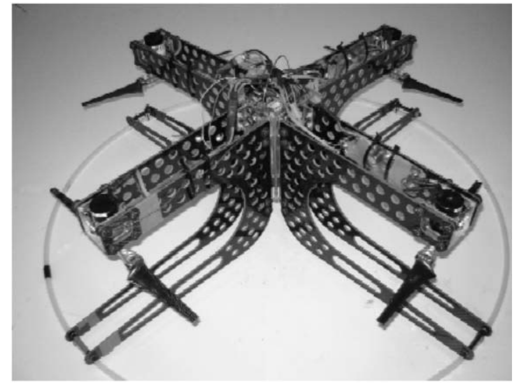
- RC-Toys Draganflyer (2002)
  - RC vehicle with stabilization

- Microdrones MD4-200 (2006)
  - Full GPS waypoint tracking
- Ascending Technology Hummingbird (2007)

- Aeryon Scout (2009)

RC Quadrotors

- Other research projects
  - X-4 Mark II, Robert Pounds, ANU, Australia
  - UAV SWARM, Jonathan How, MIT, USA
  - Nick Roy, MIT, USA
  - Javiator, Christoph Kirsch, Austria
  - Has now exploded to 100s of labs



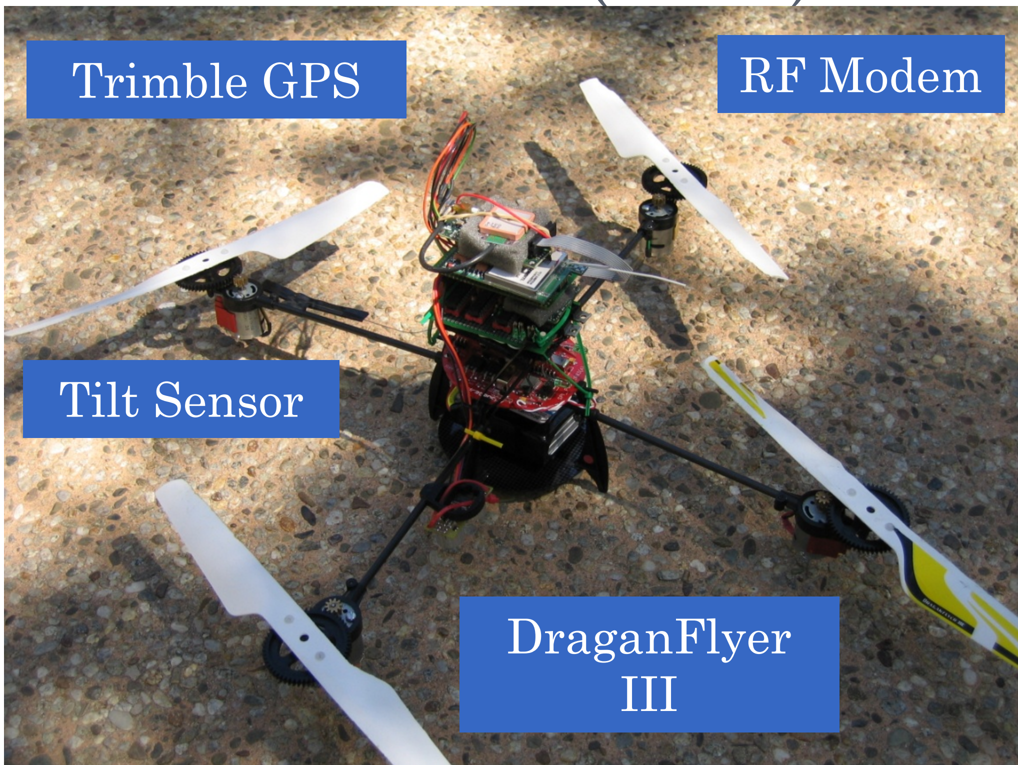
X-4 Mark II, 2005

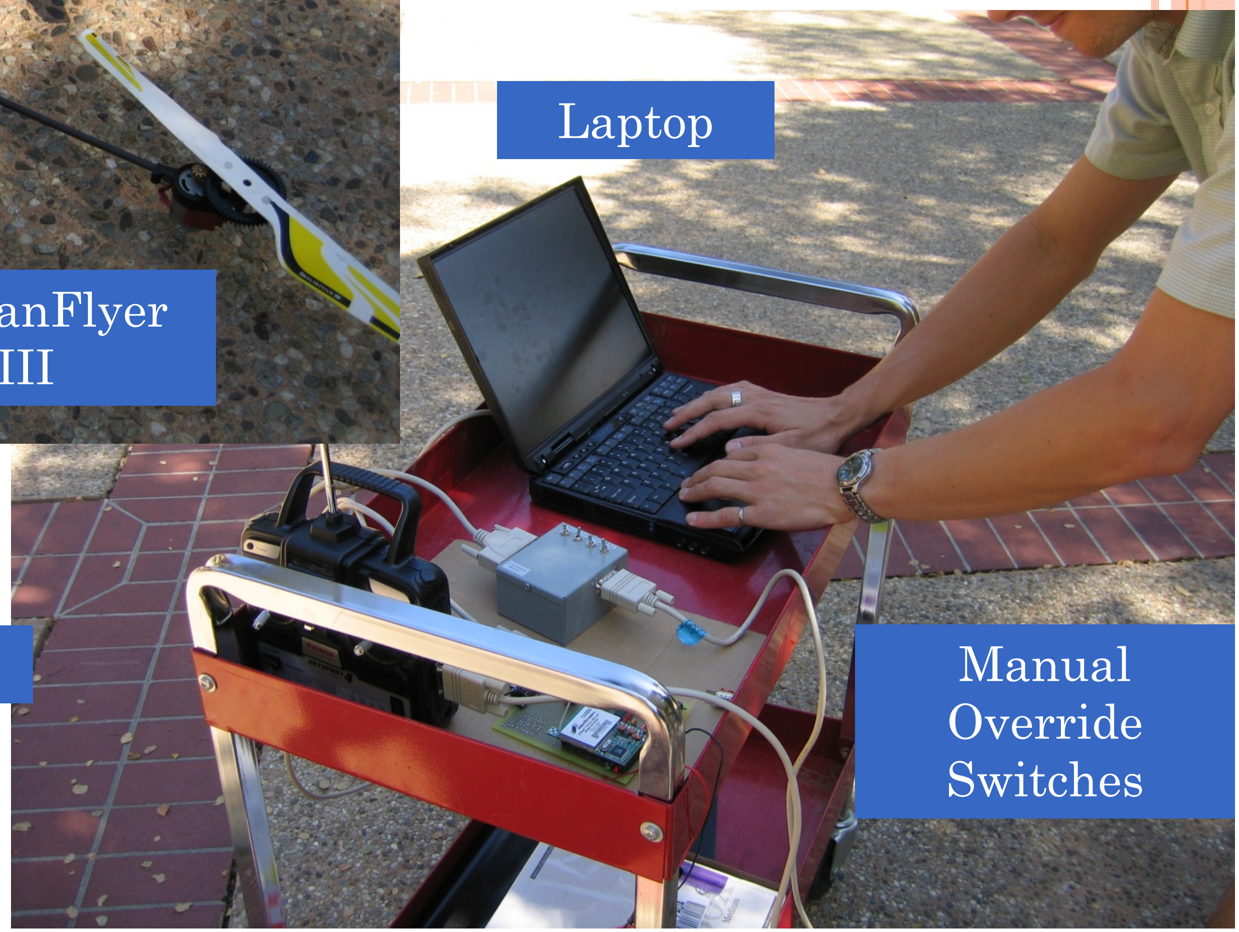
Nick Roy, 2009

MIT SWARM, 2007

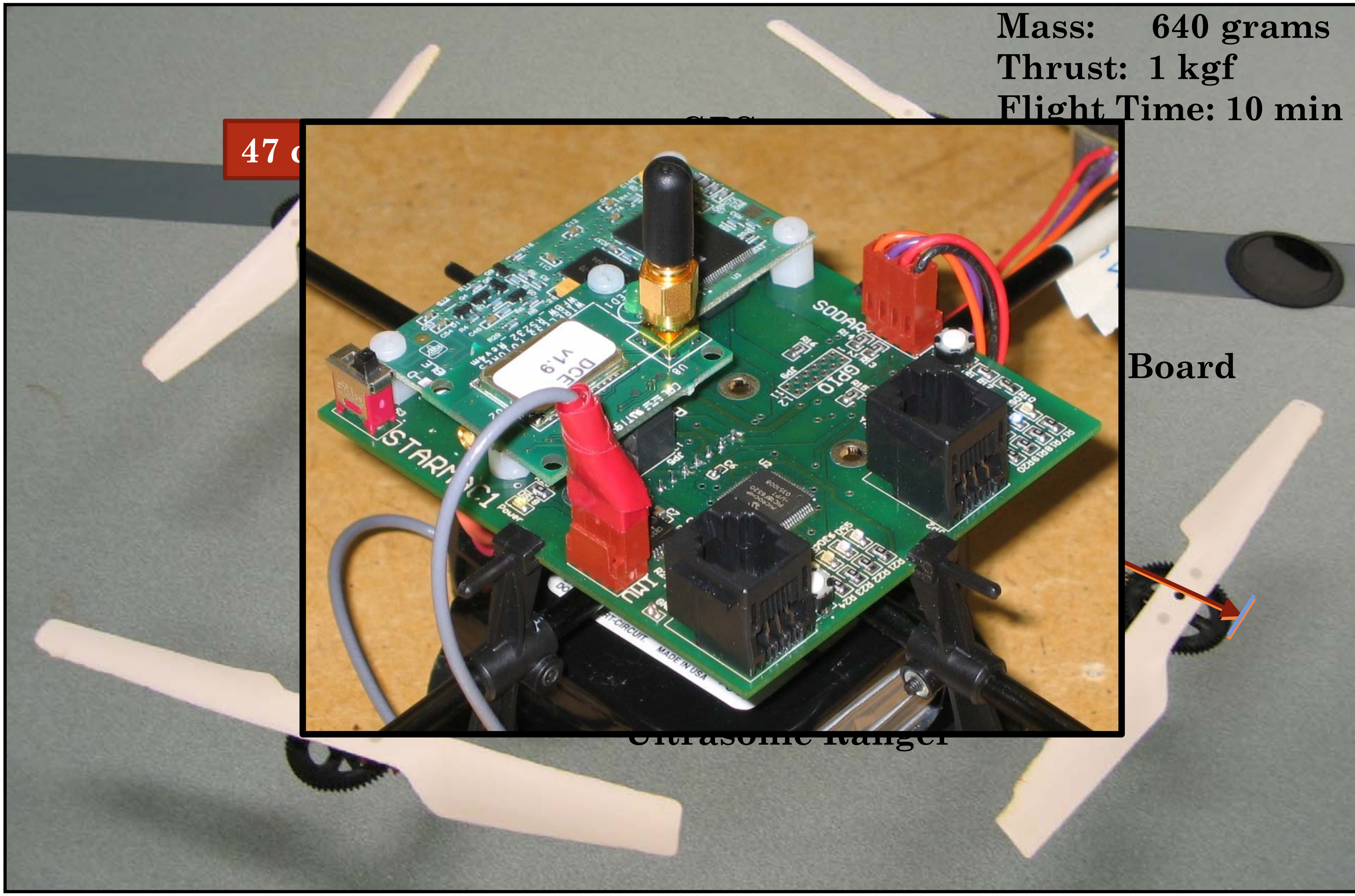
Javiator, 2009

First Design (2003)

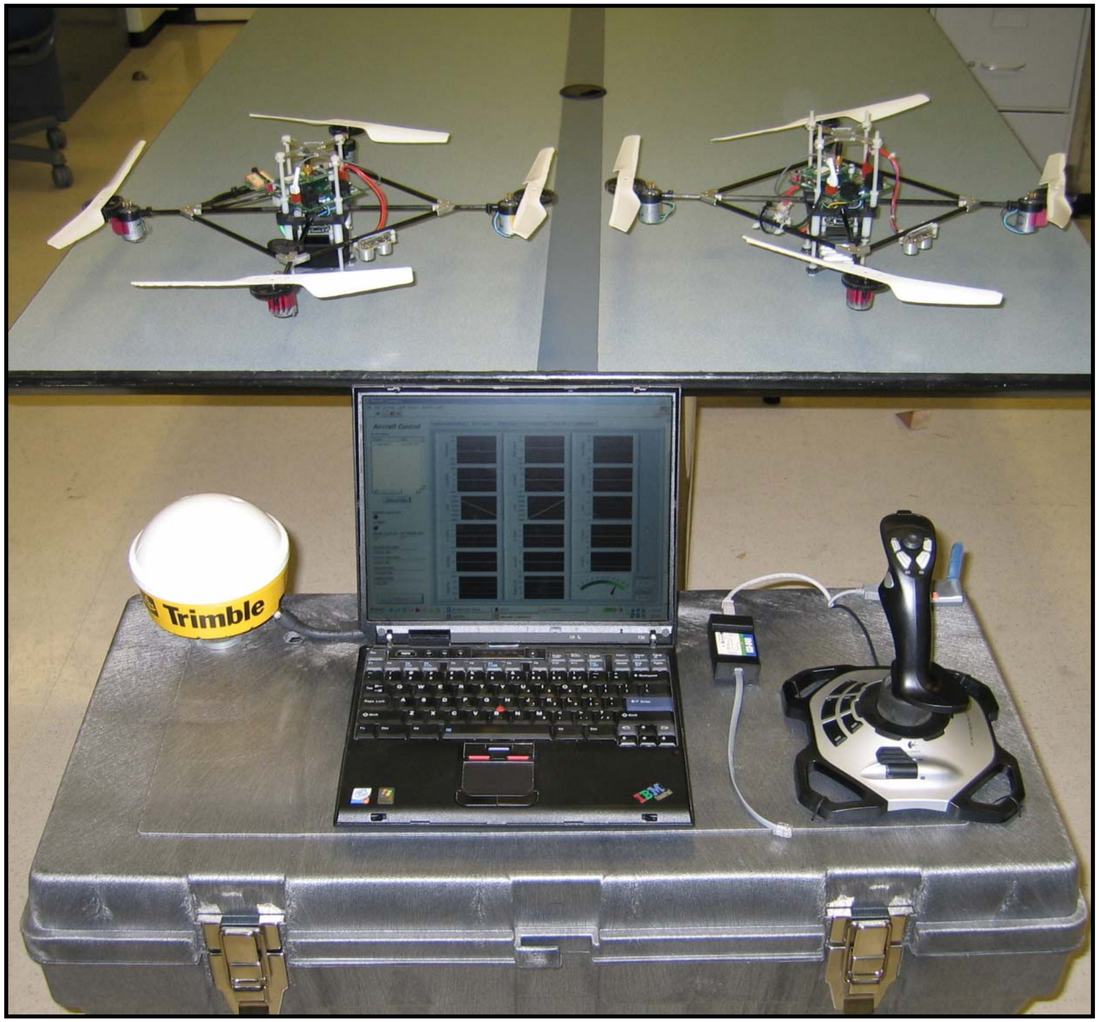




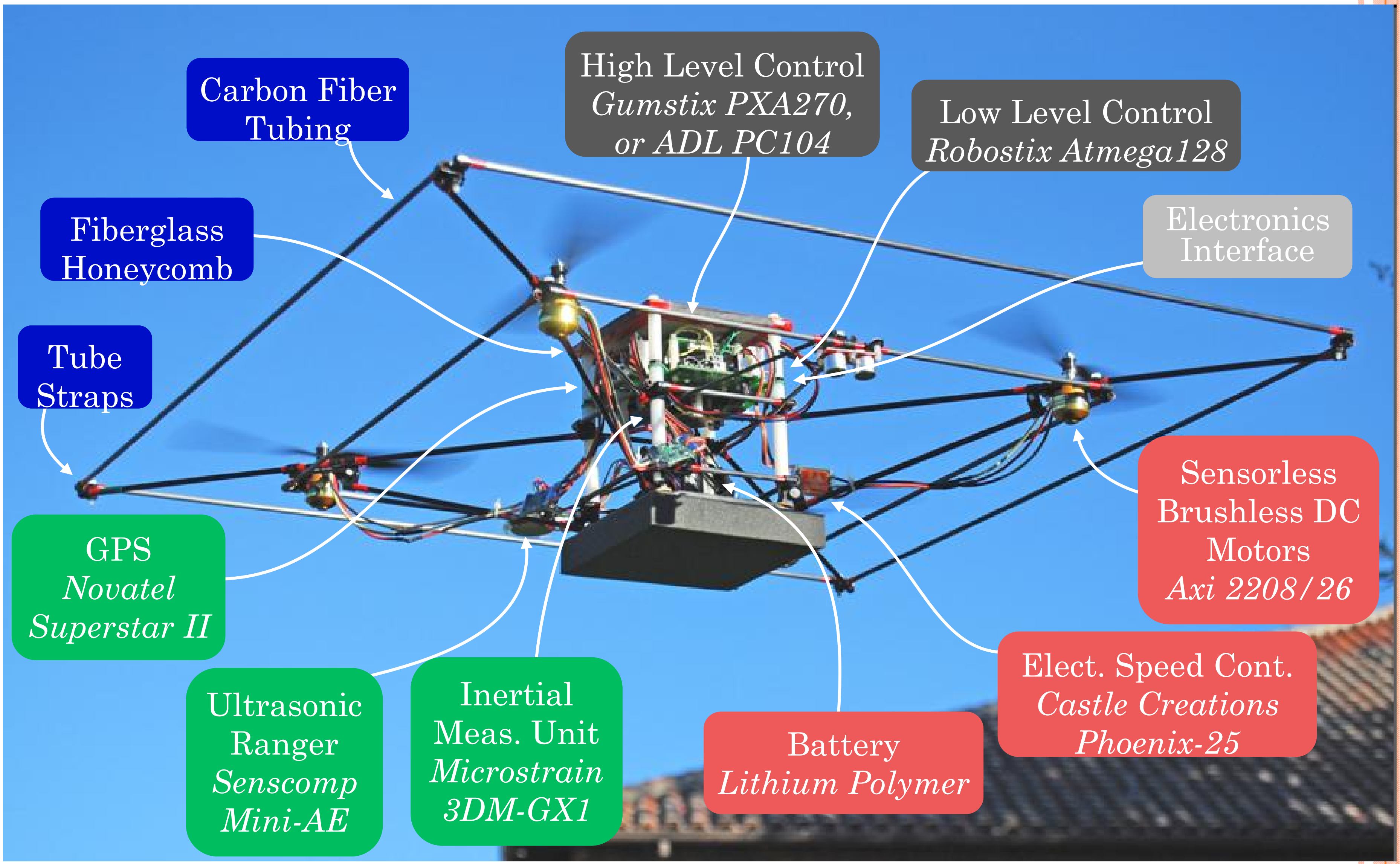
SECOND DESIGN (2004-2006) - VEHICLE



Second Design (2004-2006) - System

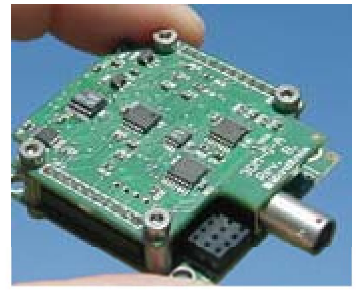


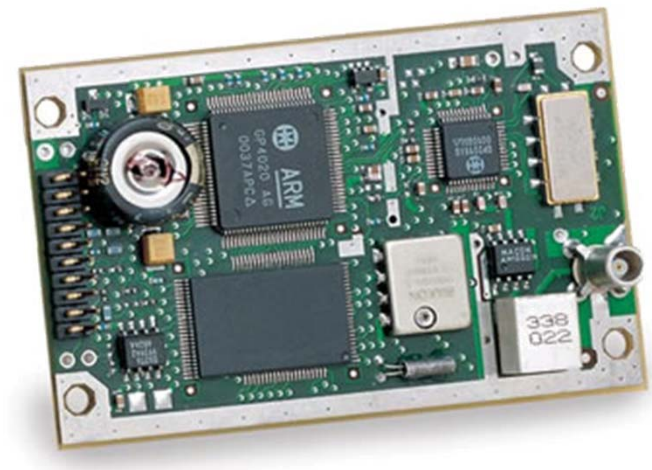
Third Design (2006 - 2009) - Vehicle



SENSORS

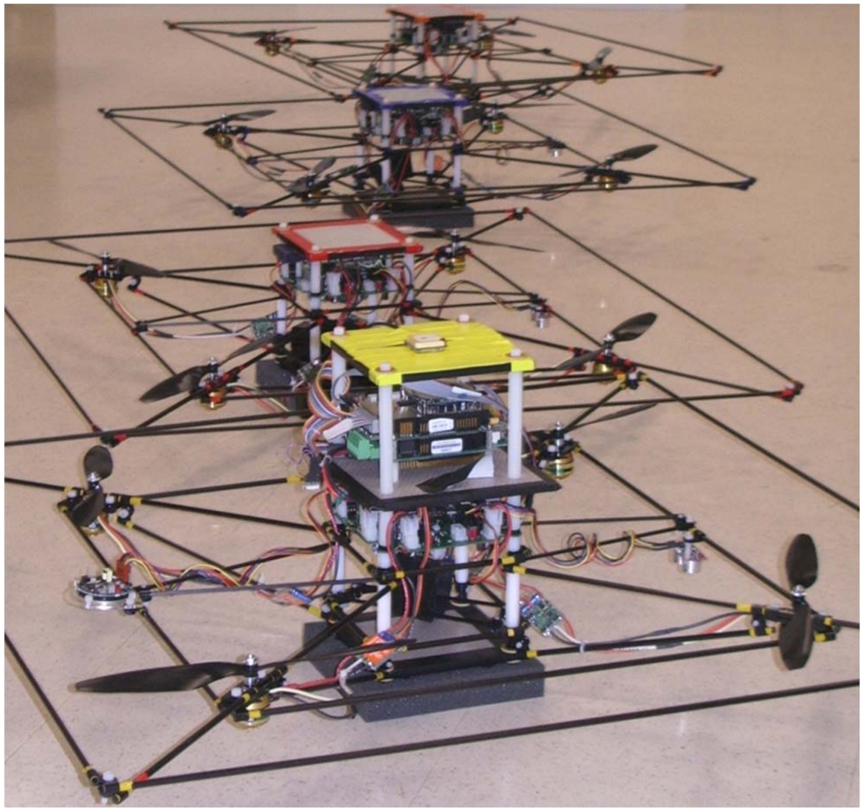
- IMU
  - Microstrain 3DM-GX1
  - Attitude, attitude rate, acceleration @ 76 Hz
- GPS
  - Novatel Superstar II
  - Carrier Phase Position & velocity @10 Hz
- SODAR
  - Senscomp Mini AE
  - Up to 50 Hz update
  - Up to 40 ft range



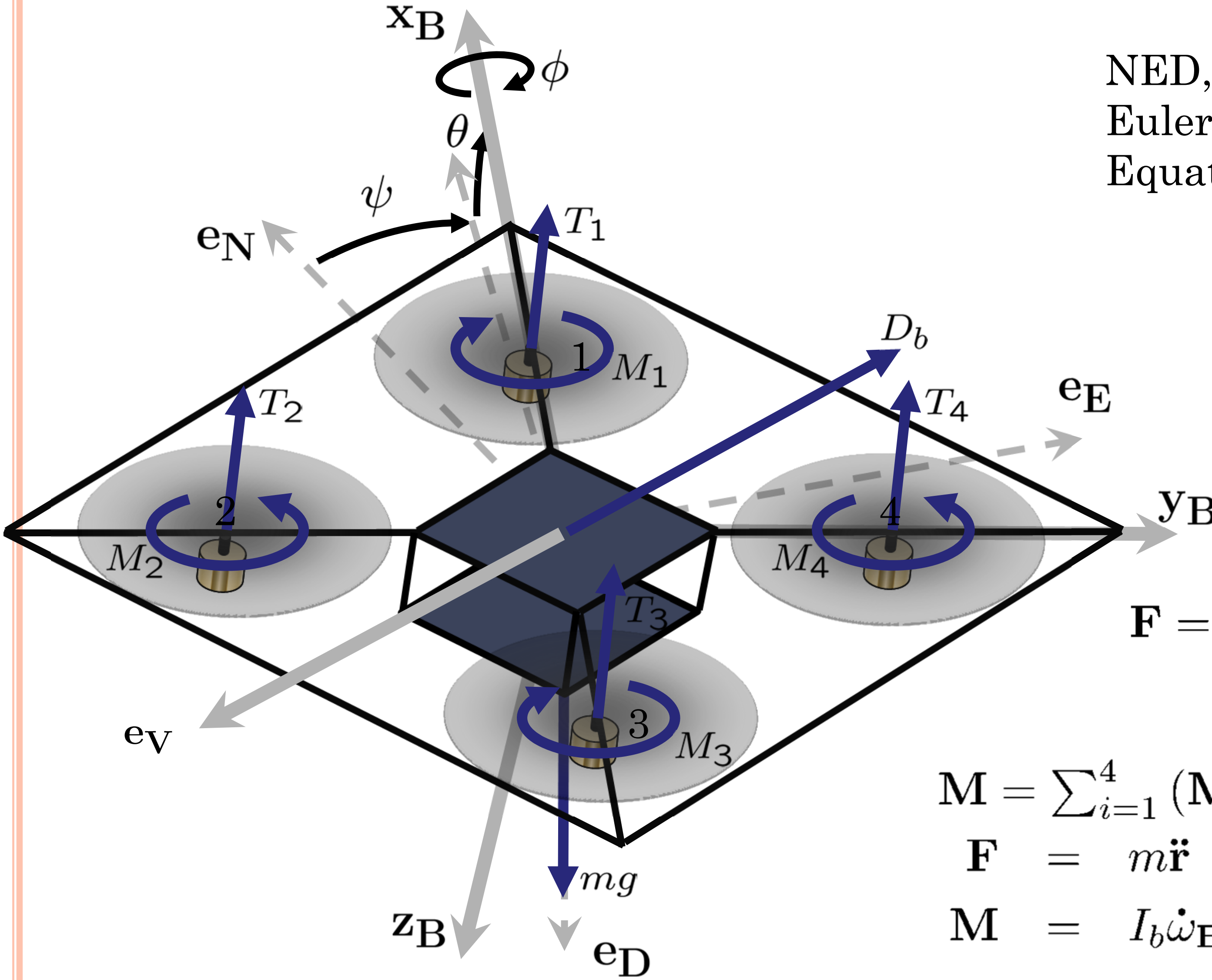


Outline

- Introduction
- Platform Development
- Vehicle modeling
  - Dynamics of vehicle
  - Aerodynamic Effects
  - Effect of Wind
- Estimation and Control
- Mapping
- Motion Planning
- Multi-Vehicle Coordination



Free Body Diagram



NED, Body coordinates
Euler Angles
Equations of motion:

$$\mathbf{F} = -D_b\mathbf{e_v} + mg\mathbf{e_D} + \sum_{i=1}^{4} -\mathbf{T_i}$$

$$\mathbf{M} = \sum_{i=1}^{4} (\mathbf{M}_i + \mathbf{r_i} \times (-\mathbf{T_i}))$$

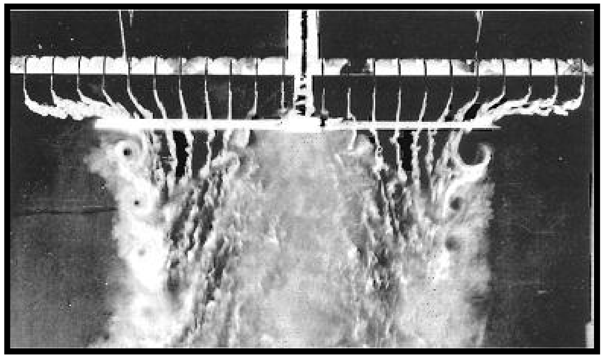
$$\mathbf{F} = m\ddot{\mathbf{r}}$$

$$\mathbf{M} = I_b\dot{\omega}_\mathbf{B} + \omega_\mathbf{B} \times I\omega_\mathbf{B}$$

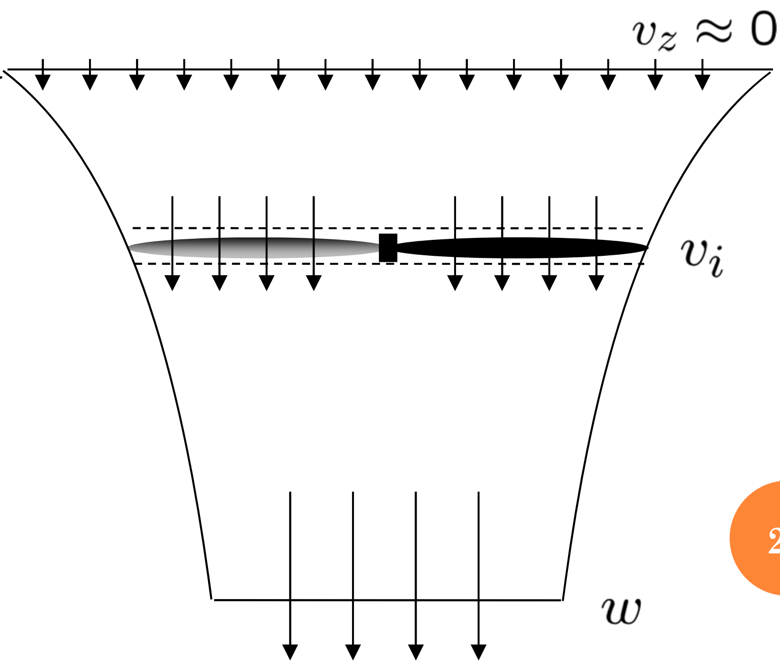
1. Total Thrust Variation

- From conservation of mass, momentum and energy:

$$\begin{aligned} \dot{m} &= \rho A v_i \\ T &= \dot{m} w \\ T v_i &= \frac{1}{2}\dot{m} w^2 \end{aligned}$$



Leishman, 2000



- Solving for induced velocity

$$w = 2v_i \qquad v_i = \sqrt{\frac{T}{2\rho A}}$$

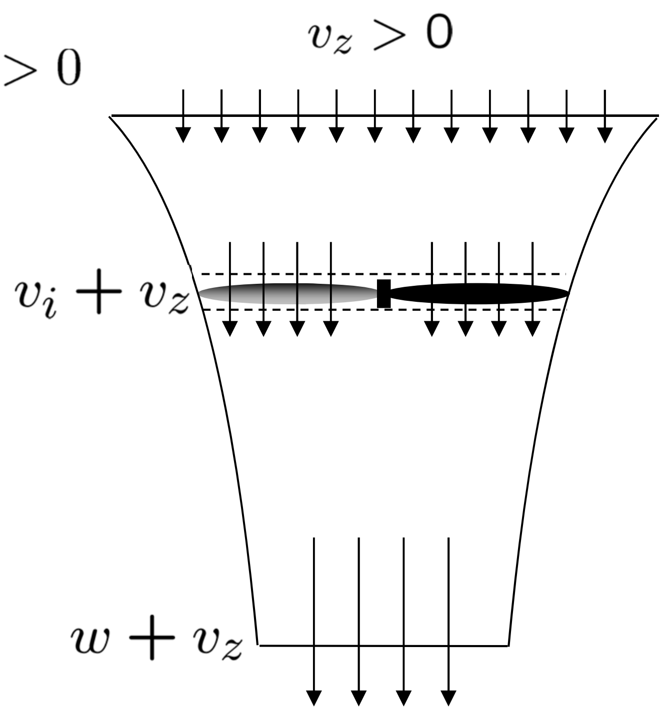
- Ideal power required at hover

$$P = T v_i = \frac{T^{3/2}}{\sqrt{2\rho A}}$$

Vertical Motion

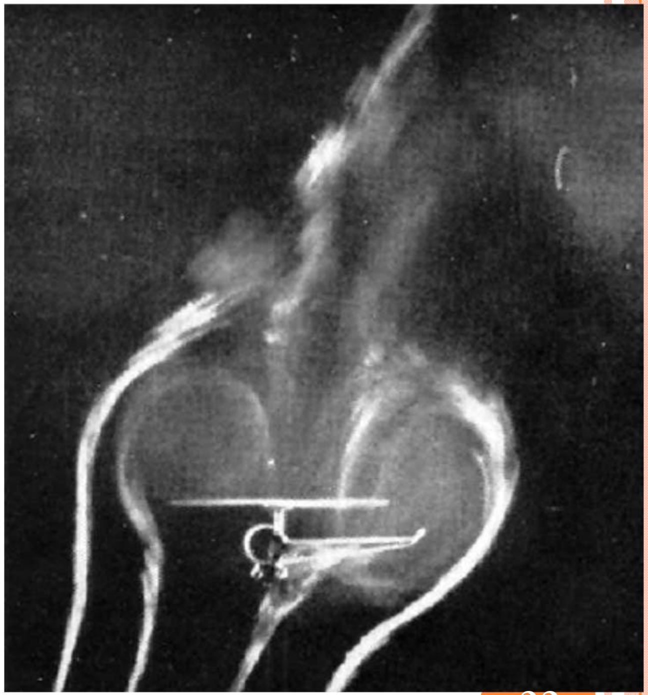
- Thrust can be modeled in three separate regions
  - Assumptions of momentum theory not always valid

  - Normal working state (ascent): $v_z > 0$
    - Momentum theory model valid
    - More power required for hover thrust



VERTICAL MOTION

- Thrust can be modeled in three separate regions
  - Assumptions of momentum theory not always valid
  - Normal working state (ascent): $v_z > 0$
    - Momentum theory model valid
    - More power required for hover thrust
  - Vortex ring state: $-2v_h < v_z < 0$
    - Transition state, unsteady flow
    - Empirical models for average thrust



Vortex Ring State
(Brown et al. 2002)

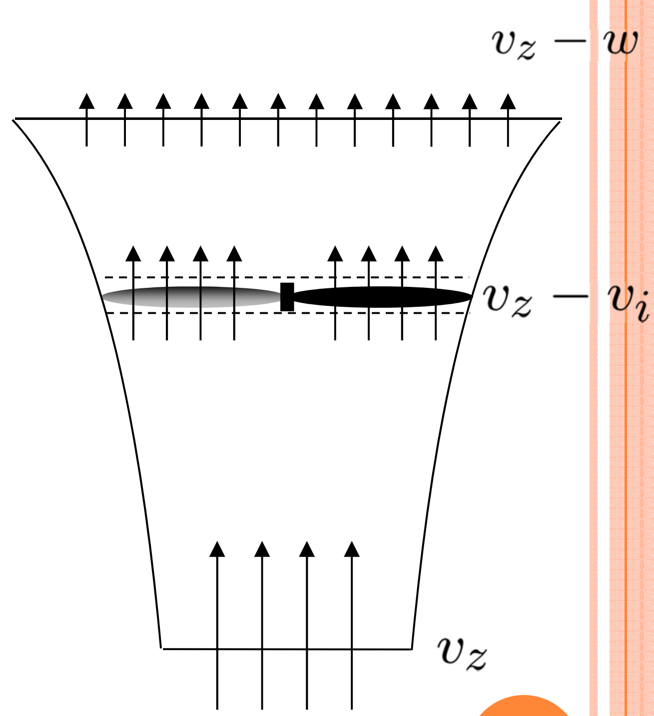
Vertical Motion

- Thrust can be modeled in three separate regions
  - Assumptions of momentum theory not always valid

  - Normal working state (ascent): $v_z > 0$
    - Momentum theory model valid
    - More power required for hover thrust

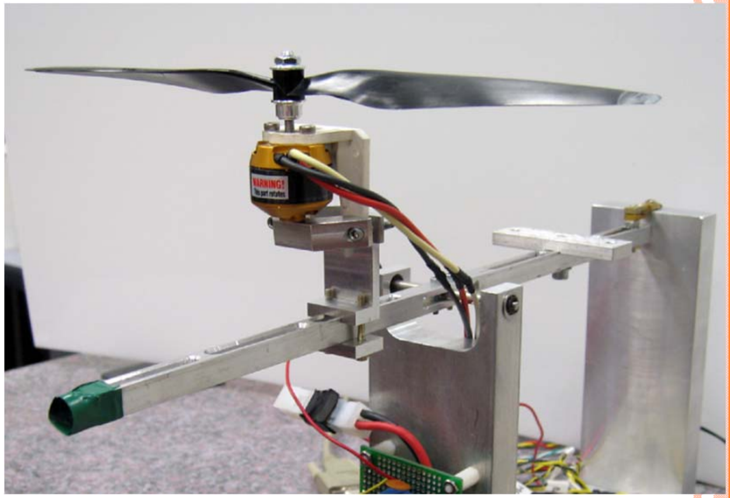
  - Vortex ring state: $-2v_h < v_z < 0$
    - Transition state, unsteady flow
    - Empirical models for average thrust

  - Windmill brake state: $v_z < -2v_h$
    - Momentum theory model valid
    - Less power required for hover thrust



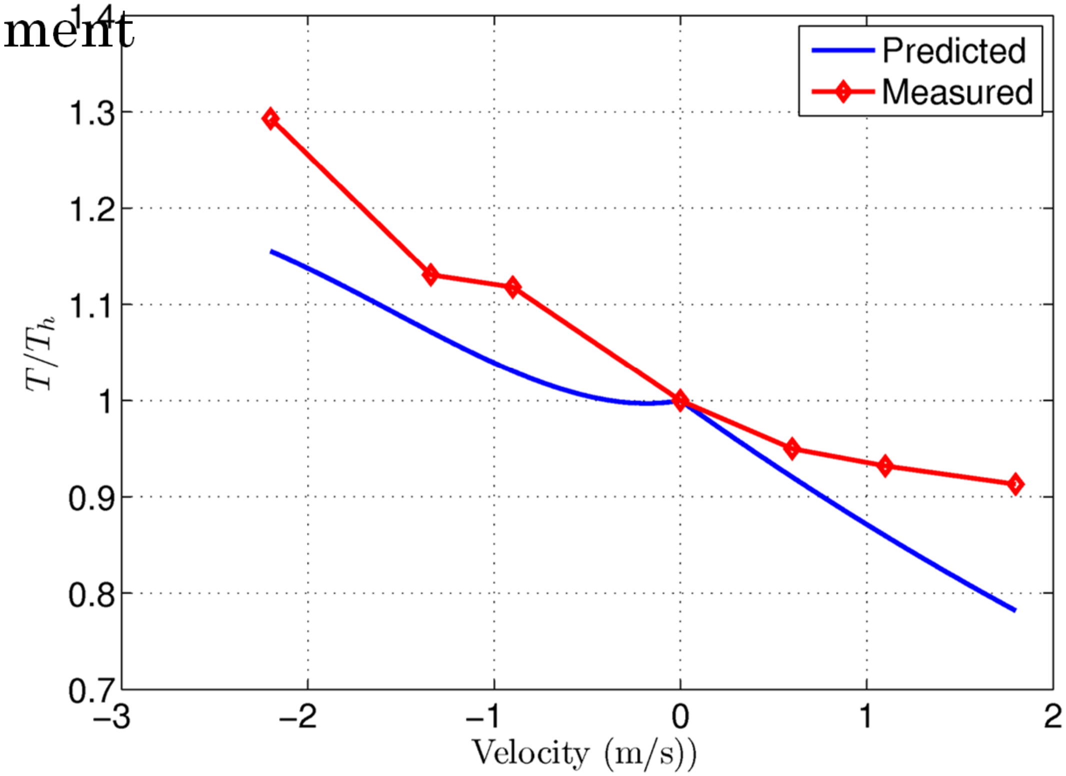
Test Stand Measurements

- Motor thrust measurements performed with load cell,
- Freestream velocity generated with cooling fan
- Wind speed measured with digital anemometer
- Convenient Labview interface for automated testing



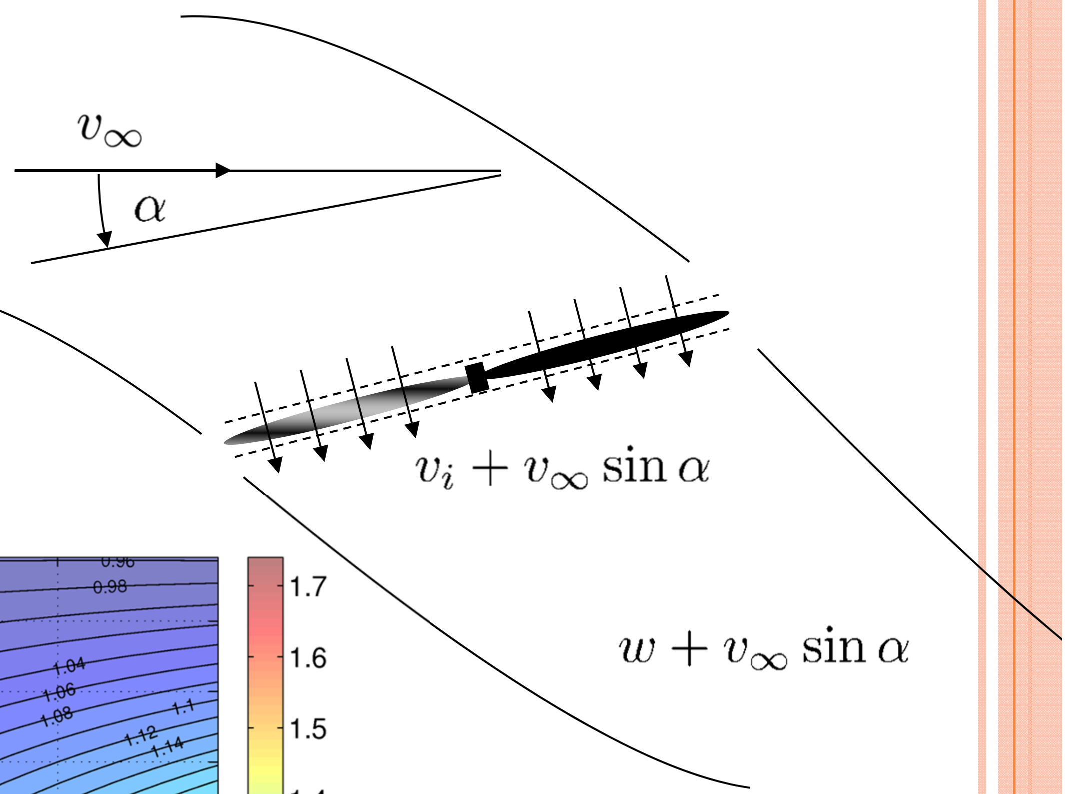
Experimental Results: Vertical Motion

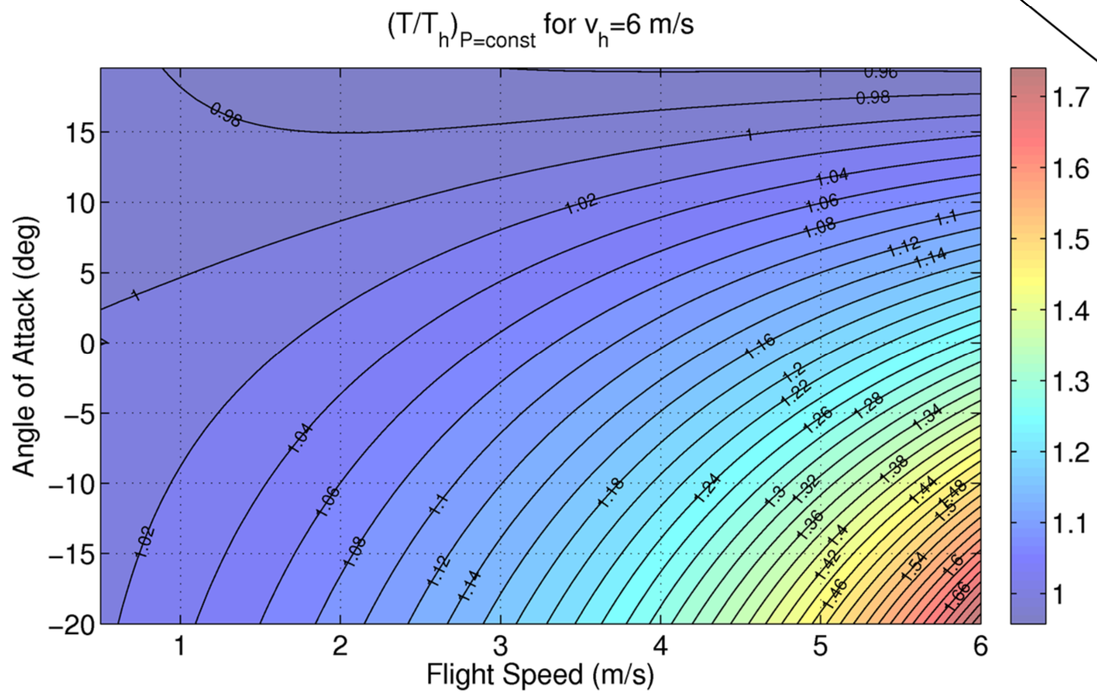
- Experiment consistent with predicted variation in thrust
- Noisy measurements in vortex ring state region
  - Refinement of apparatus and model may yield better agreement



TRANSLATIONAL MOTION

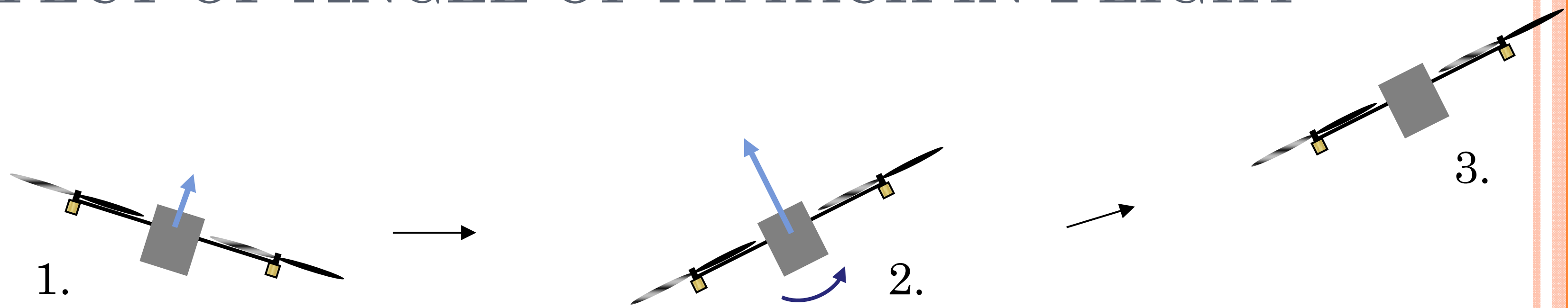
- Include non-zero freestream velocity and rotor angle of attack



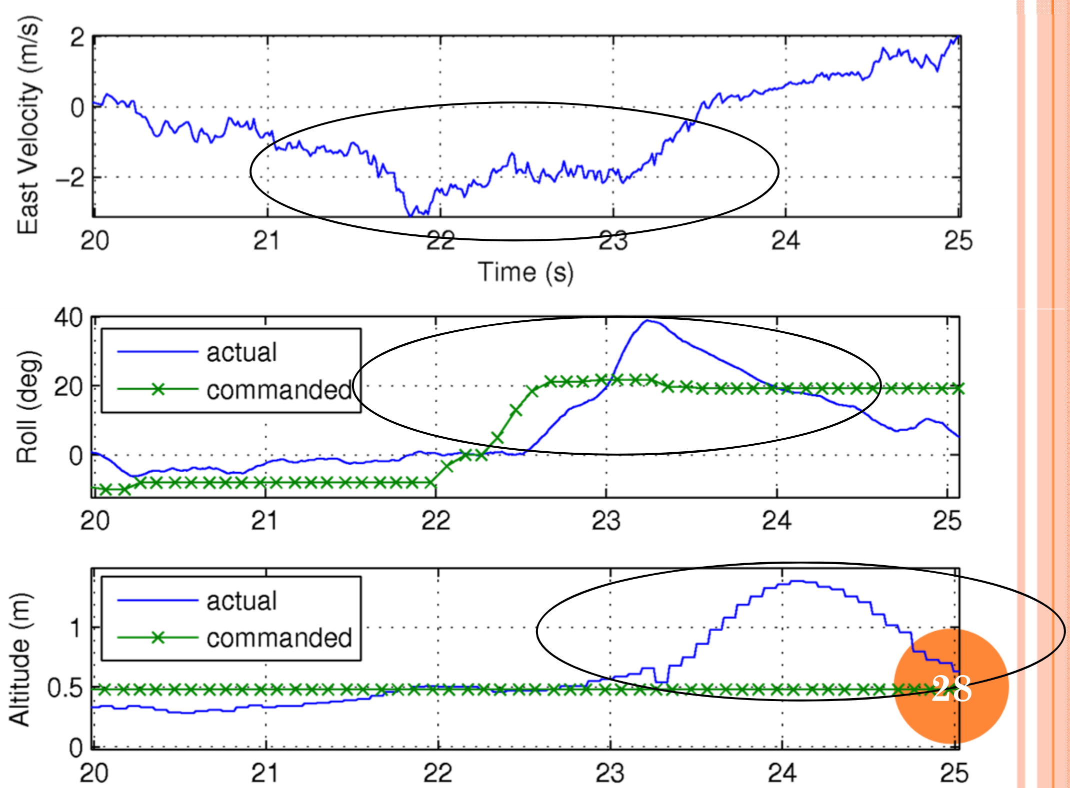


Angle of Attack in Flight

EFFECT OF ANGLE OF ATTACK IN FLIGHT

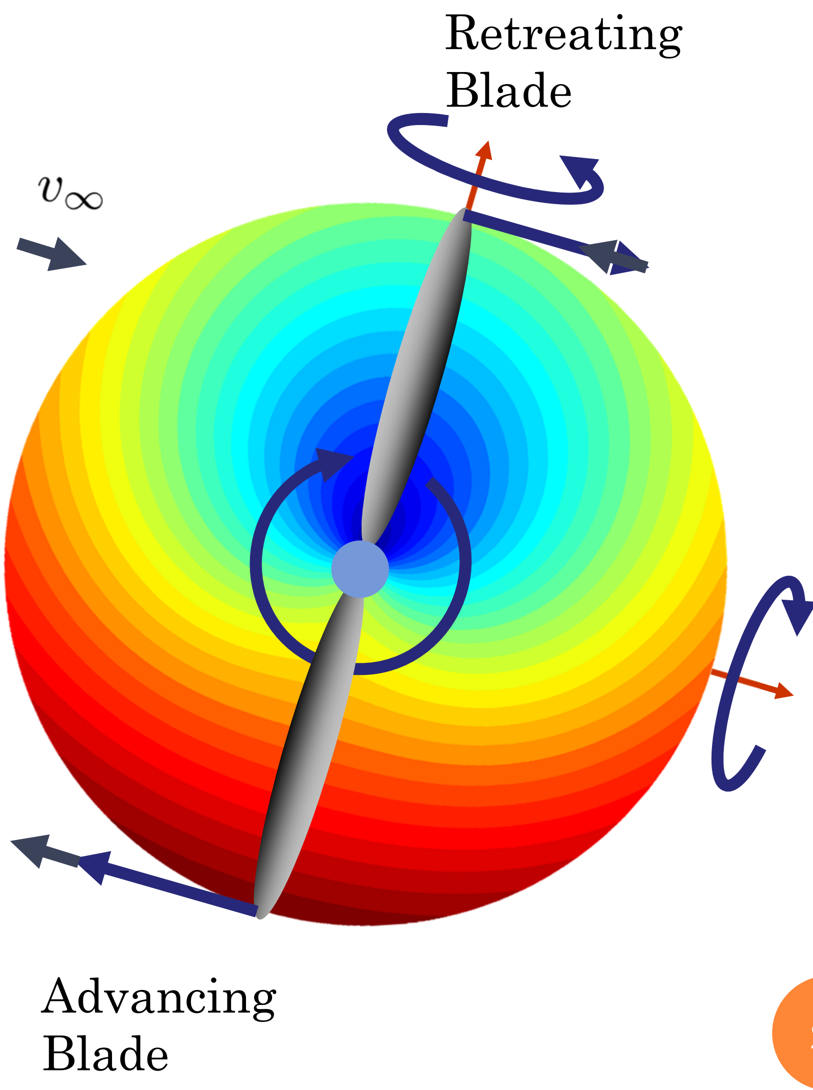


1. Fly at constant velocity.
2. Command large change in angle of attack
3. Results in significant “pop-up” in altitude

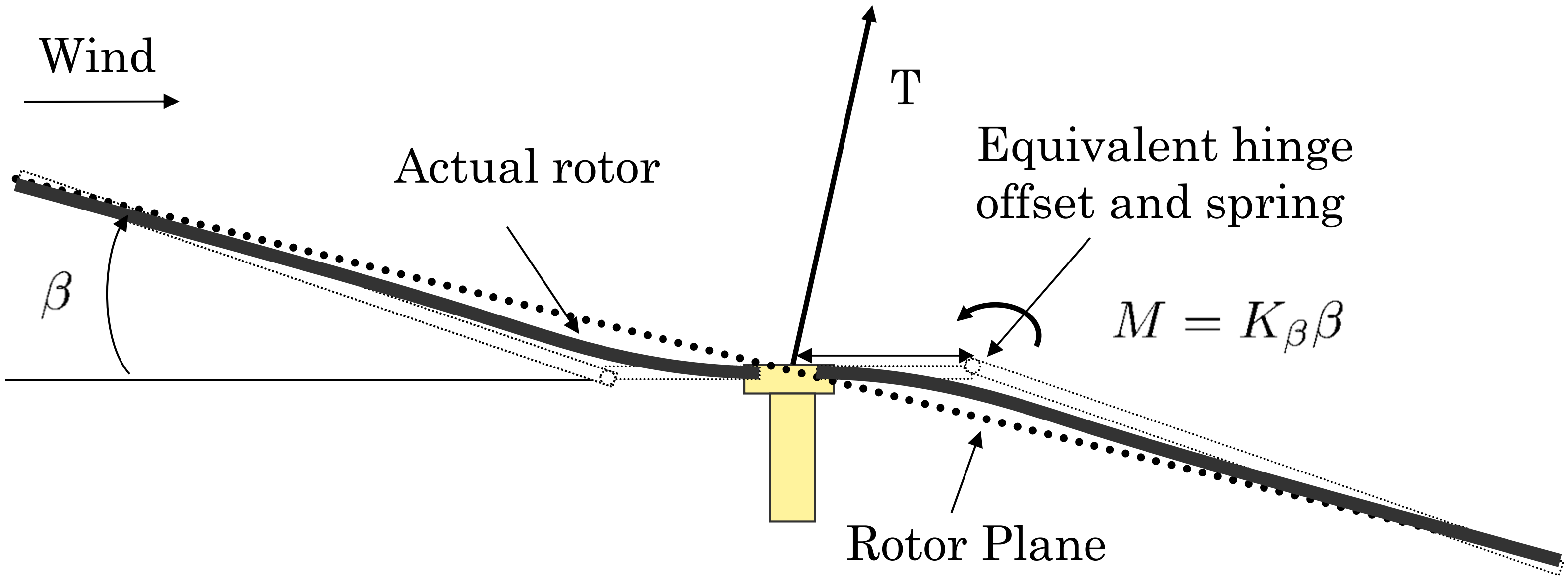


2. Blade Flapping

- Asymmetry of airflow relative to blade
- Produces asymmetric thrust
- Results in blade flapping
  - Lateral (roll) moment from force imbalance
  - Longitudinal (pitch) moment from deflection of rotor plane

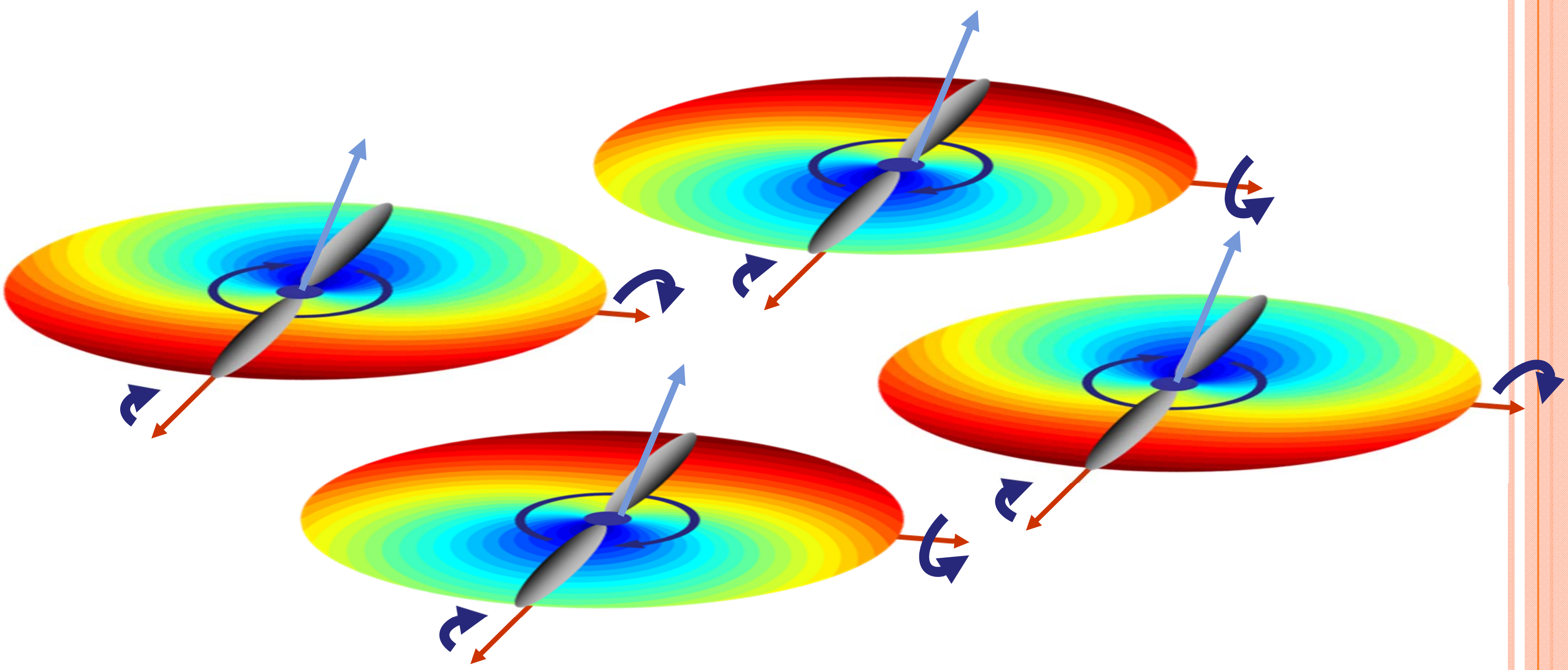


BLADE FLAPPING CONT'D



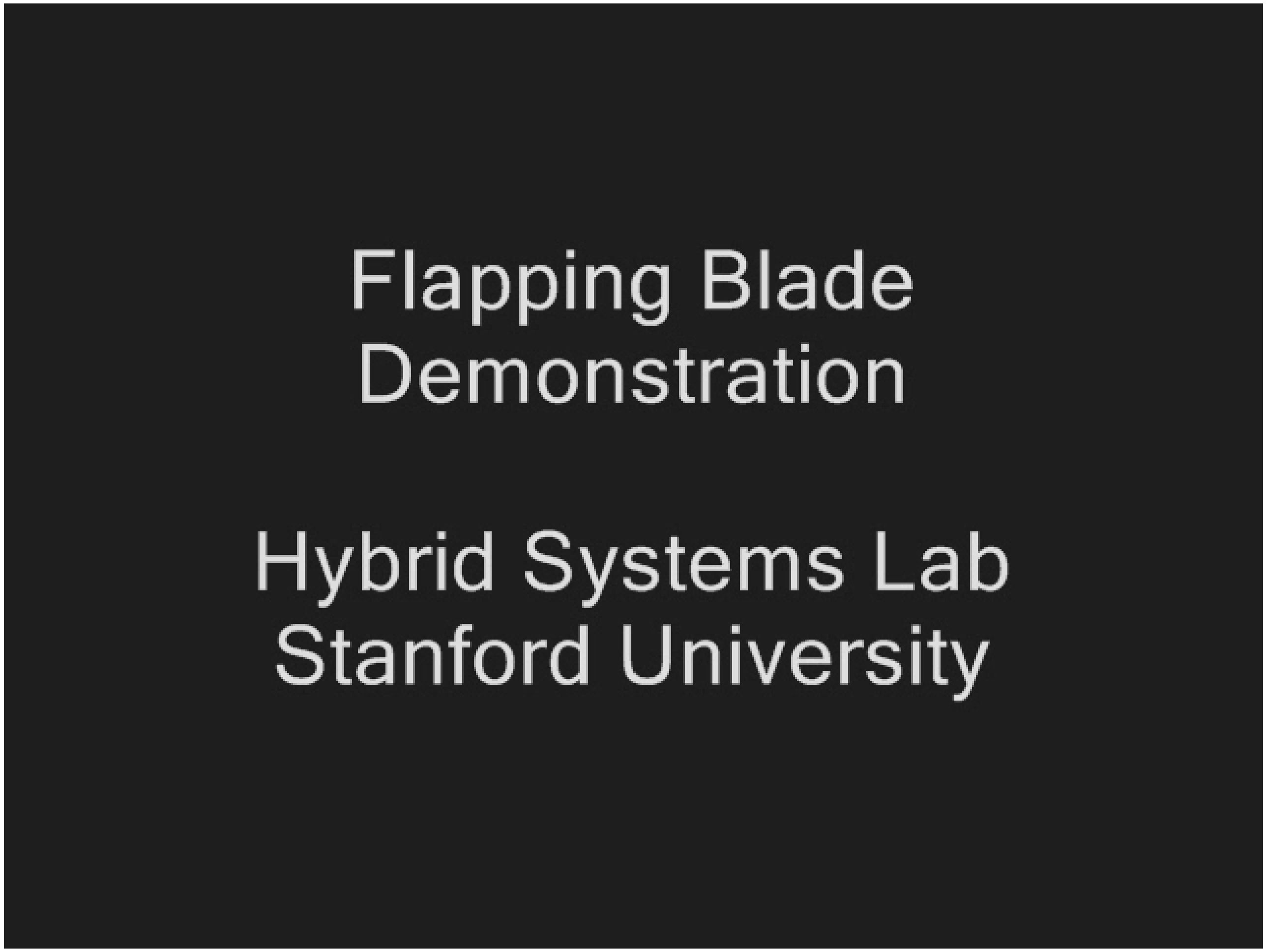
- Flexible blades with fixed hub modeled as hinged blades with a hinge offset and angular spring
  - Maximum deflection near 90°, affected by hinge offset, spring constant
- Thrust acts perpendicular to rotor plane
  - If CG is not in line with rotor plane, a moment results

Effect on Quadrotors



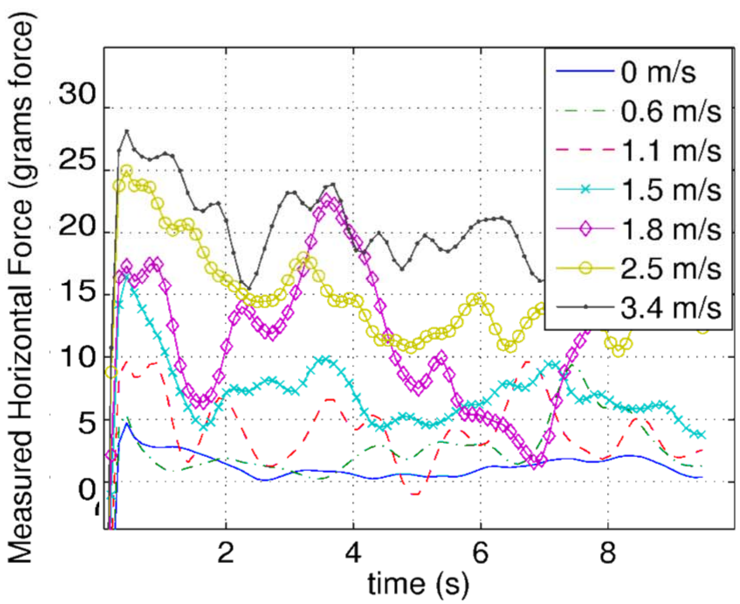
For quadrotors, lateral moments cancel !
Longitudinal moments cancel if CG is aligned with rotor plane

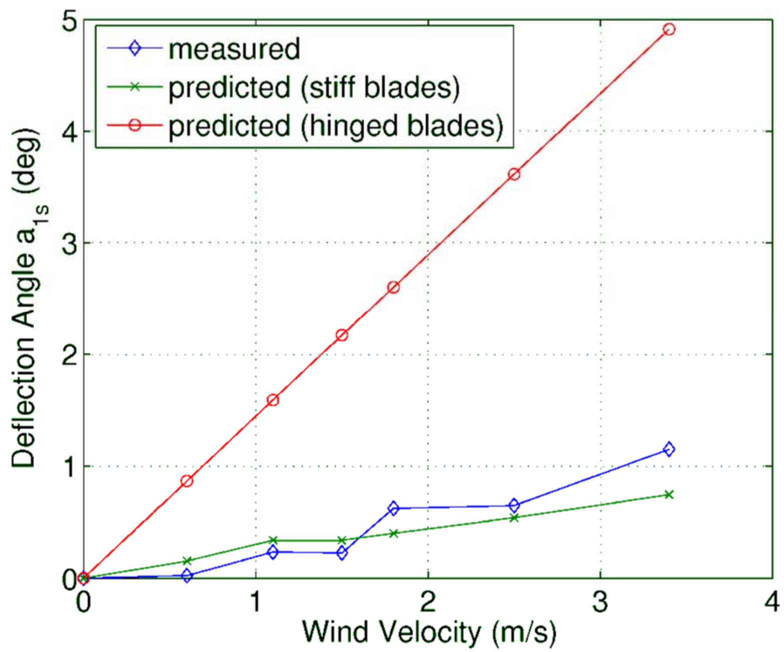
EXPERIMENT: BLADE FLAPPING



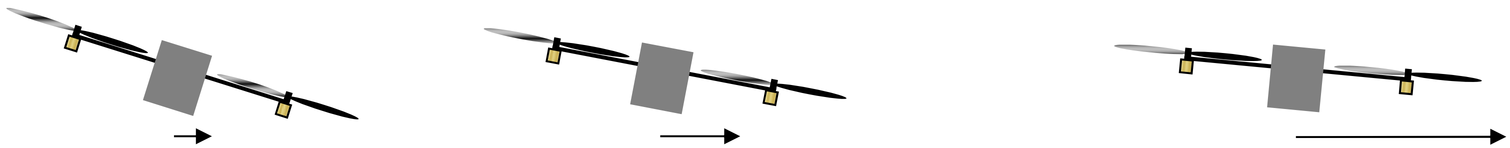
EXPERIMENTAL RESULTS: BLADE FLAPPING

- Measurement of lateral thrust in varying lateral wind conditions matches prediction
  - Blade spring constant measured
  - Hinge offset estimated from deflection profile

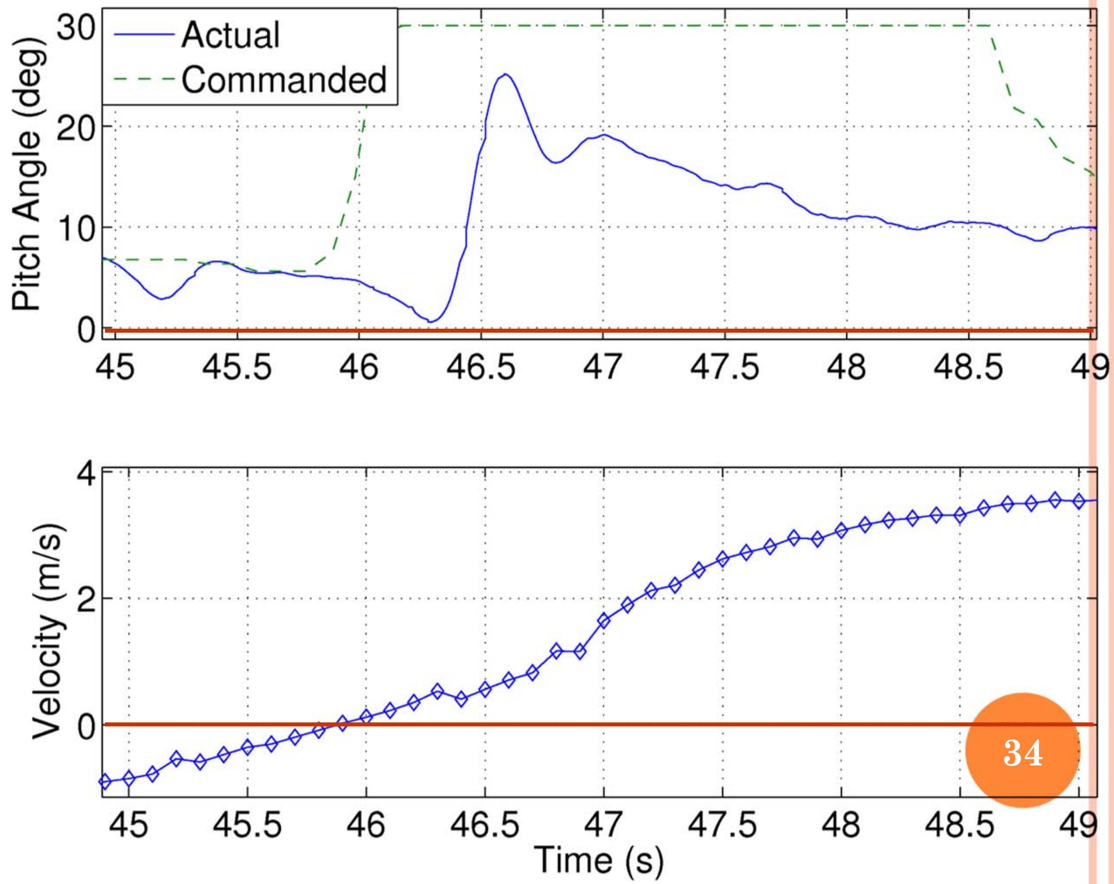




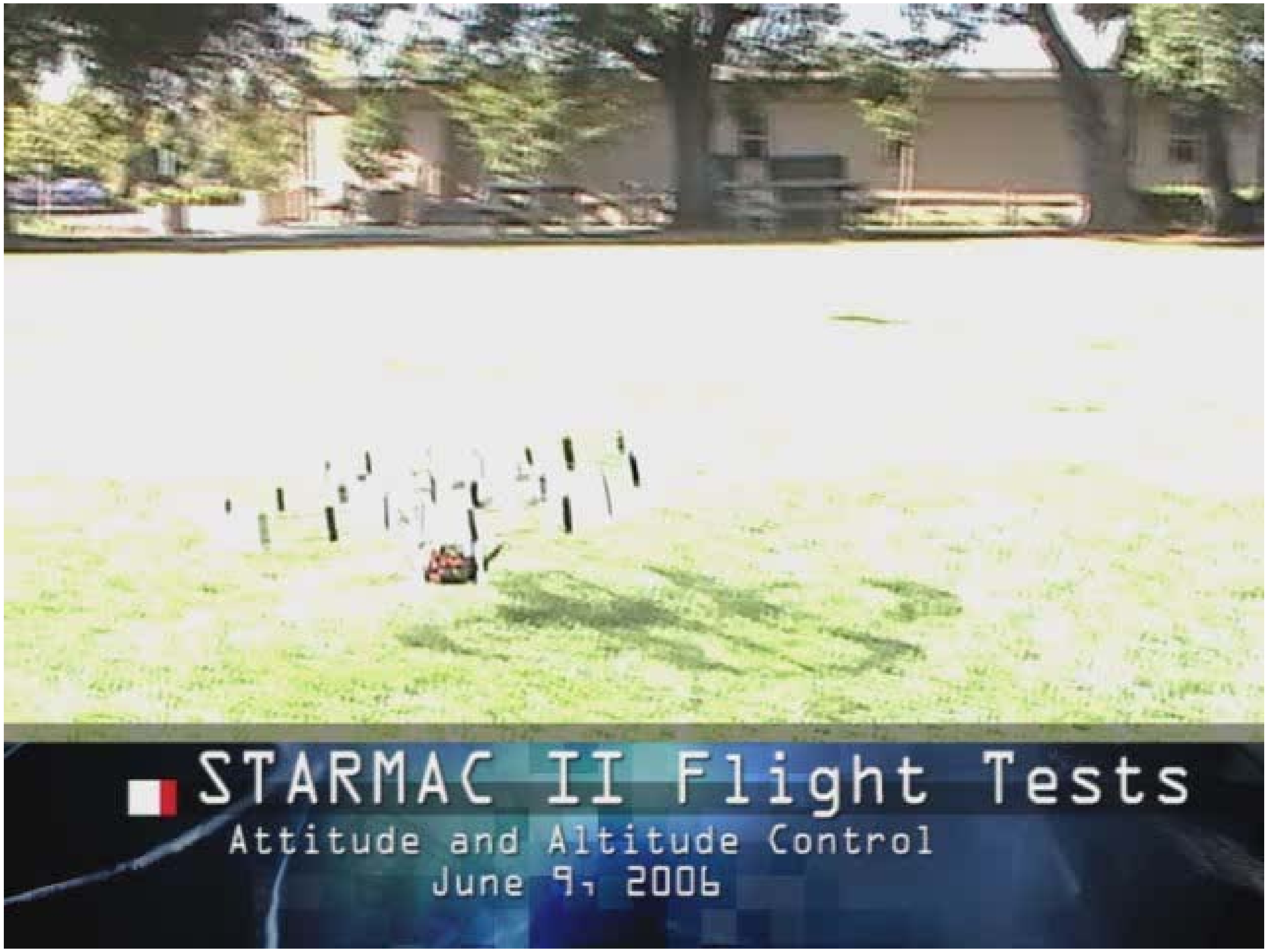
Effect of Blade Flapping in Flight



1. Large constant pitch angle commanded
2. As forward velocity increases, achieved pitch decreases

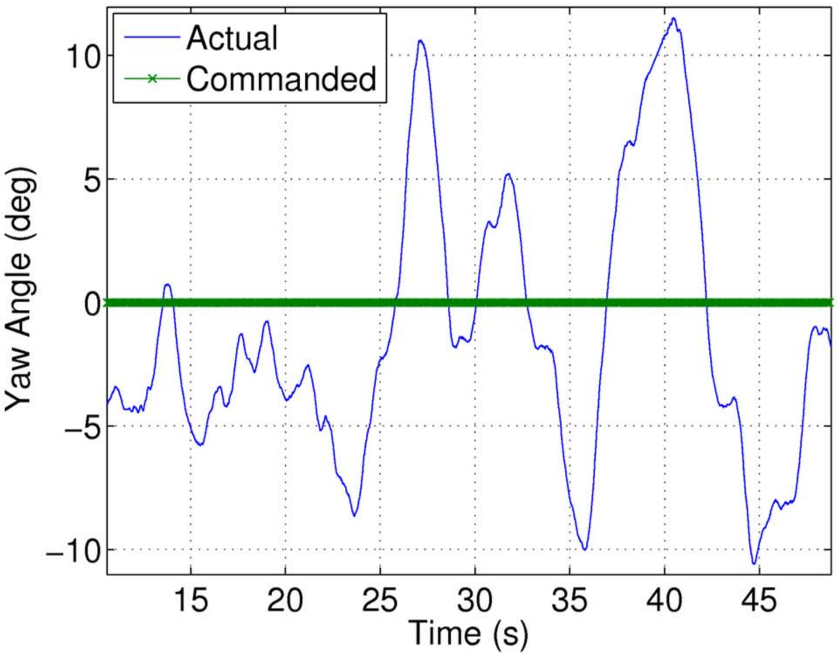


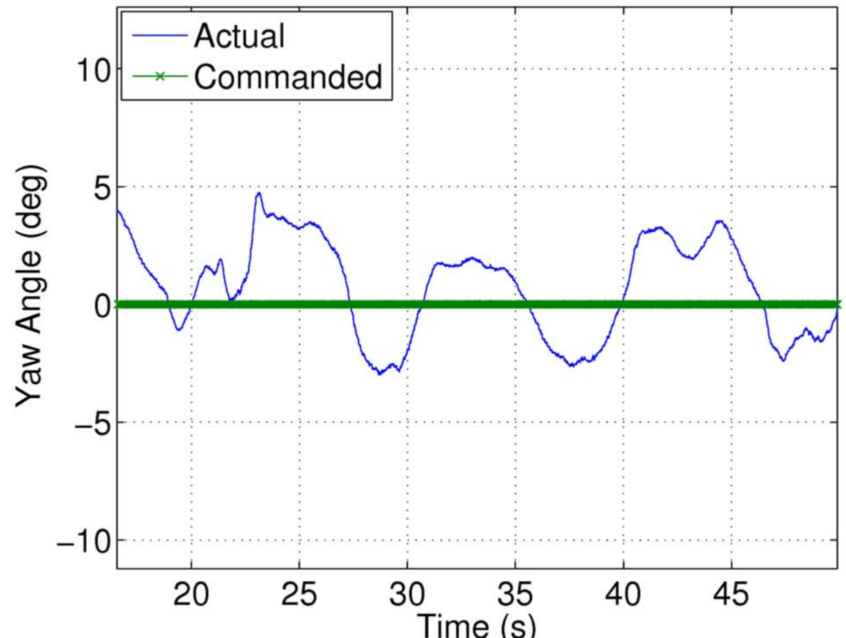
3. Airflow Disruptions

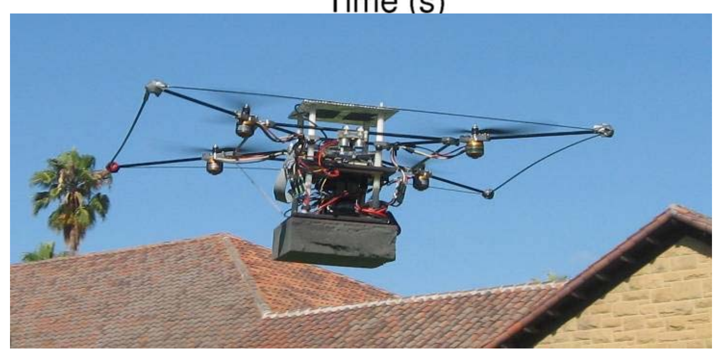


AIRFLOW DISRUPTIONS

- Yaw control accuracy with and without rotor shrouds

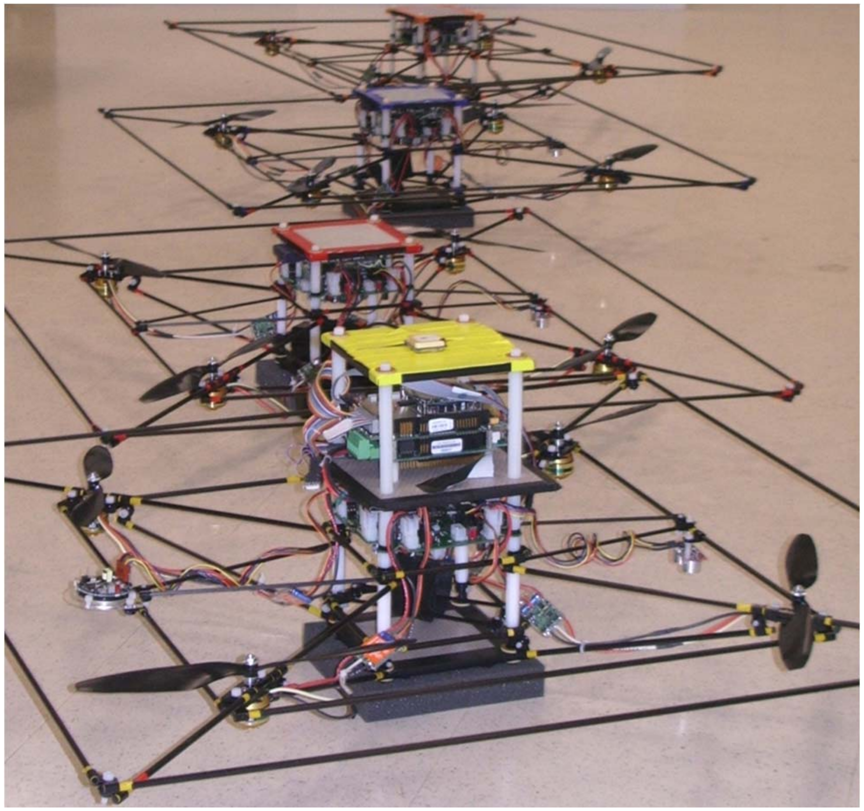




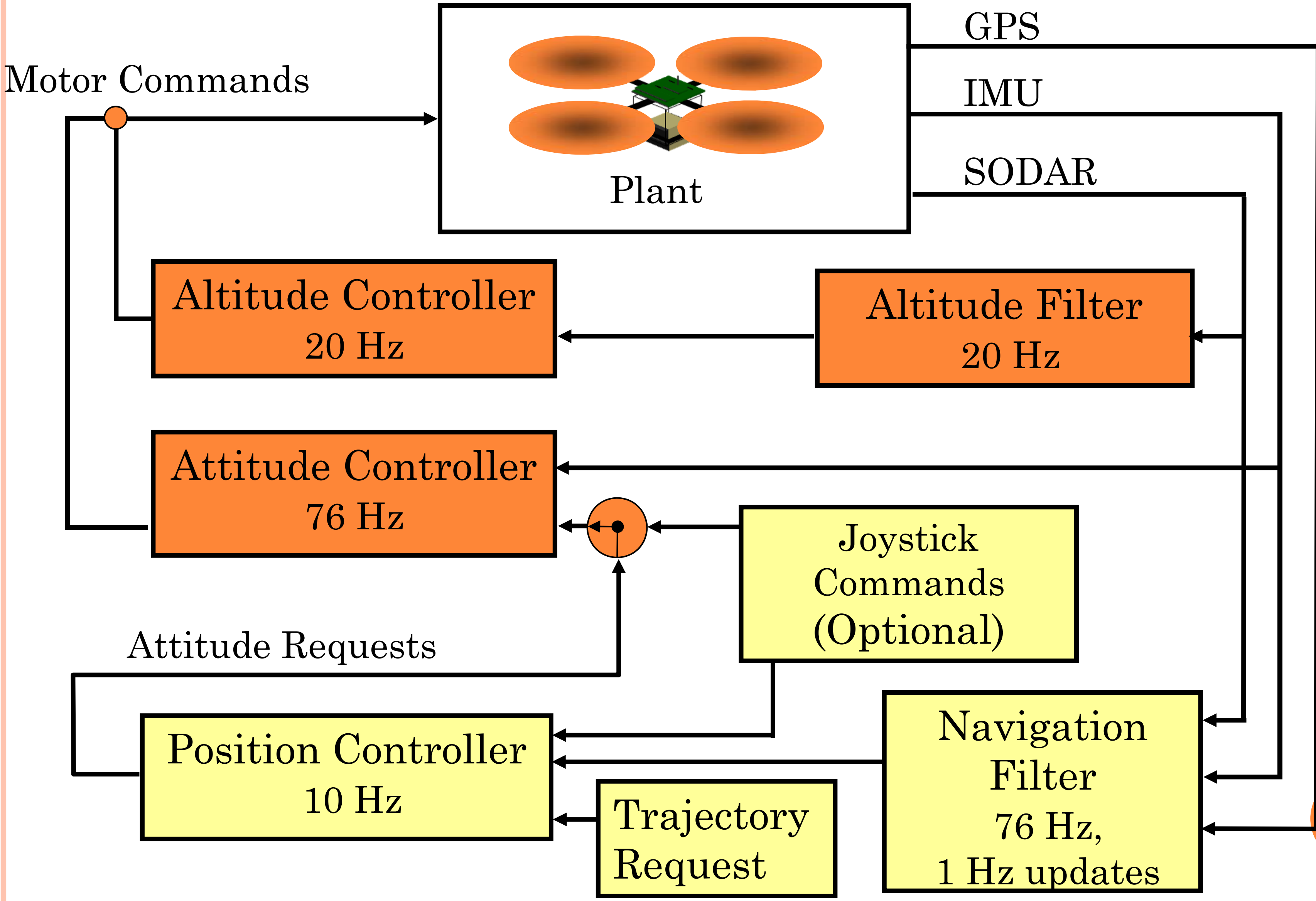


OUTLINE

- Introduction
- Platform Development
- Vehicle modeling
- Estimation and Control
  - Structure
  - Attitude Control
  - Altitude Control
  - Position Control
  - Trajectory Tracking
  - Wind Disturbance Rejection
- Mapping
- Motion Planning
- Multi-Vehicle Coordination

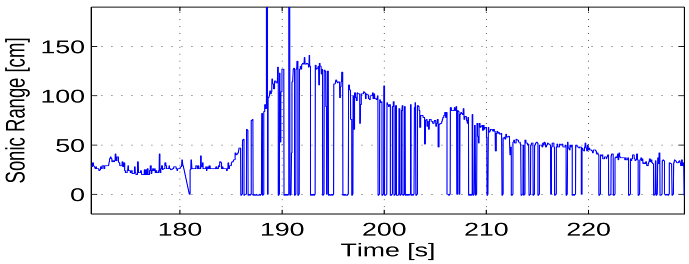


Estimation and Control Structure



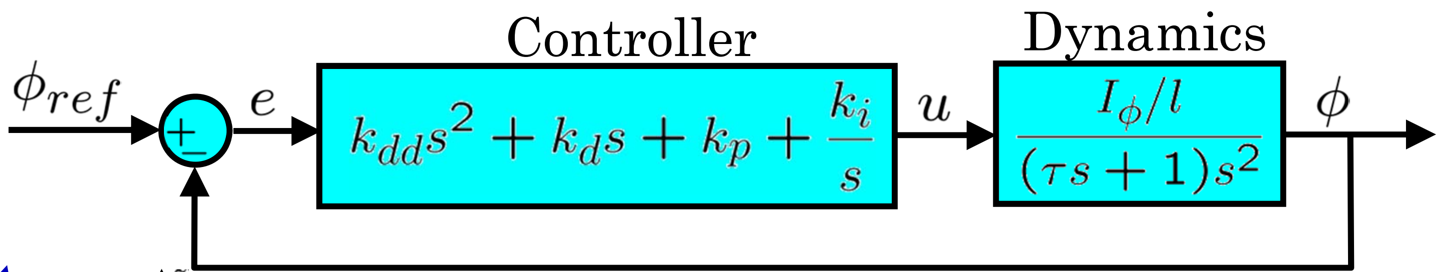
ESTIMATION & MEASUREMENT MODEL

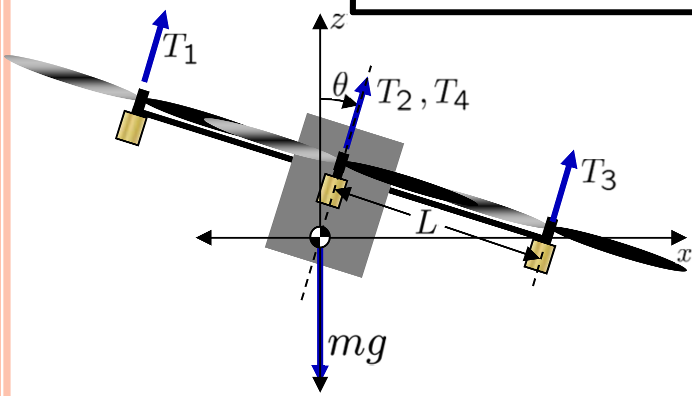
- Altitude filter
  - Outlier rejection
  - Kinematic Kalman Filter



- Standard EKF used for Navigation Filter
  - GPS – inertial position, velocity
  - Sodar, Barometer – inertial altitude
  - Gyros – body angular rates + biases
  - Magnetometers – body vector + current calibration
  - Accelerometers – body accelerations – gravity
    - Ignored during GPS measurement updates

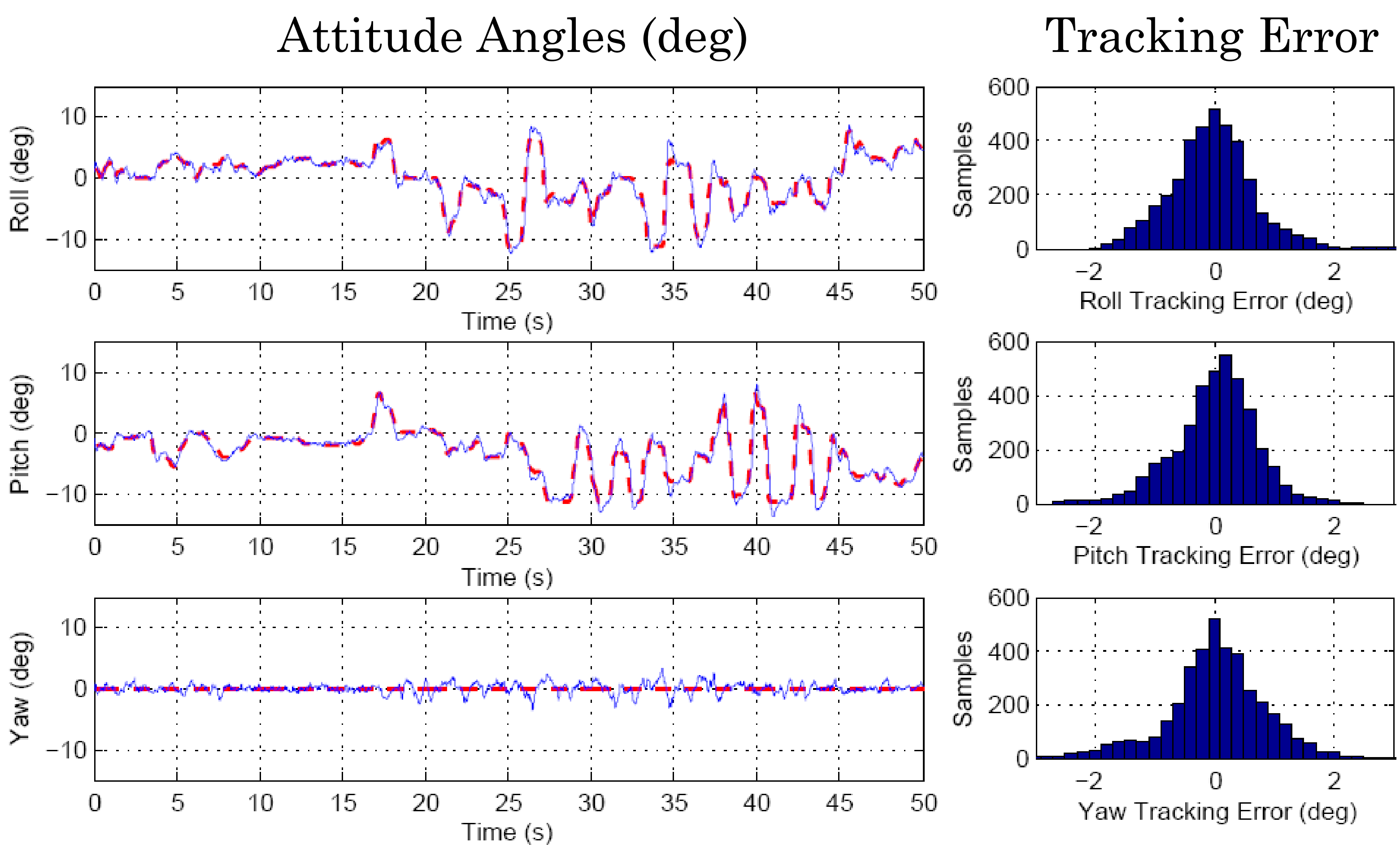
Quadrotor Attitude Control





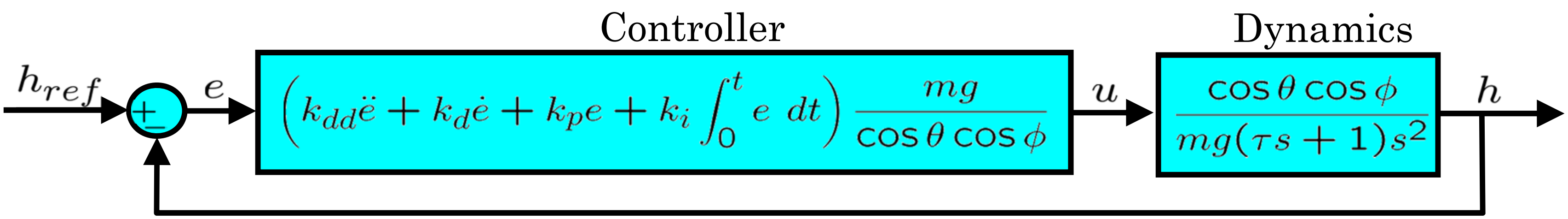
- Key Developments
  - Angular Accel. Feedback (specific thrust)
  - Command Tracking
  - Frame Stiffness
  - Tip Vortex Impingement

TRACKING REFERENCE COMMANDS



Root mean square error of 0.65°

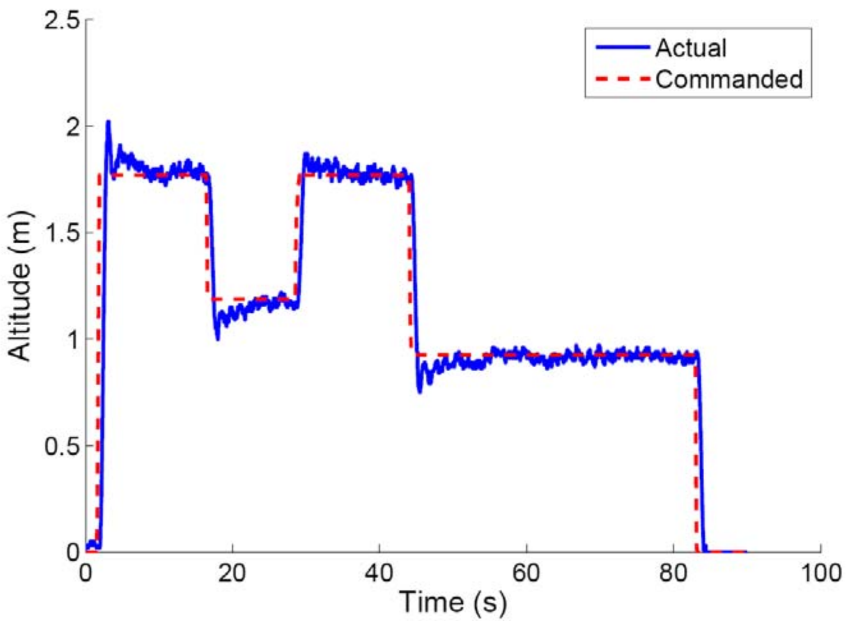
Quadrotor Altitude Control



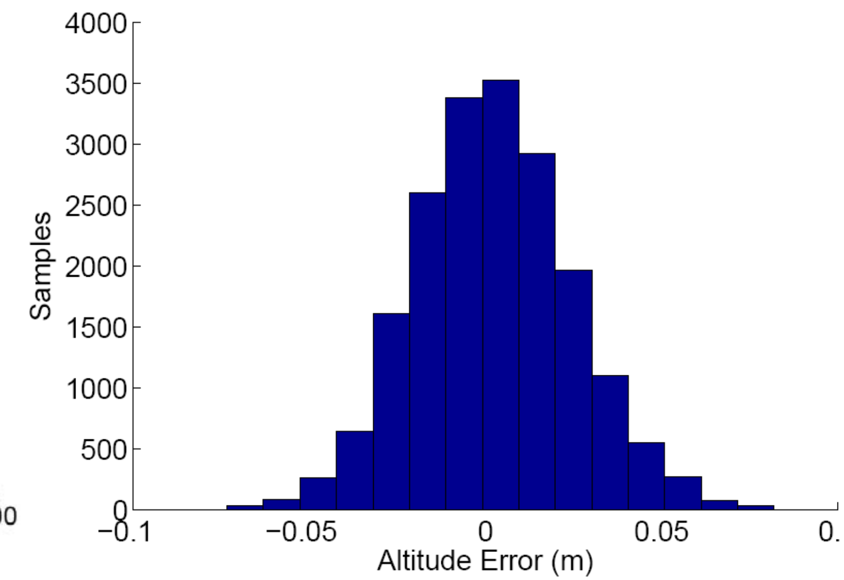
- Key Developments
  - Acceleration Feedback (specific thrust)
  - Tip Vortex Impingement
  - Tilt Compensation
  - Sensor Selection

TRACKING REFERENCE COMMANDS

Altitude



Error while in Hover



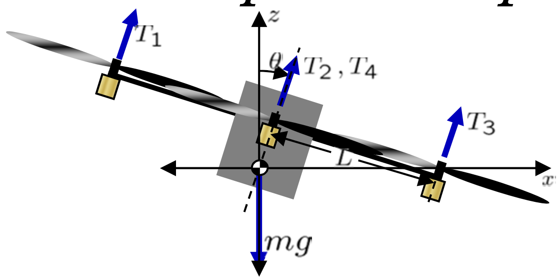
Root mean square error of 0.02 m in hover

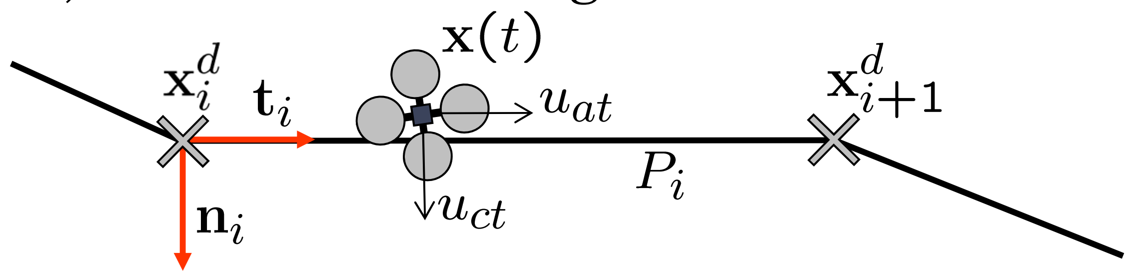
Disturbance Rejection - Indoor

DISTURBANCE REJECTION - OUTDOOR

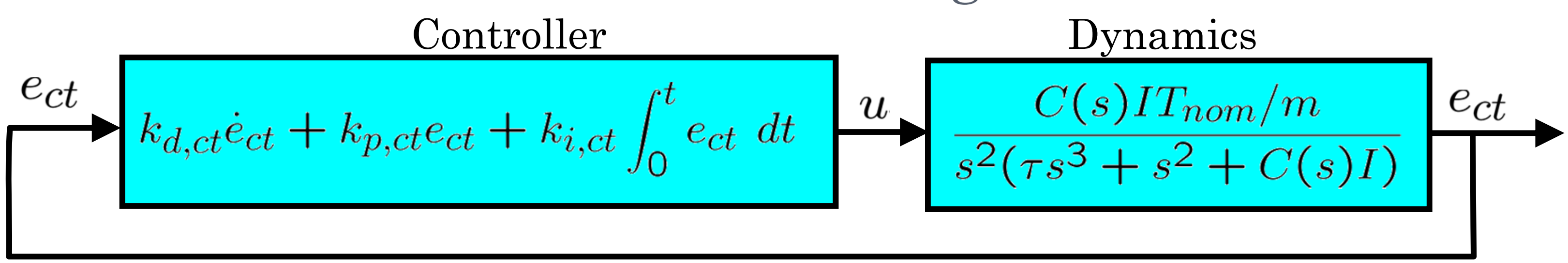
LINE TRACKING

- Follows *planned path*, no corner cutting

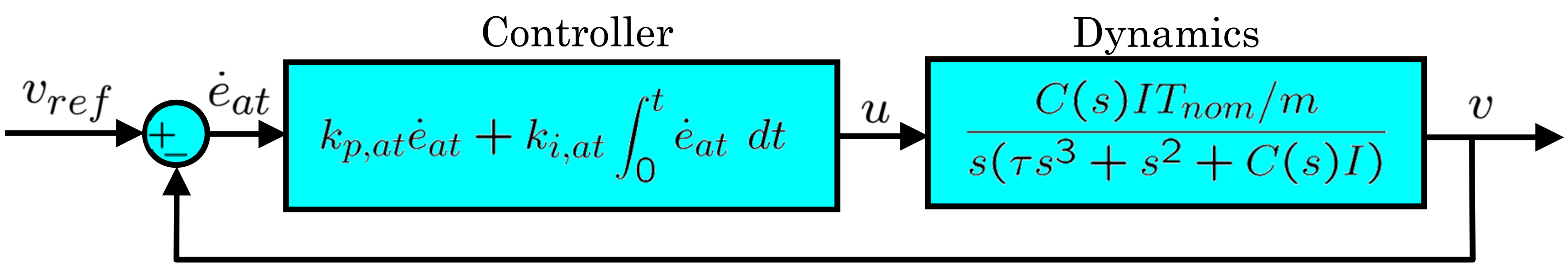




- PID Cross-Track Position Tracking



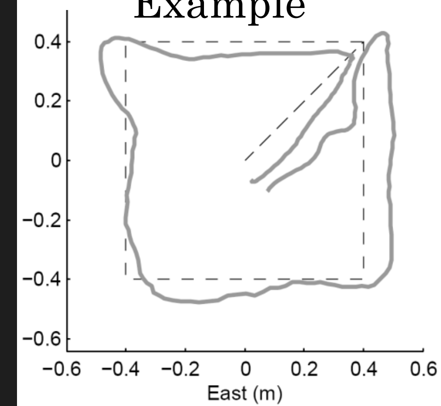
- PI Along-Track Velocity Tracking



Transform Commands to Body Frame

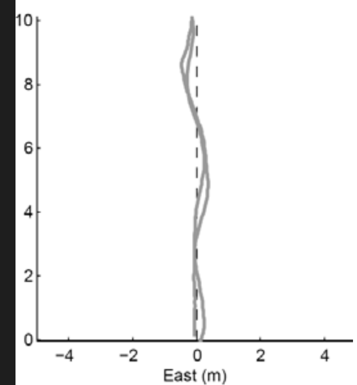
Quadrotor Position Control

Indoor Example



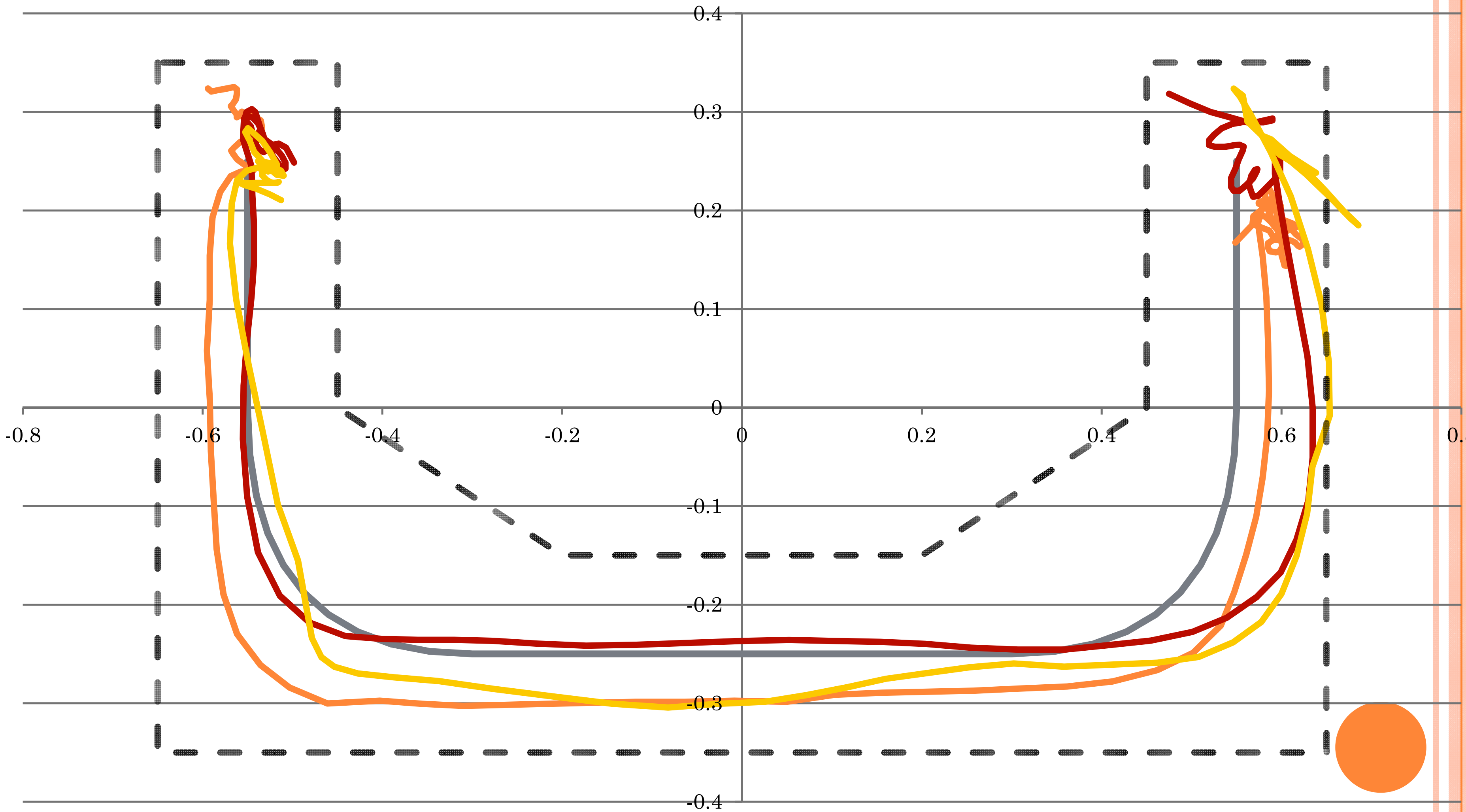
Quadrotor Position Control

Outdoor
Example



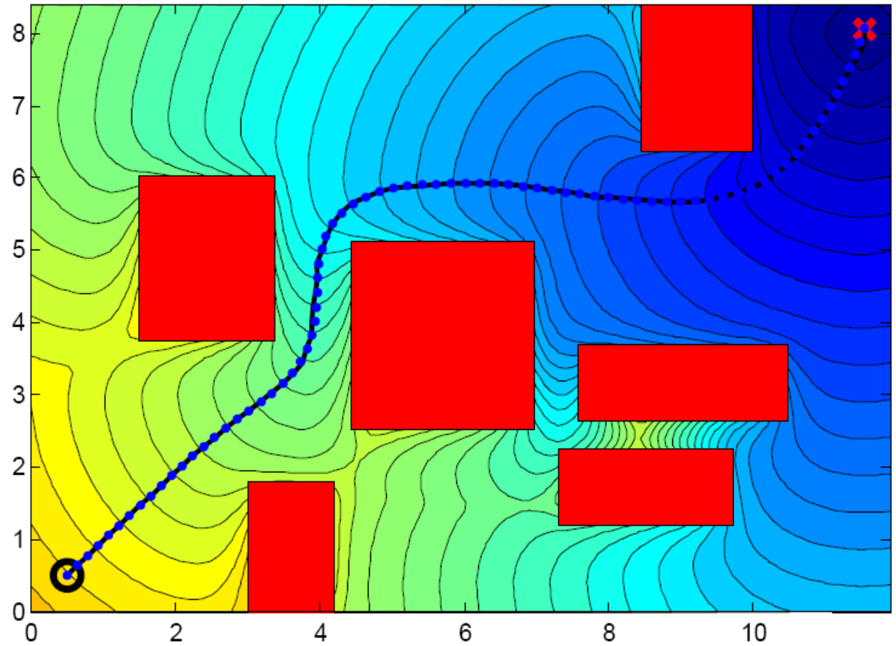
Aeryon Scout with PID Position Control

Aeryon Scout with PID Position Control



AR Parrot Drone Results

Time-Optimal Feasible Trajectory



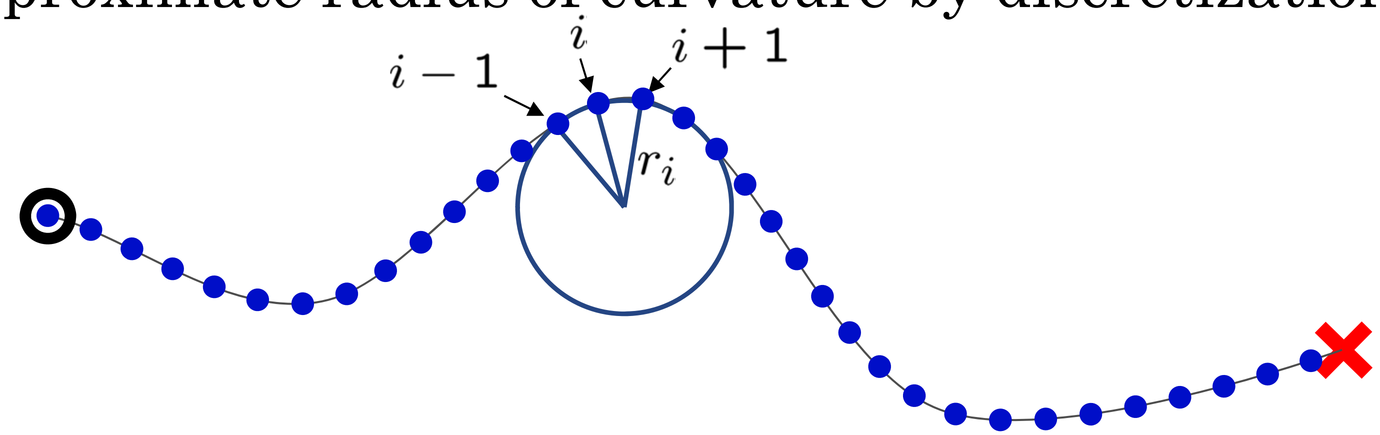
1. Find maximum speed for lateral acceleration constraint
2. Enforce reverse acceleration constraint
3. Enforce forward acceleration constraint
4. Compute required lateral acceleration

Feasible Trajectory Generation

- Speed limit for *cross-track* acceleration constraint

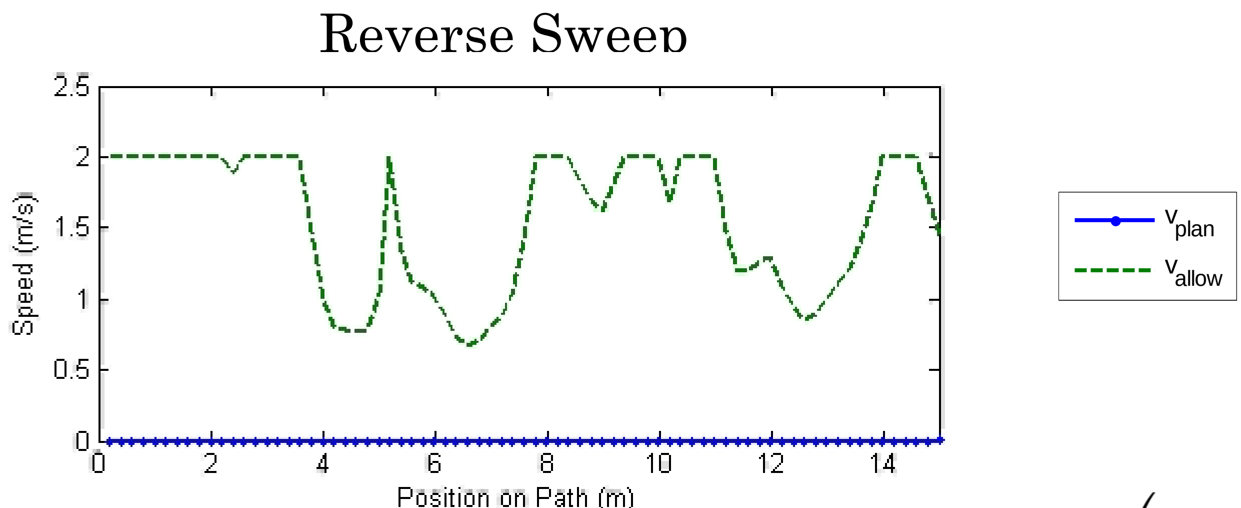
$$a_{ct,i} = \frac{v_i^2}{r_i}$$
$$\Rightarrow v_{i,allow} \leq \sqrt{a_{max} r_i}$$

- Approximate radius of curvature by discretization



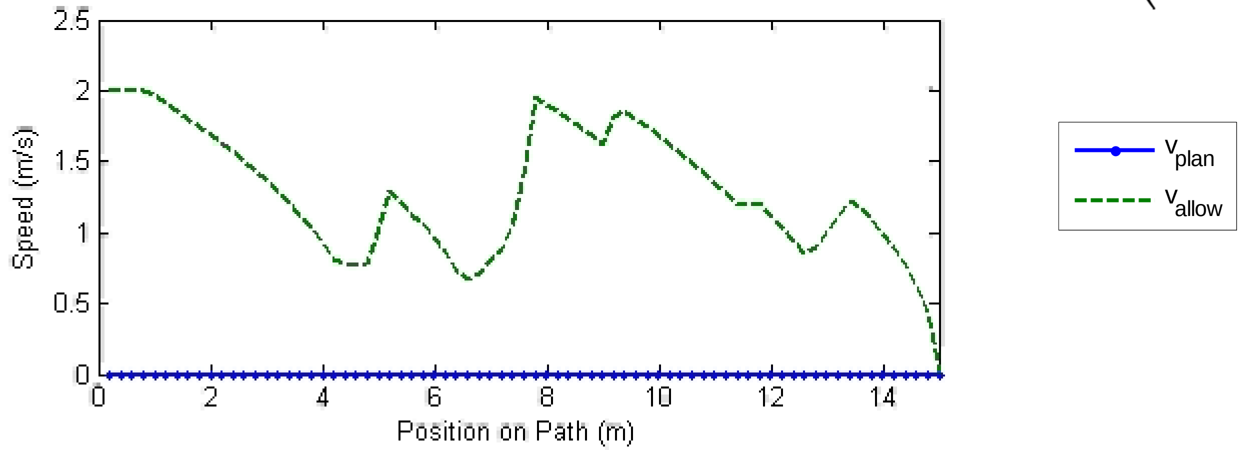
- Check for divide by zero (infinite radius of curvature)

Longitudinal Acceleration Constraint

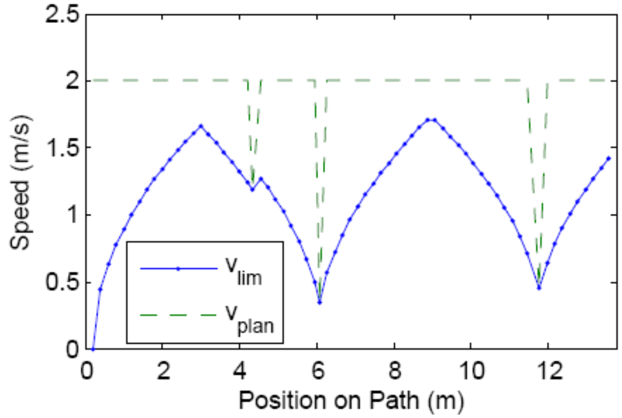


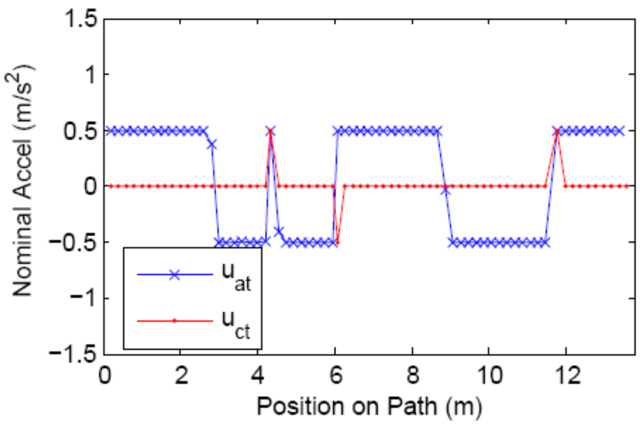
$$a_{at,i} = \min\left(a_{max},\ \frac{v_i^2 - \bar{v}_{i+1}^2}{2(s_i - s_{i+1})}\right)$$

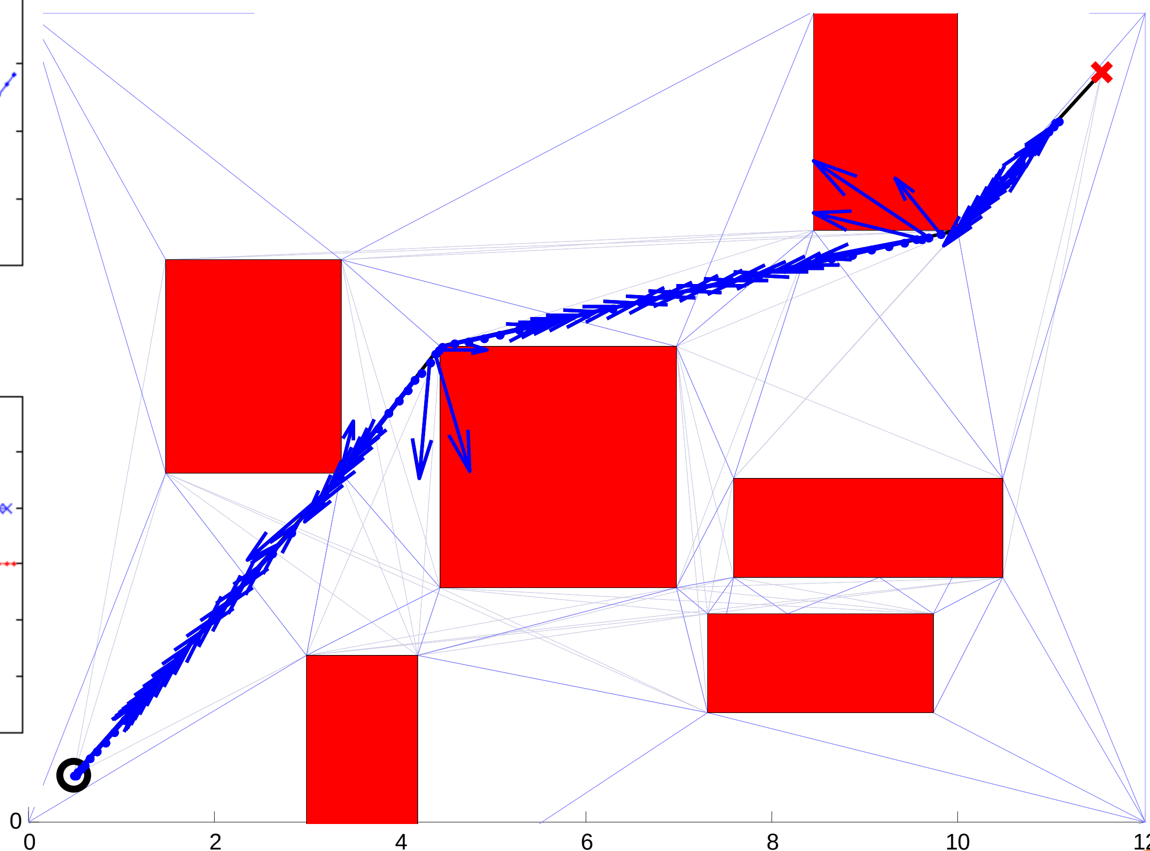
Forward Sweep



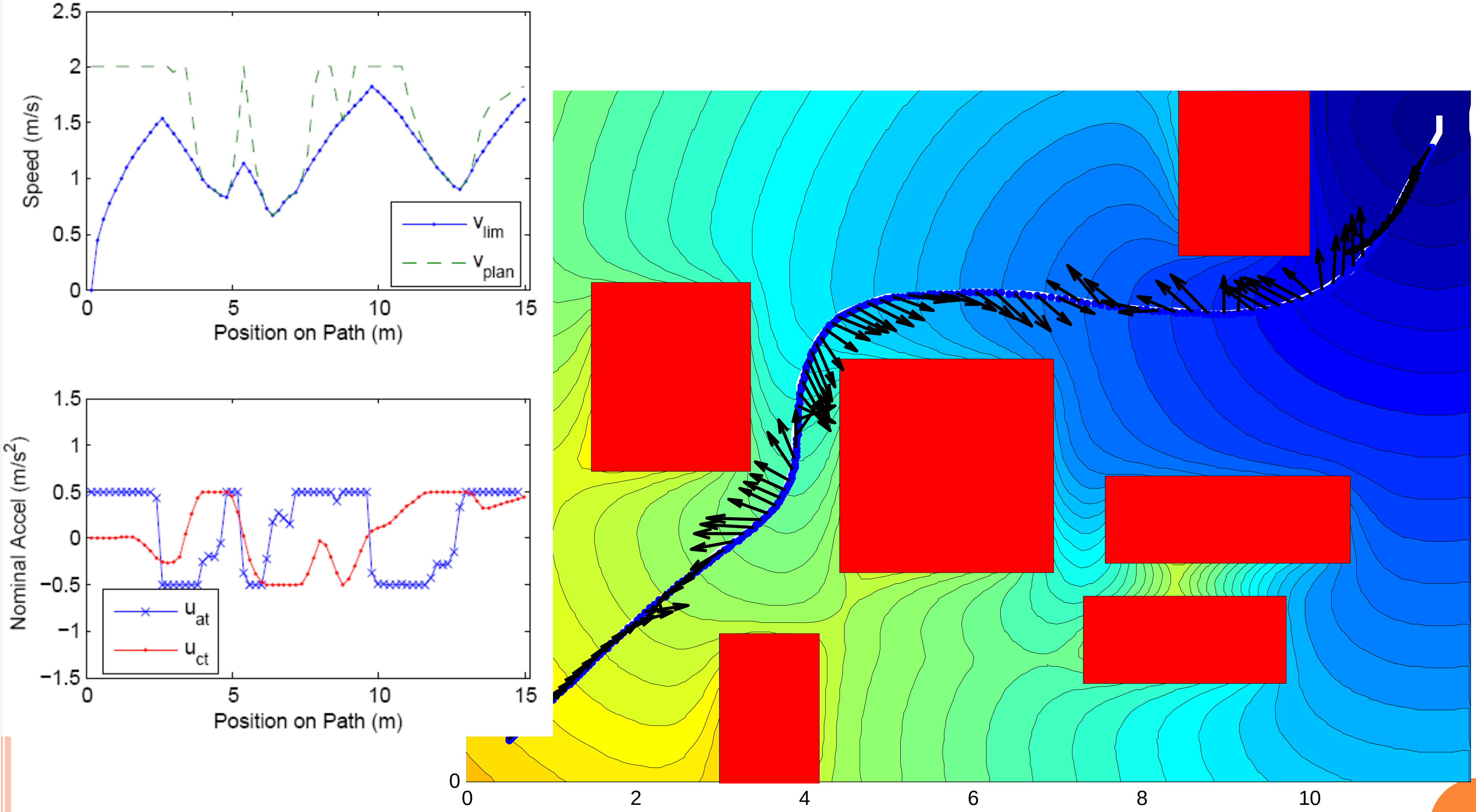
Visibility Graph Example







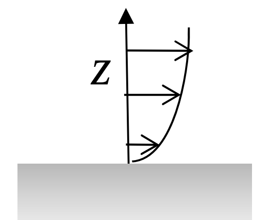
Fast Marching Example



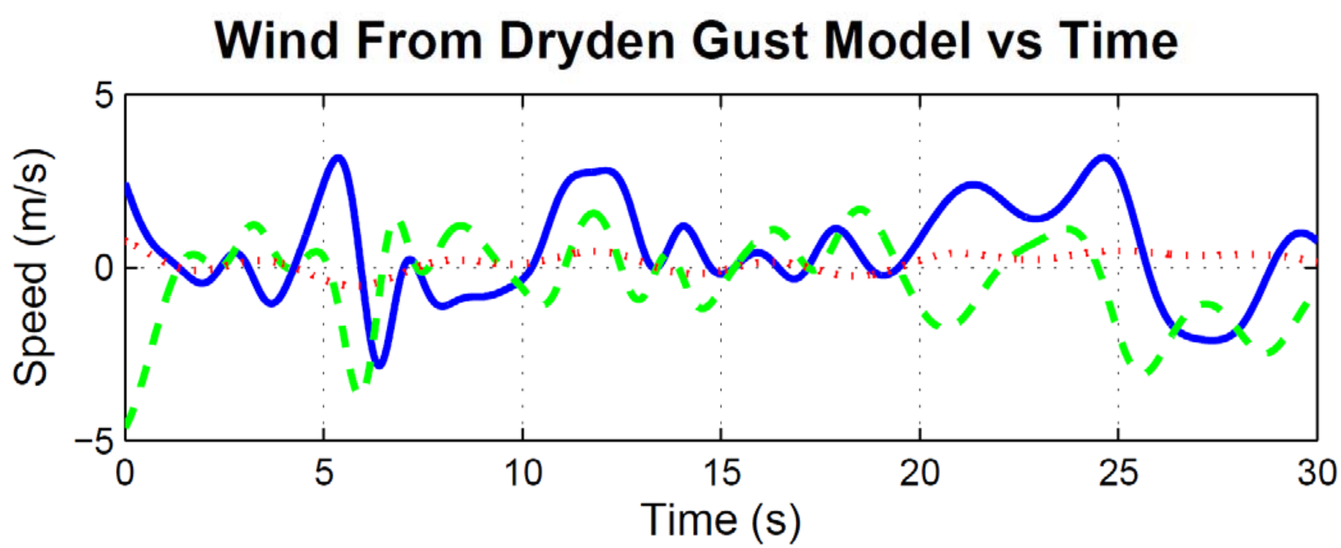
Wind Modeling

- Boundary layer effects for nominal wind speed

$$v_w(z) = \frac{1}{k} v_w^* \ln\left(\frac{z}{z_0}\right)$$



- Dryden wind gusts, linear combination of sinusoids
  - 0.1-1.5 rad/s $v_w(t) = v_w^0 + \sum_{i=1}^{n} a_i \sin(\Omega_i t + \gamma_i)$



$v_w^0 = 0$

Complete Force Models

- Thrust perpendicular to vehicle body plane from each rotor

$$T_i = C_\infty(v_\infty, \alpha, v_i)T_h$$
$$= C_\infty(v_\infty, \Theta, V_i)C_h V_i^2$$

- Drag parallel to vehicle body plane

$$D = C_D v_\infty + C_{bf} v_\infty \cos\alpha$$

  - Simple linear drag and blade flapping model reasonably accurate

- Moments are modeled similarly, not used in wind estimation

Wind Estimator

- Key idea: Since accelerometer measurements are fast relative to wind dynamics, use previous wind estimate in calculation of thrust

$$a_B = \sum_{i=1}^{4}\left(-C_\infty(v_I - \tilde{v}_w, \Theta, V_i) C_h V_i^2 \hat{\boldsymbol{z}}\right) + R_I^B\left(C_{bf}(v_I - v_w)\cos\alpha \hat{\boldsymbol{e}}_h + C_D(v_I - v_w)\hat{\boldsymbol{e}}_\infty\right)$$

- Solve for wind velocity, measurement model inversion

$$\hat{v}_w = v_I - \frac{R_B^I}{C_{bf}\cos\alpha \hat{\boldsymbol{e}}_h + C_D \hat{\boldsymbol{e}}_\infty}\left(\sum_{i=1}^{4}\left(C_\infty(v_I - \tilde{v}_w, \Theta, V_i) C_h V_i^2 \hat{\boldsymbol{z}}\right) + m a_B\right)$$

WIND COMPENSATOR

- North – East Position Control
- Reject accelerations caused by wind disturbance as they happen
- Rely on full thrust model to determine total thrust produced

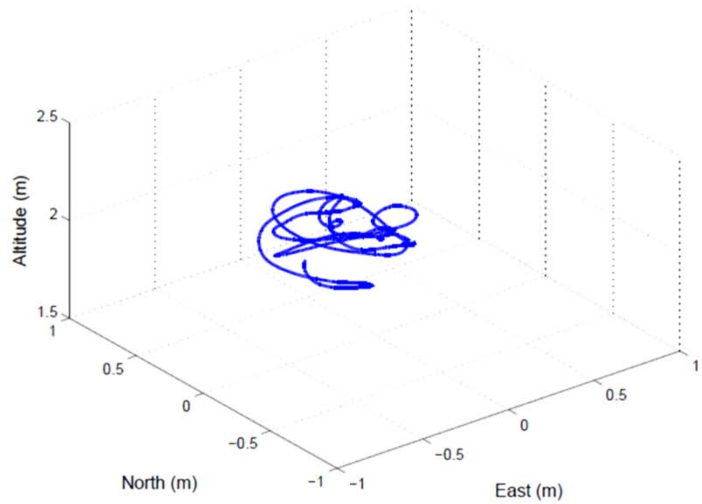
$$T_t = \sum_{i=1}^{4} T_i$$

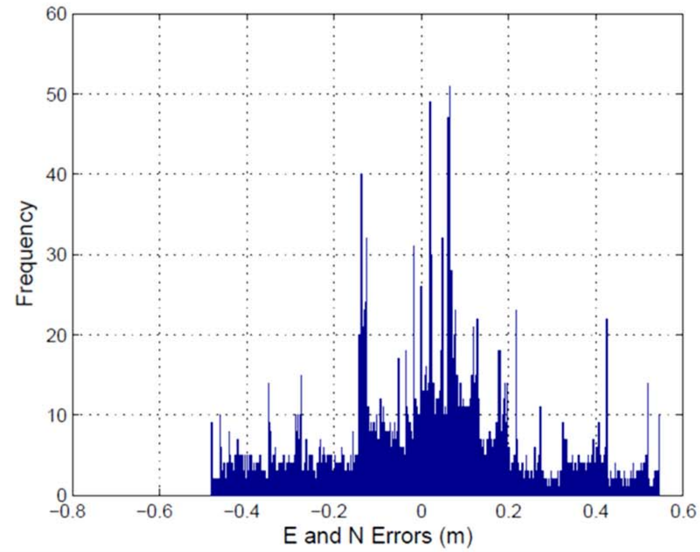
- For small angle commands, simplify to linear offset to roll and pitch request

$$T_t \begin{bmatrix} \sin \partial\theta \\ \sin \partial\phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_I^B C_D v_w \quad \Rightarrow \quad \partial\theta \approx \frac{C_D v_{w,x}}{T_t}, \quad \partial\phi \approx \frac{-C_D v_{w,y}}{T_t}$$

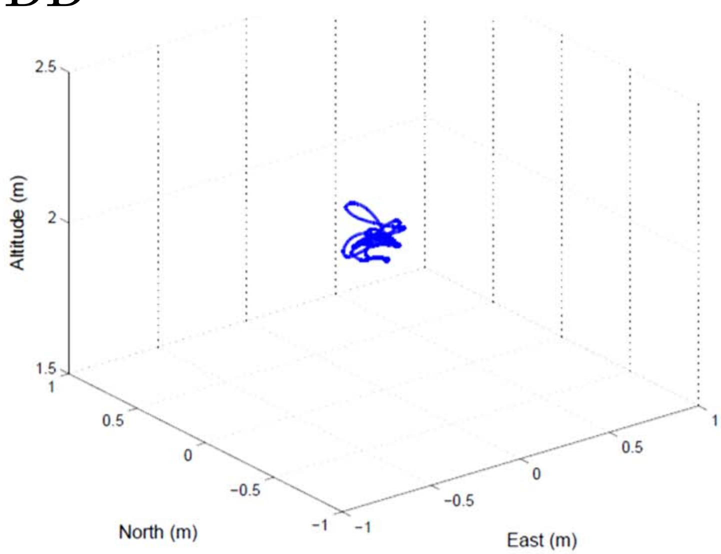
Hover Control Simulation Results

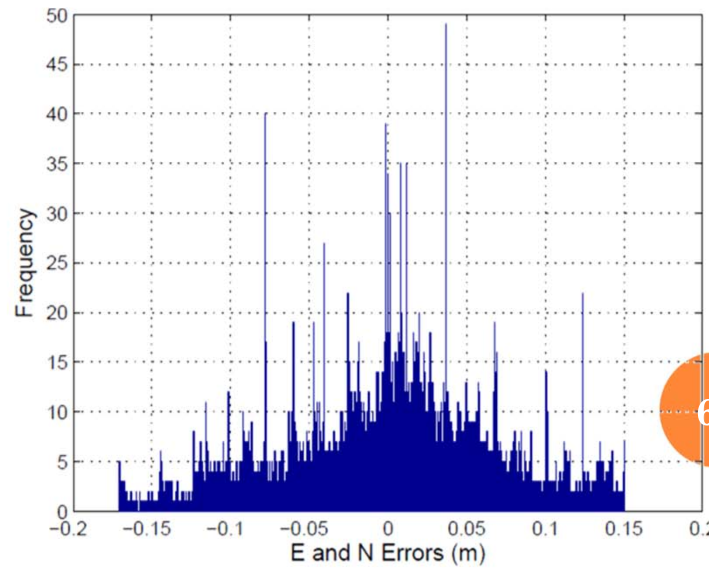
PID





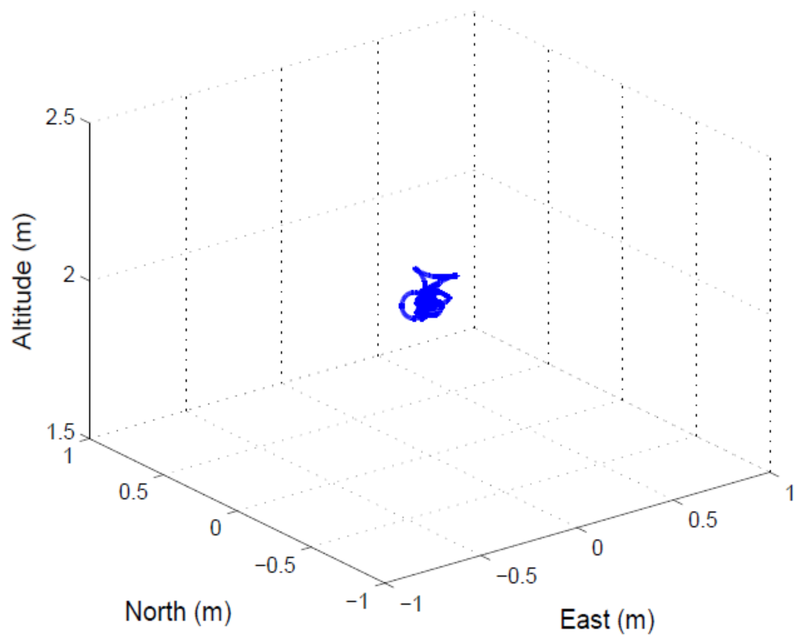
PID,DD

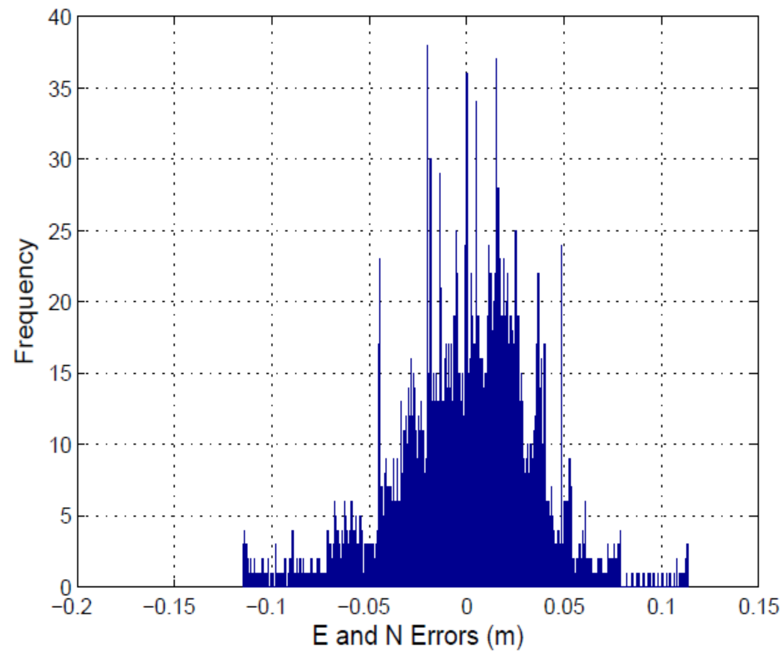




Hover Control Simulation Results

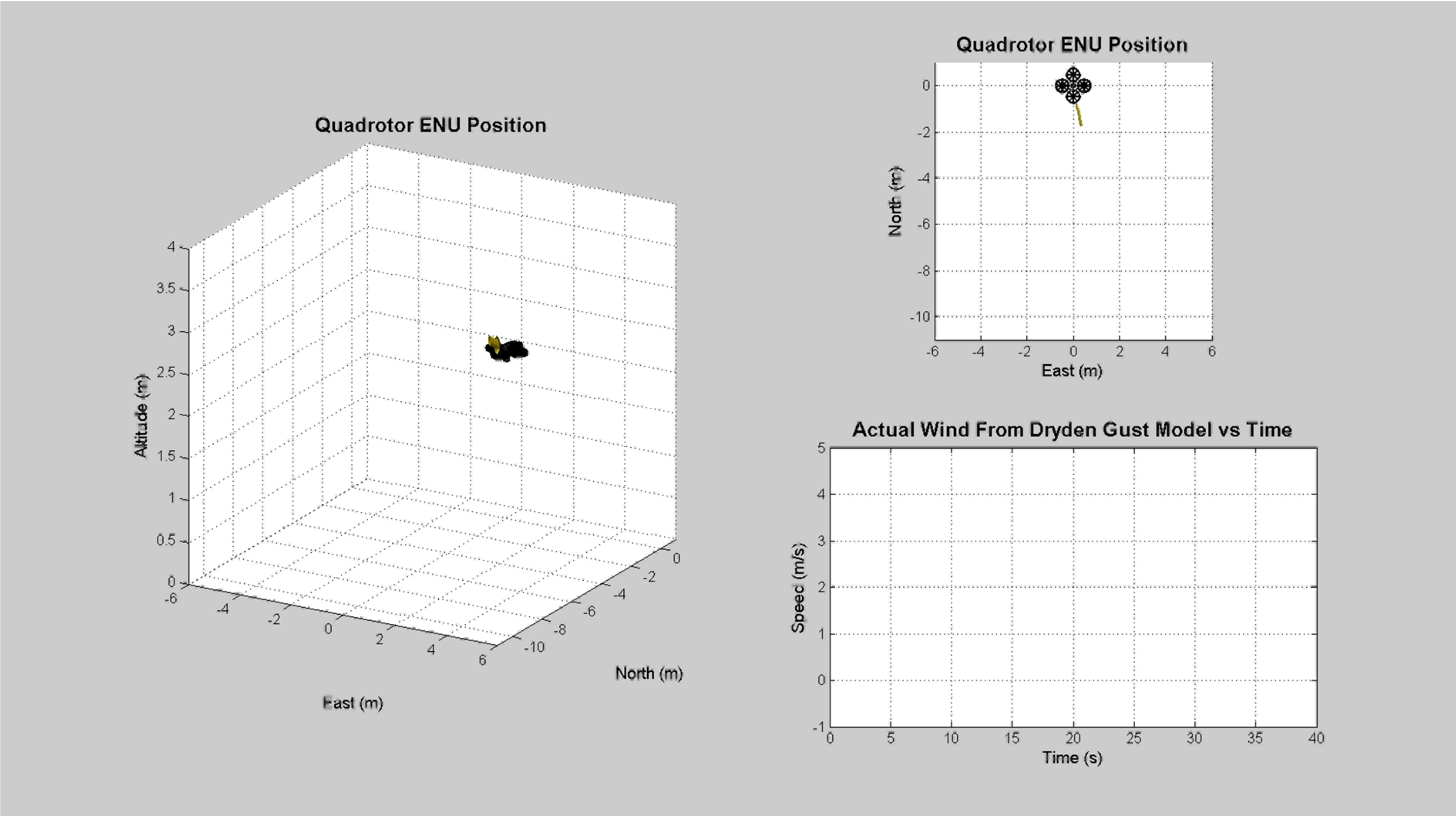
PID,DD + WC





- Improvement from ±0.4m to ±0.15m to ±0.1m error for the same wind

Circular Trajectory



MODEL-AIDED VISUAL-INERTIAL FUSION WITH WIND EFFECTS

Process Equation

$$\begin{bmatrix} {}^{E}\dot{\mathbf{p}} \\ {}^{B}\dot{\mathbf{v}} \\ \dot{\lambda} \\ \dot{\mathbf{\Theta}} \\ \dot{\boldsymbol{\beta}}_a \\ \dot{\boldsymbol{\beta}}_g \\ {}^{E}\dot{\mathbf{v}}_w \end{bmatrix} = \begin{bmatrix} {}^{E}_{B}R\,{}^{B}\mathbf{v} \\ {}^{B}_{E}R\mathbf{g} + ({}^{B}a_z - \beta_{az})\mathbf{e}_3 - \bar{D}_L({}^{B}\mathbf{v} - {}^{B}_{E}R^{T}\,{}^{E}\mathbf{v}_w) + \boldsymbol{\eta}_v \\ \eta_\lambda \\ \Xi({}^{B}\omega_g - \beta_g + \boldsymbol{\eta}_g) \\ \boldsymbol{\eta}_{\beta a} \\ \boldsymbol{\eta}_{\beta g} \\ \boldsymbol{\eta}_w \end{bmatrix}$$

Accelerometer Measurement

$$\mathbf{h}_i = -k_1 \Upsilon({}^{B}\mathbf{v} - {}^{B}_{E}R^{T}\,{}^{E}\mathbf{v}_w) + \Upsilon\boldsymbol{\beta}_a + \Upsilon\boldsymbol{\eta}_a$$

Vision Measurement

$$\mathbf{h}_{vp} = \lambda\,{}^{e}\mathbf{p} + \boldsymbol{\eta}_p$$

$$\mathbf{h}_{vo} = \mathbf{\Theta} + \boldsymbol{\eta}_{\mathbf{o}}$$

- Key Questions
  - Which states are observable?
  - Conditions on observability?

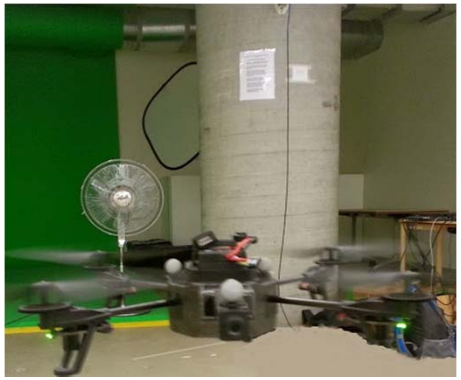
Results

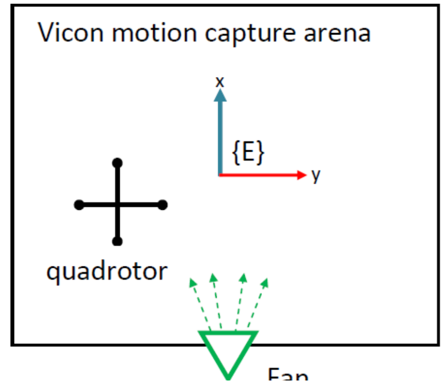
- Non-linear analytical observability analysis reveals that all states except for body frame z component of wind velocity are observable (in a locally-weak observability sense)

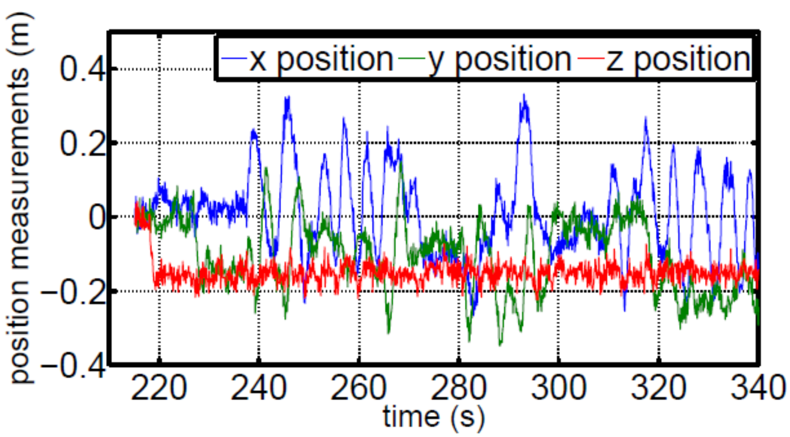
- Observability depend on the quadrotor maintaining non-zero accelerations **in either body frame x or y directions**

- Unobservable component of wind can be removed by assuming wind velocity is smoothly varying **in inertial frame.**

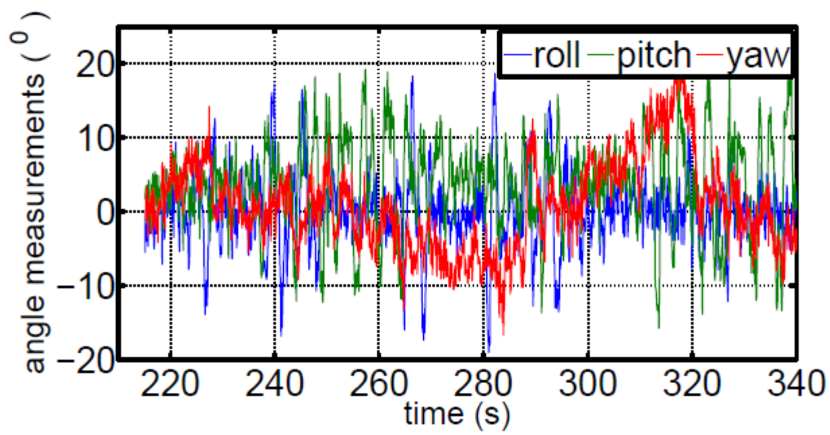
Experimental Validation





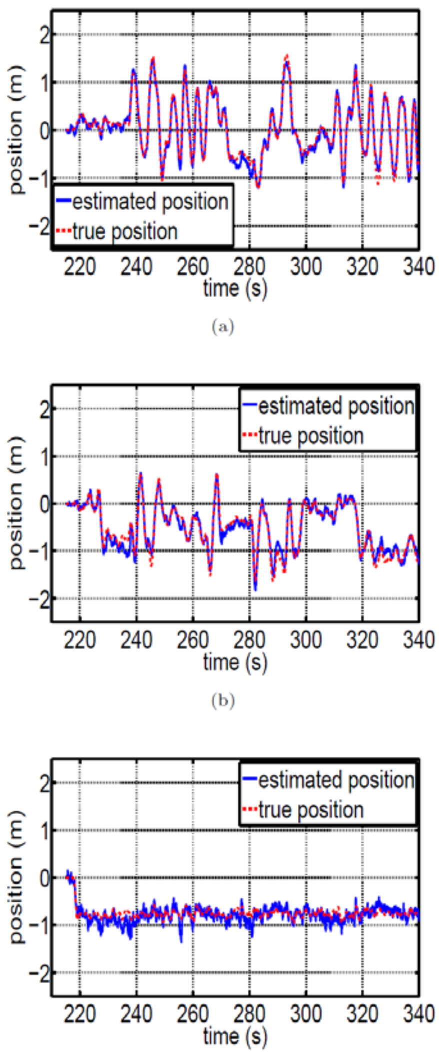


Simulated VSLAM Position
Scale:1/5
Noise: 1m standard deviation

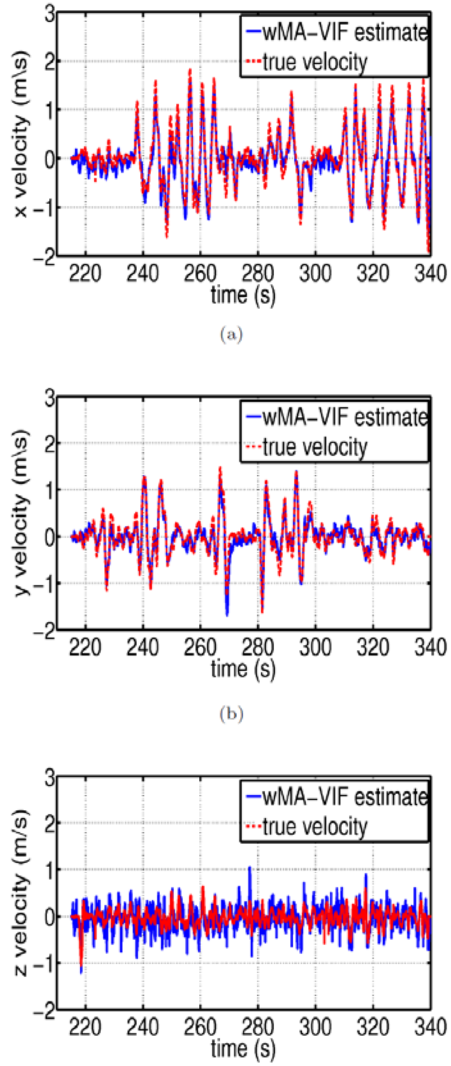


Simulated VSLAM orientation
Noise: 2 degrees standard deviation

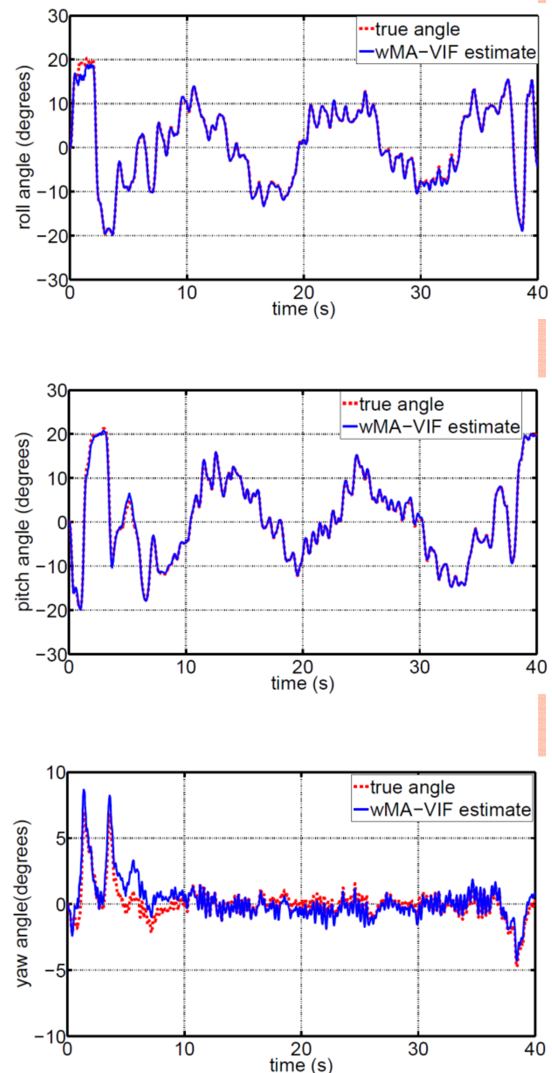
EXPERIMENTAL RESULTS



Position

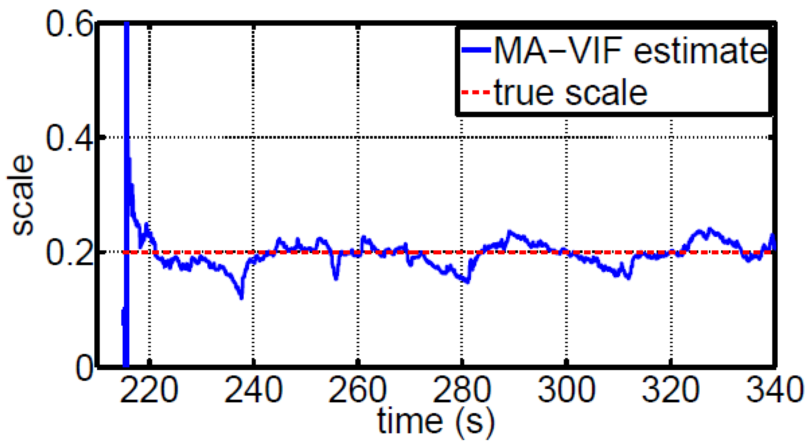


Velocity

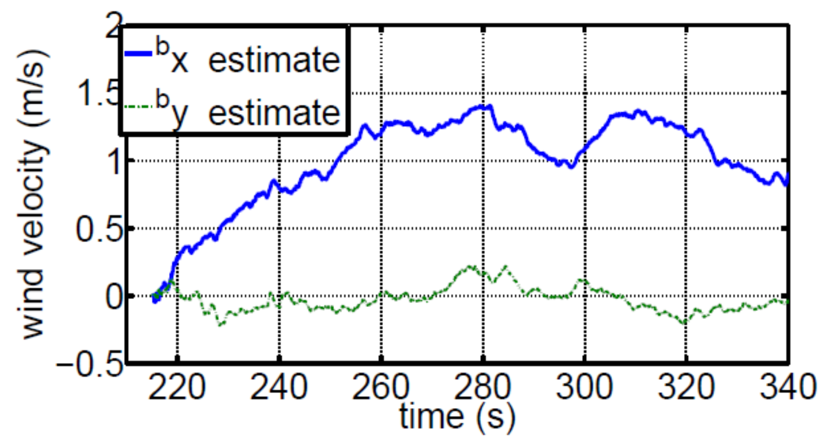


Orientation

Experimental Results

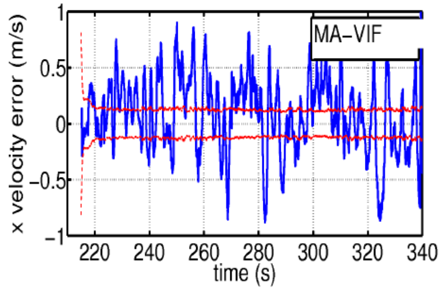


Scale Estimate

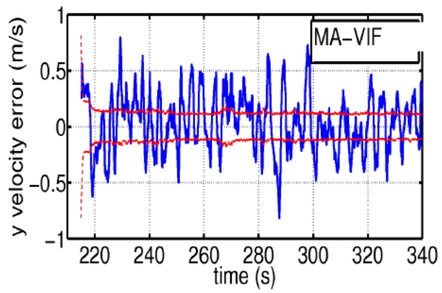


Wind Velocity Estimate

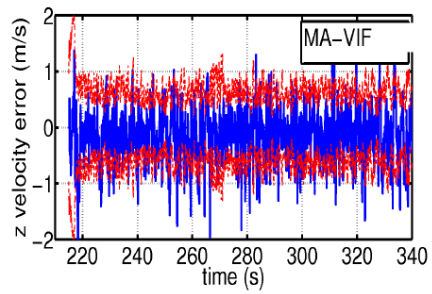
Experimental Results



(d)



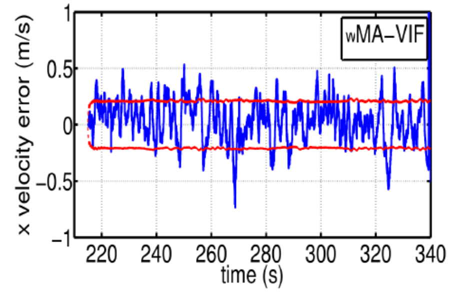
(e)



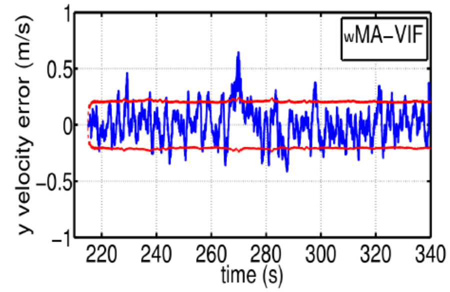
(f)

Without modelling the effects of wind

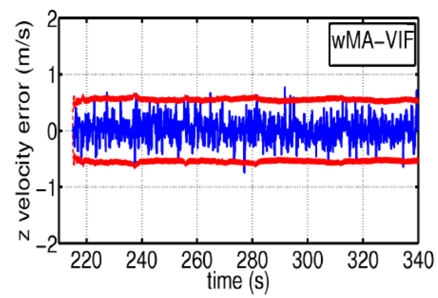
Velocity estimation errors and 2*sigma bounds



(a)



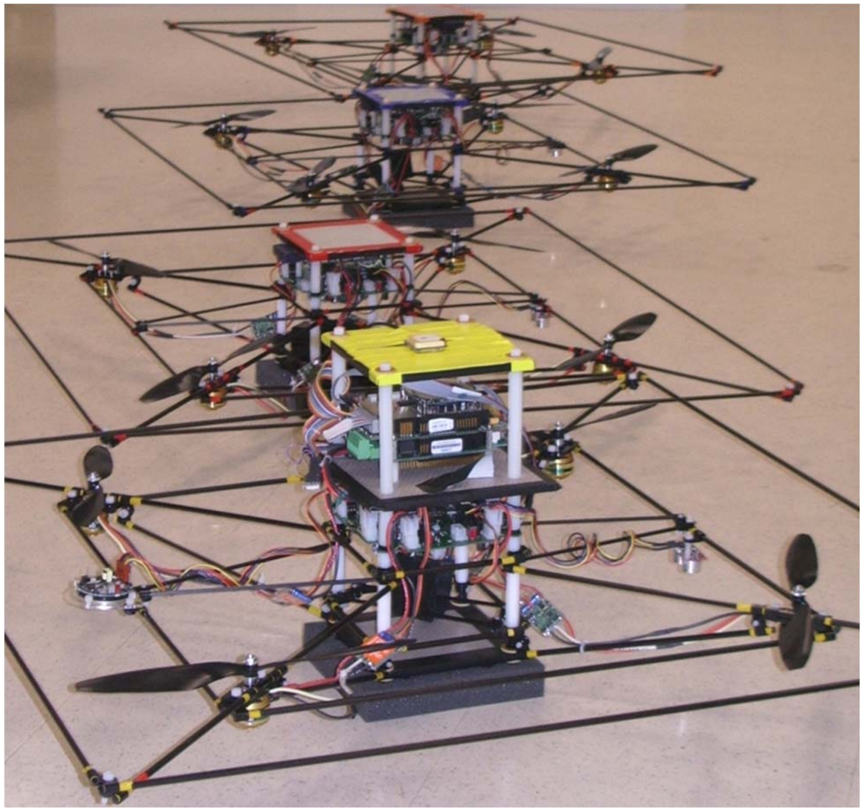
(b)



With effects of wind modelled

Outline

- Introduction
- Platform Development
- Vehicle modeling
- Estimation and Control
- Mapping
- Motion Planning
- Multi-Vehicle Coordination



2D-Laser-Based Mapping on Quadrotors

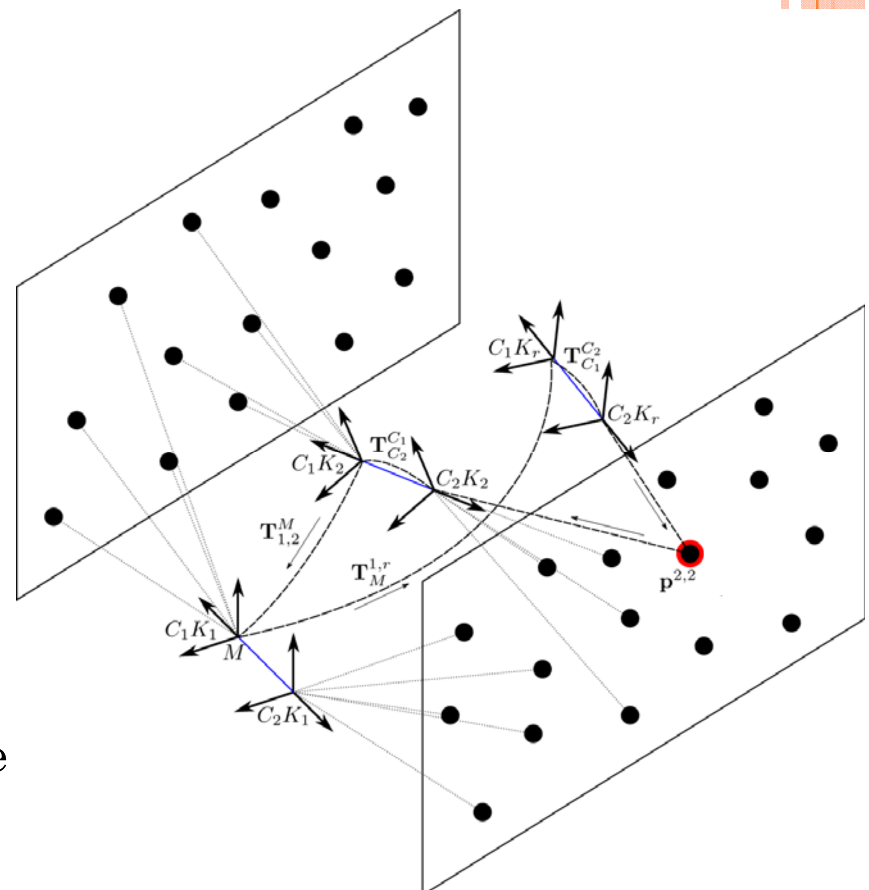
Multi-Camera Localization (and Mapping)

- Multi-camera parallel tracking and mapping (MCPTAM) (McGill – Harmat, Sharf)
  - Built on PTAM base
  - Omni-directional camera model
  - Tracking onboard X8 Atom Board
  - Mapping on the ground

- Multi-camera cluster relative pose estimation (Waterloo – Tribou, Wang, Waslander)
  - Built on EKF or bundle adjustment
  - Inverse depth, spherical parameterizations for seamless initialization
  - Overlap not necessary for global scale recovery
  - No motion models assumed (relative motion)
- Available at https://github.com/aharmat/mcptam

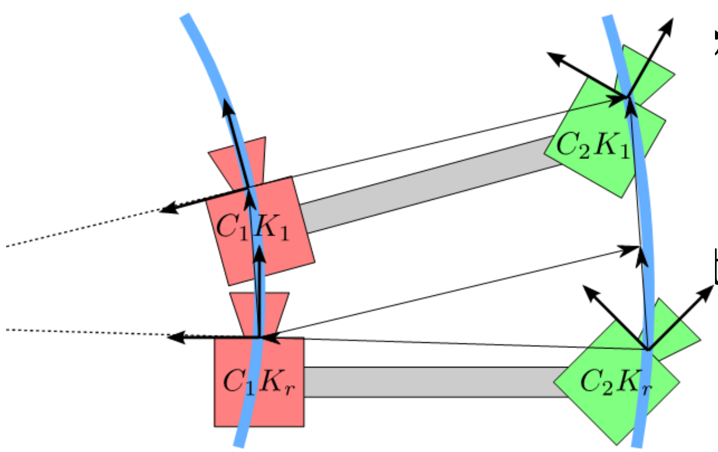
Multi-Camera Localization (and Mapping)

- Initialization
  - Without overlap, solve map up to scale
  - With overlap, solve map with scale
  - Store keyframe cluster in map
- Tracking
  - Fixed map = fast computation
  - Strong-ish feature correspondence from PTAM (windowed, tracked features)
- Mapping
  - Keyframe selected based on distance
  - First keyframe a feature is observed in defines feature coordinate frame
  - Backend optimization using g2o resolves scale



Multi-Camera Localization (and Mapping)

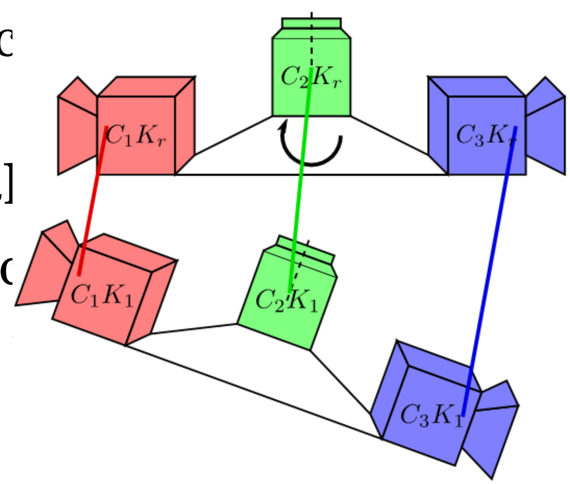
- Degeneracy analysis for two and three cameras
  - Identify conditions for which measurement Jacobian is degenerate
  - Can be reduced to a 6X6 matrix rank check
- Examples of degenerate motions that could occur on UAVs:



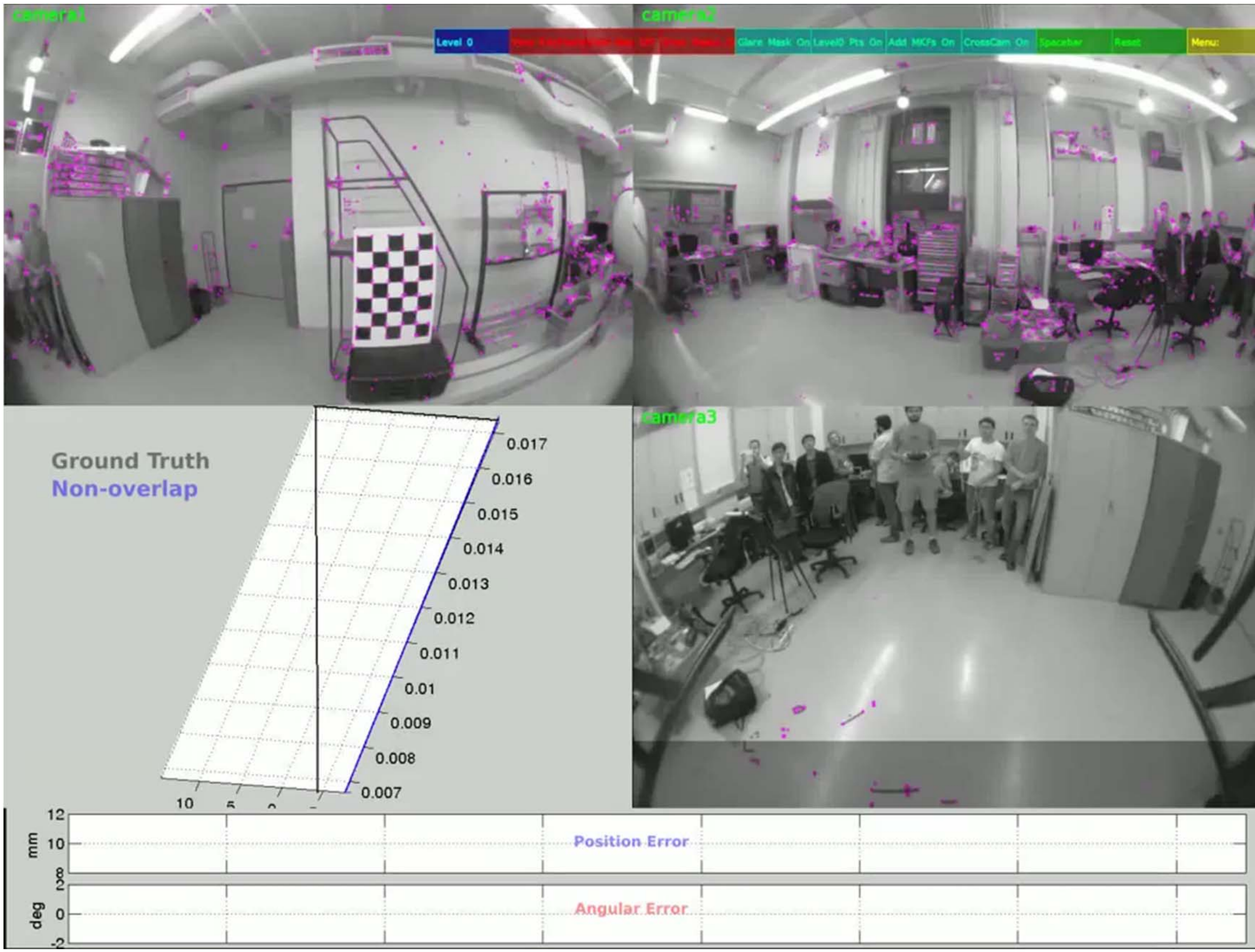
: circ traight

in tl ıtres

th ac



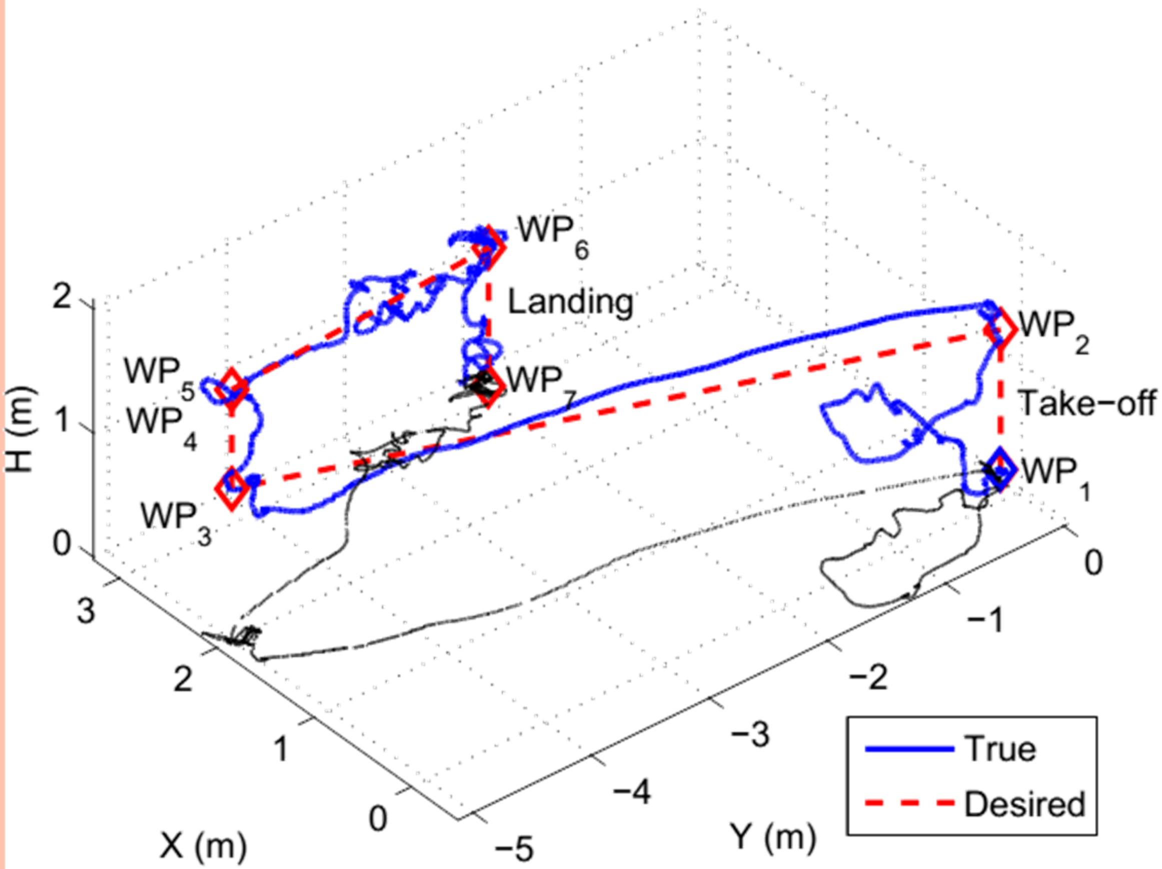
Multi-Camera Localization (and Mapping)

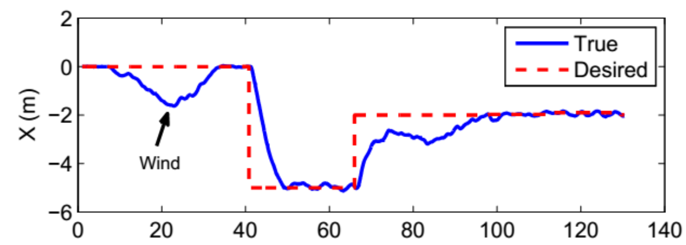


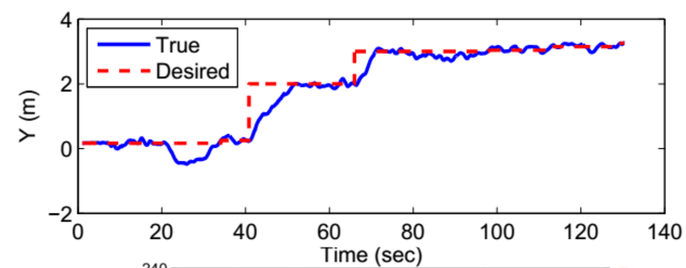
MCPTAM In Flight

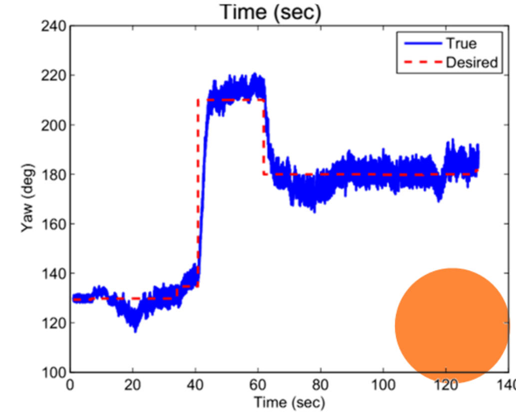
Multi-Camera Localization (and Mapping)

- Closed loop outdoor control using MCPTAM on Draganflyer X8



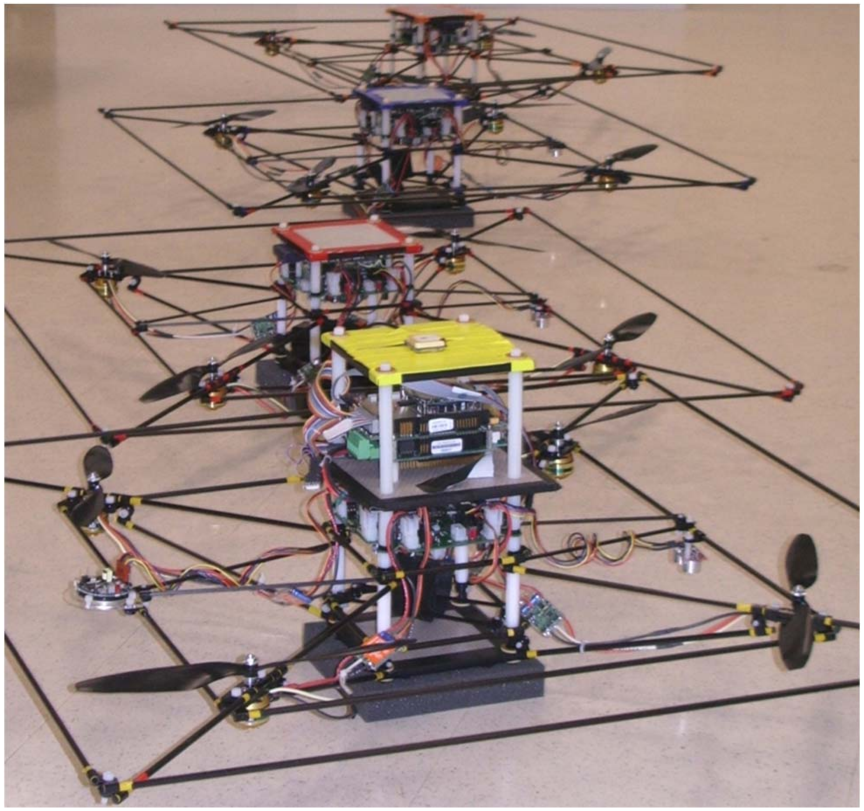






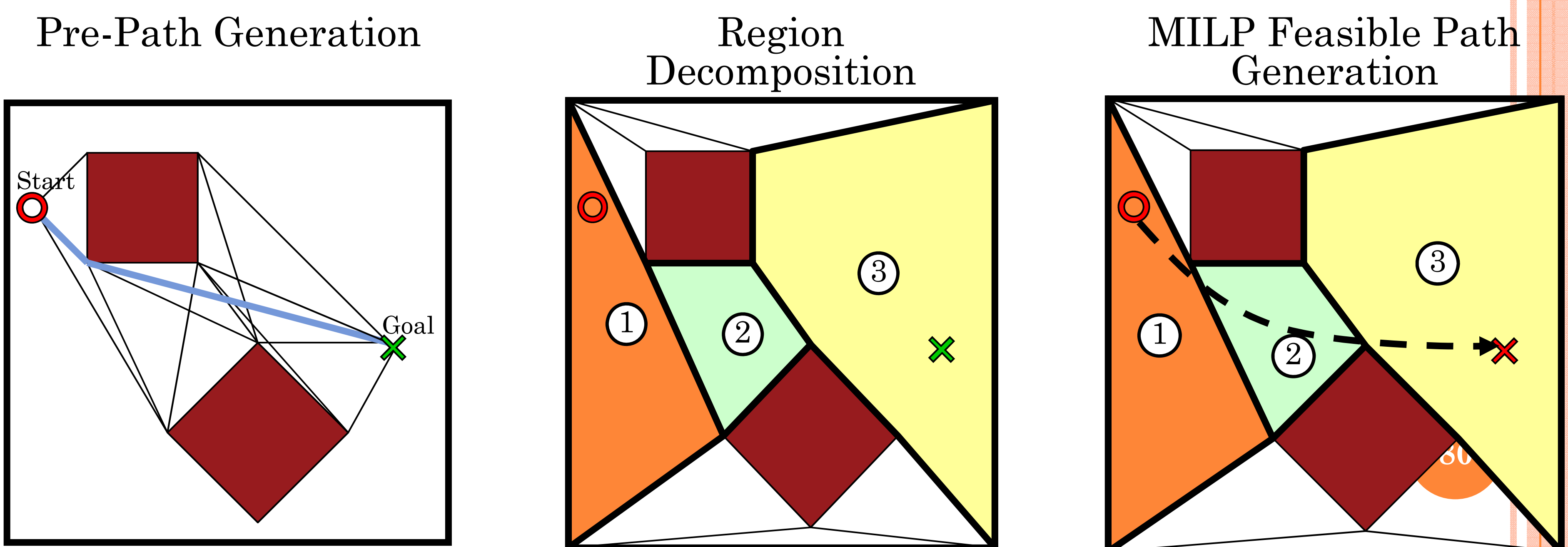
Outline

- Introduction
- Platform Development
- Vehicle modeling
- Estimation and Control
- Mapping
- Motion Planning
  - Tunnel-MILP
  - PRM/NLP
  - Reachable Sets
- Multi-Vehicle Coordination



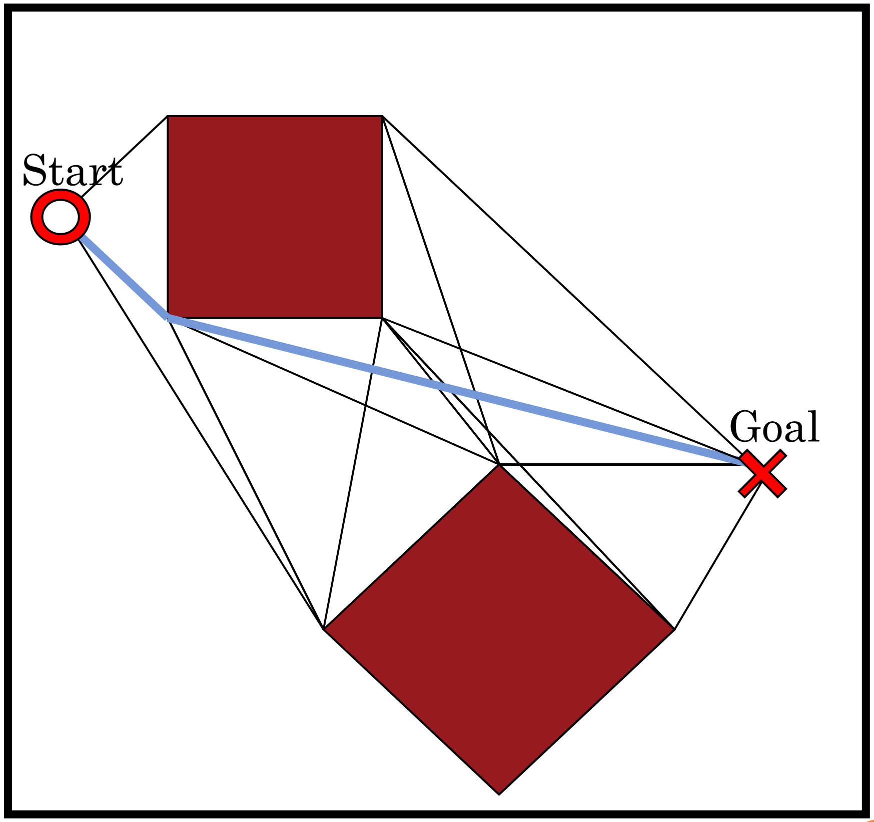
TUNNEL-MILP ALGORITHM

1) Pre-path determination ignoring the dynamics of the vehicle.
2) Region decomposition into convex polygons and identify tunnel.
3) Solve the dynamically feasible optimal control problem as a MILP through the tunnel.



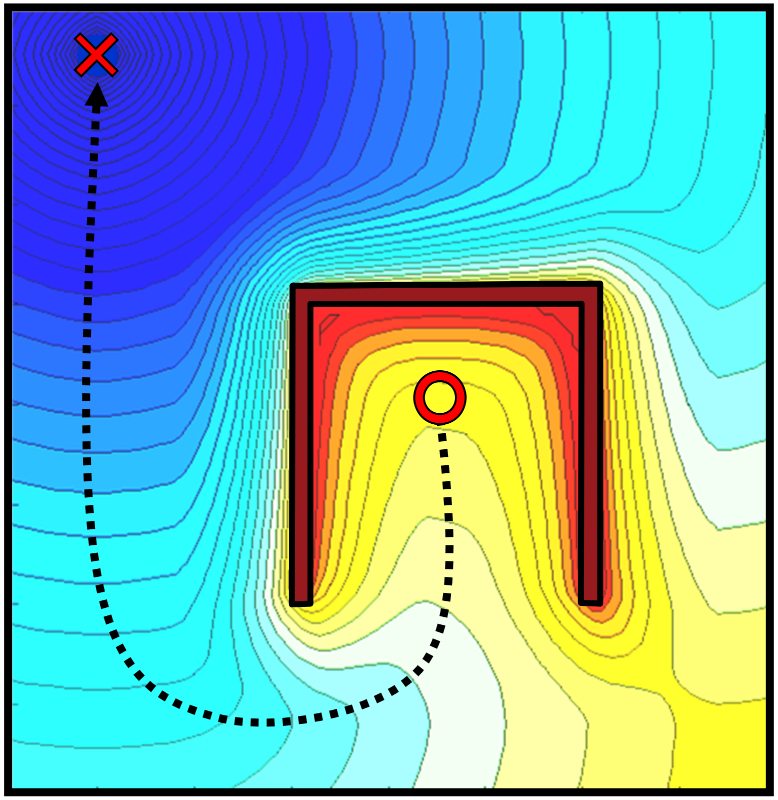
1) Pre-Path Generation: Visibility Graphs

- Standard method
  - Fully connect all obstacle vertices, start and finish
  - Discard any paths that traverse obstacles
  - A* search for shortest path
- Finds shortest path
  - Relatively quick
  - Necessarily close to obstacles
  - Kinks in path challenging for vehicles



1) Pre-Path Generation: Fast Marching

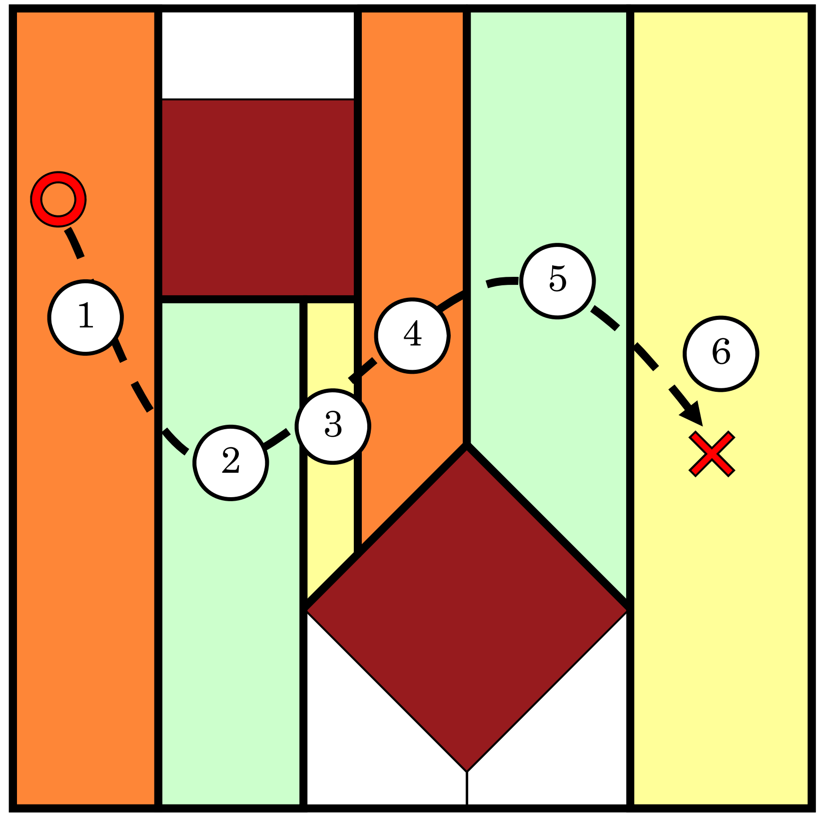
- Wave propagated from target
  - Speed of propagation varied based on proximity to obstacles
- Path determined by gradient descent of resulting potential field
- Smooth path results
  - No local minima
  - Tunable obstacle repulsion
  - Still not necessarily feasible



2) Region Decomposition: Trapezoidal

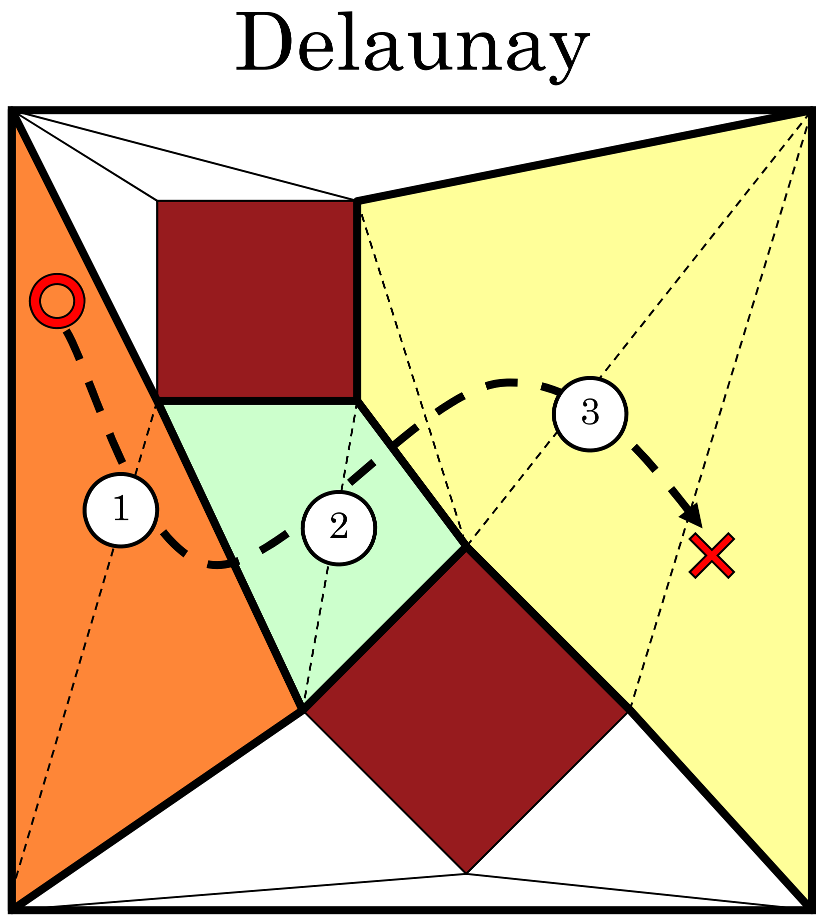
- Vertical cuts applied to each obstacle vertex
- Sequence of polytopes containing pre-path then identified

Trapezoidal



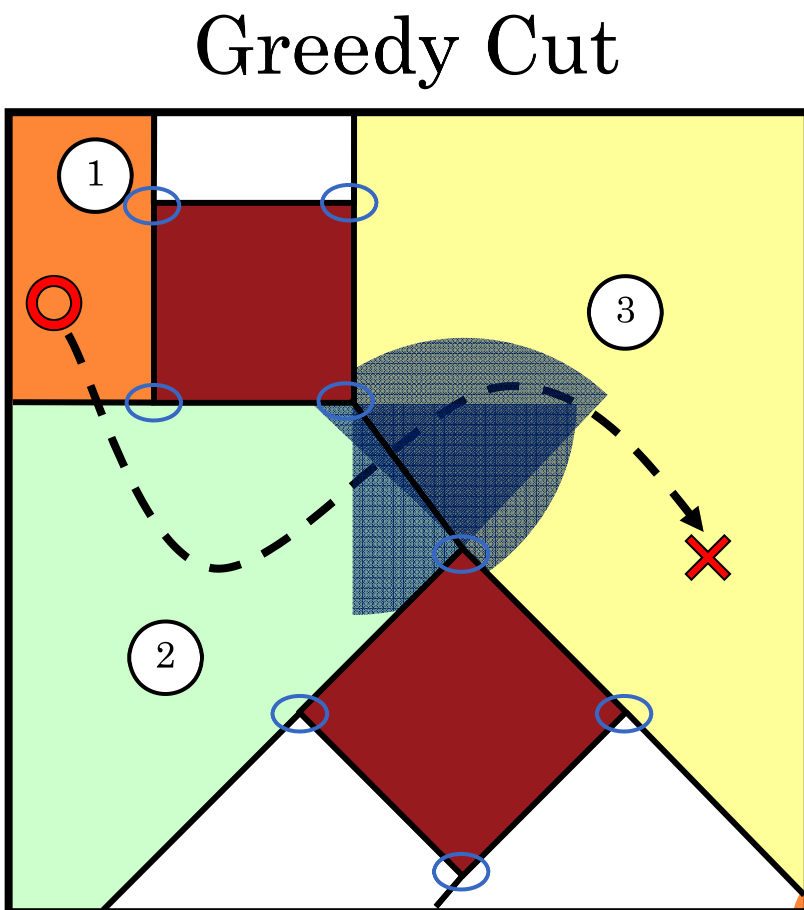
2) Region Decomposition: Delaunay

- Delaunay triangulation
  - Constrained to form triangles outside of obstacles
- Adjacent triangles along path combined if possible
  - Create larger convex regions



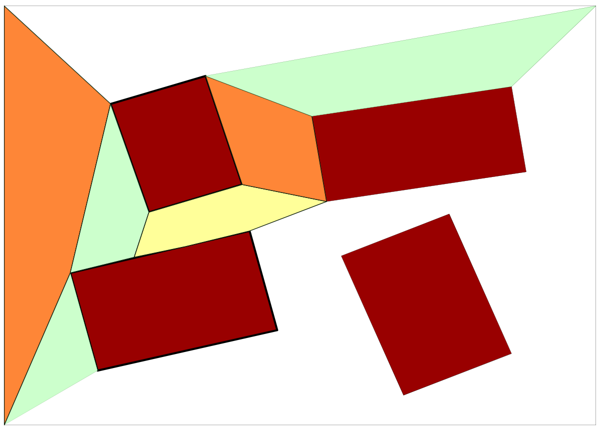
2) Region Decomposition: Greedy Cut

- Optimal Convex Decomposition for polytopes with holes is NP-Hard
- Each non-convex vertex requires at most one cut to eliminate it
- Any single cut can eliminate at most two non-convex vertices
- Greedy Cut Algorithm
  - Matching cuts
  - Vertex cuts

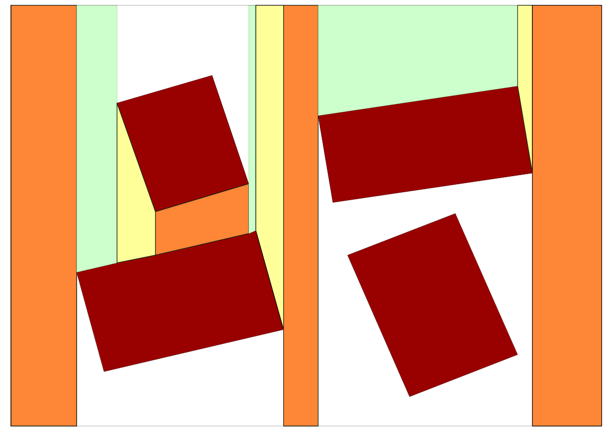


COMPARISON OF RESULTING TUNNELS

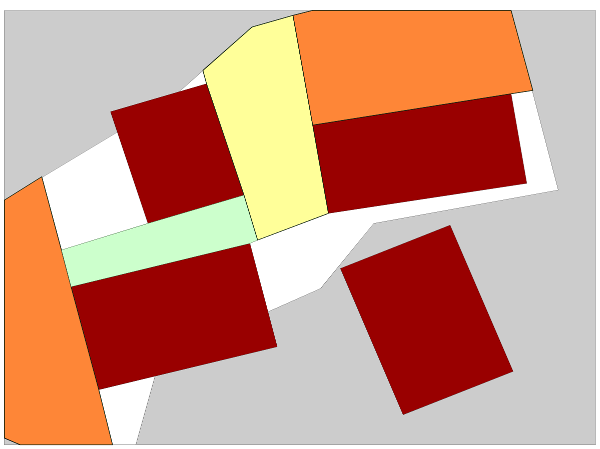
Delaunay



Trapezoidal



Greedy Cut



	Avg. Number of Regions	
	4 Obstacles	8 Obstacles
Greedy Cut	4.4	6.0
Delaunay	6.3	7.5
Trapezoidal	10.5	15.9

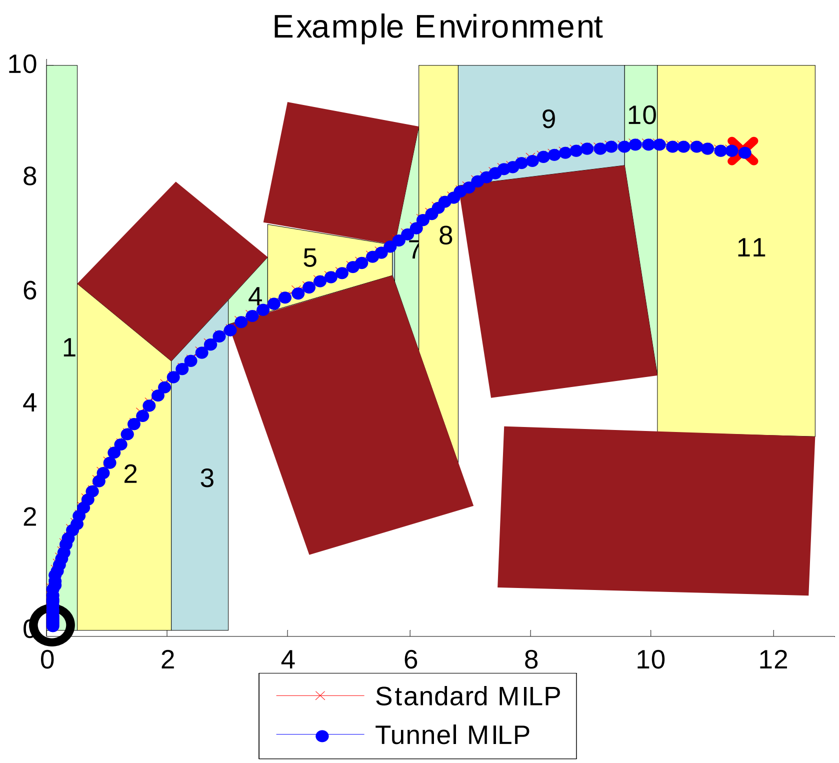
3) MILP Formulation

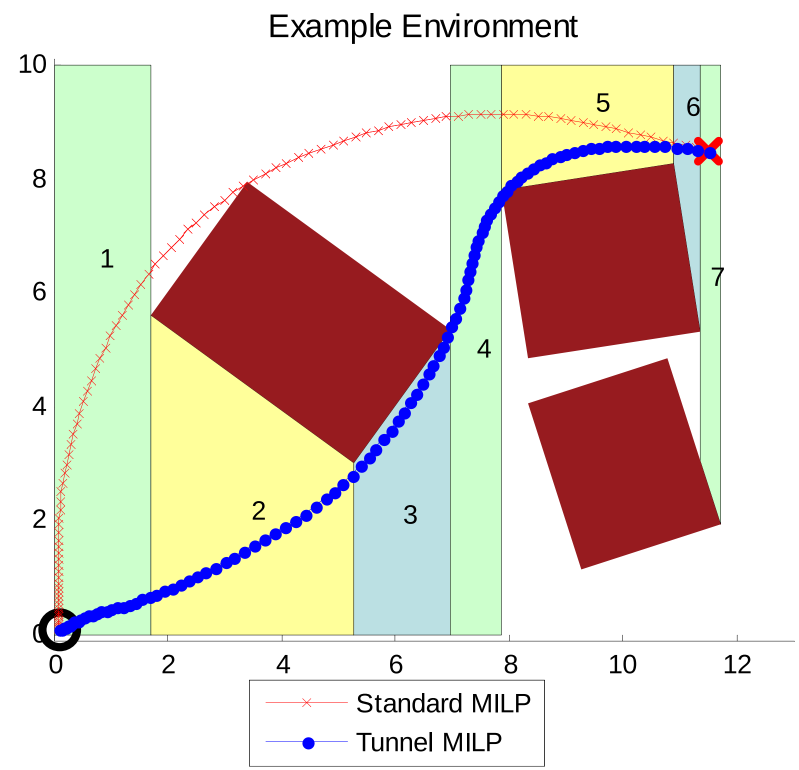
- Require polygons to be traversed in order

$$
\begin{aligned}
\text{minimize} \quad & \gamma t_f + (1-\gamma)||u||_1 \\
\text{subject to} \quad & \begin{bmatrix} x(t+1) \\ v(t+1) \end{bmatrix} = A \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + Bu(t) \\
& \underline{v} \leq v_i(t) \leq \bar{v}\,, i \in \{1,2\},\ \forall t \in \{1,\ldots,T\} \\
& \underline{u} \leq u_i(t) \leq \bar{u}\,, i \in \{1,2\},\ \forall t \in \{1,\ldots,T\} \\
& x(t_f) = x_{goal} \\
& \textstyle\bigwedge_{i\in\{1,\ldots,N\}} F_i x - b_i \leq 0 \\
& t_{i+1}(0) > t_i(f)
\end{aligned}
$$

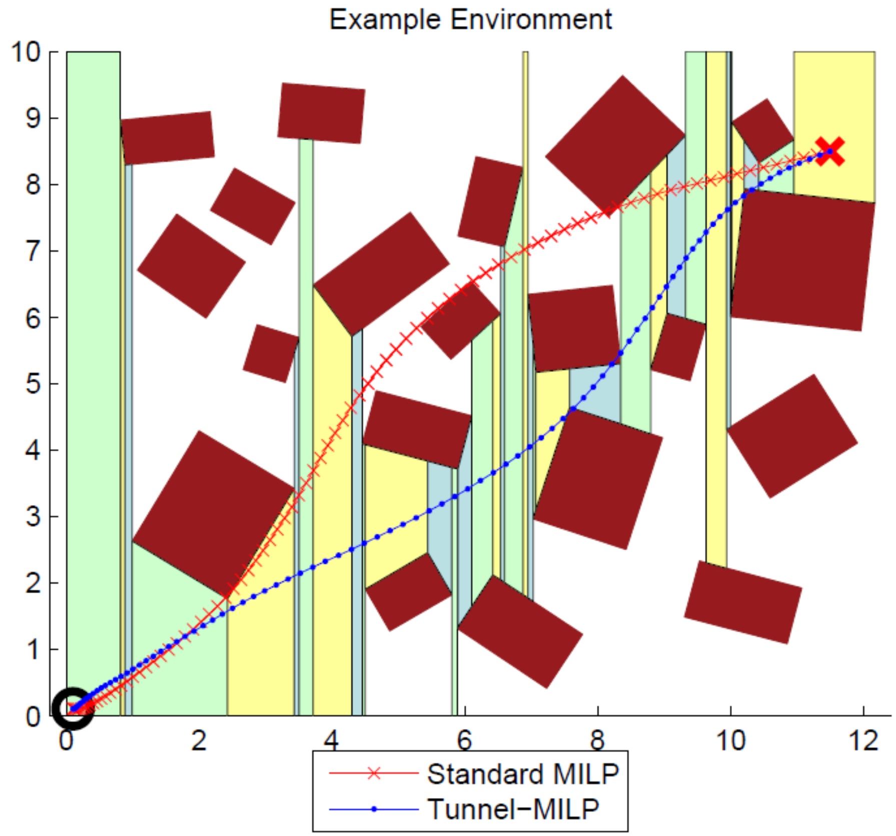
- CPLEX MILP Tricks
  - Indicator constraints (logical)
  - Feasible solution first

RESULTS: EXAMPLE SOLUTIONS



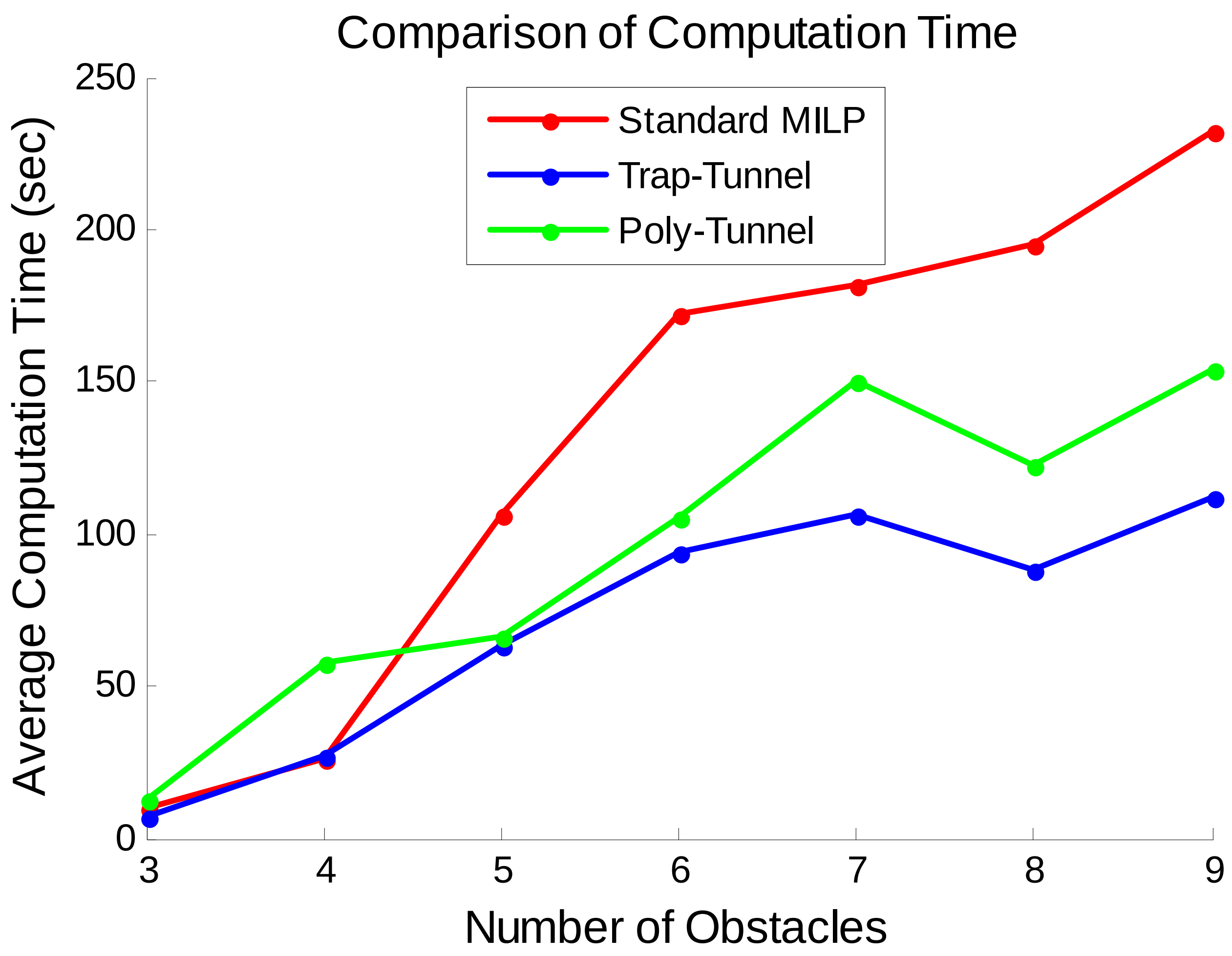


Results: Example solutions



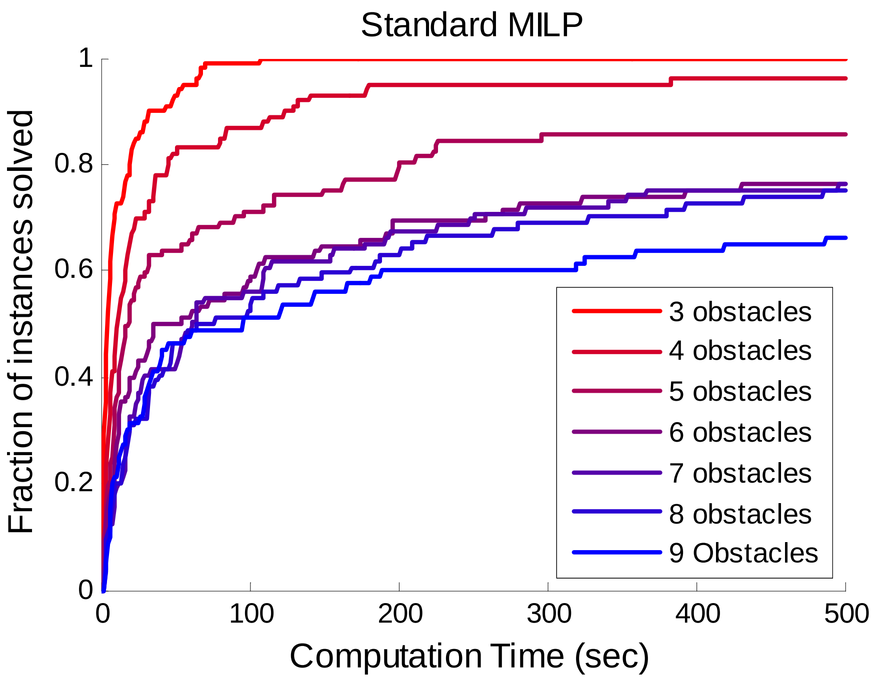
# Obstacles	% Avg. Timestep Increase	% Avg. Input Cost Increase
3	7.78	14.94
4	3.69	13.96
5	2.01	8.06
6	1.77	3.69
7	2.55	9.23
8	1.69	6.61
9	1.94	4.94
20	0.73	3.82

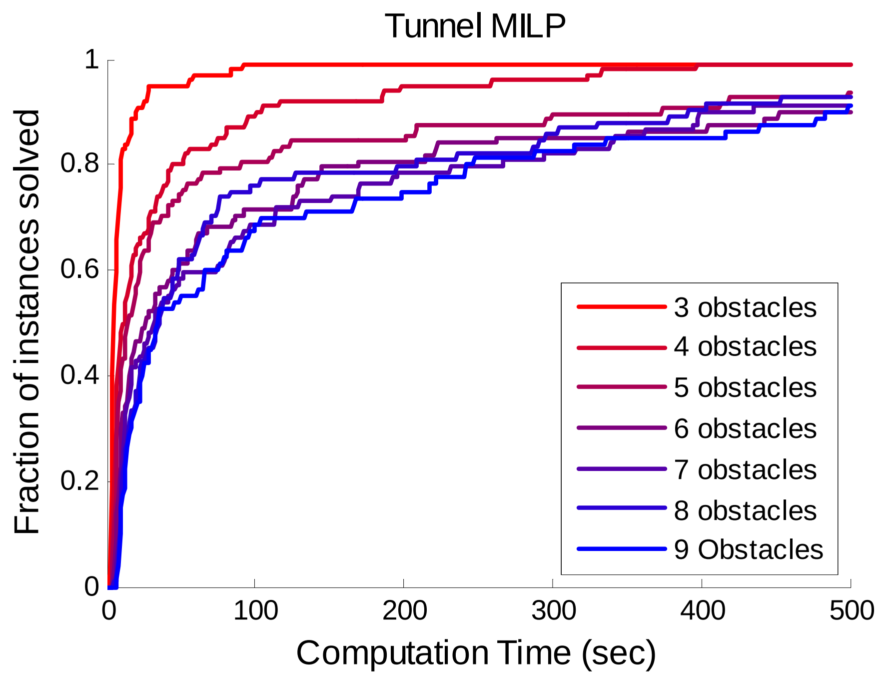
Results: Comparison of Computation



RESULTS: COMPUTATION TIMES

- Maximum 600 seconds of computation allowed
  - Yes we ran many computers for days

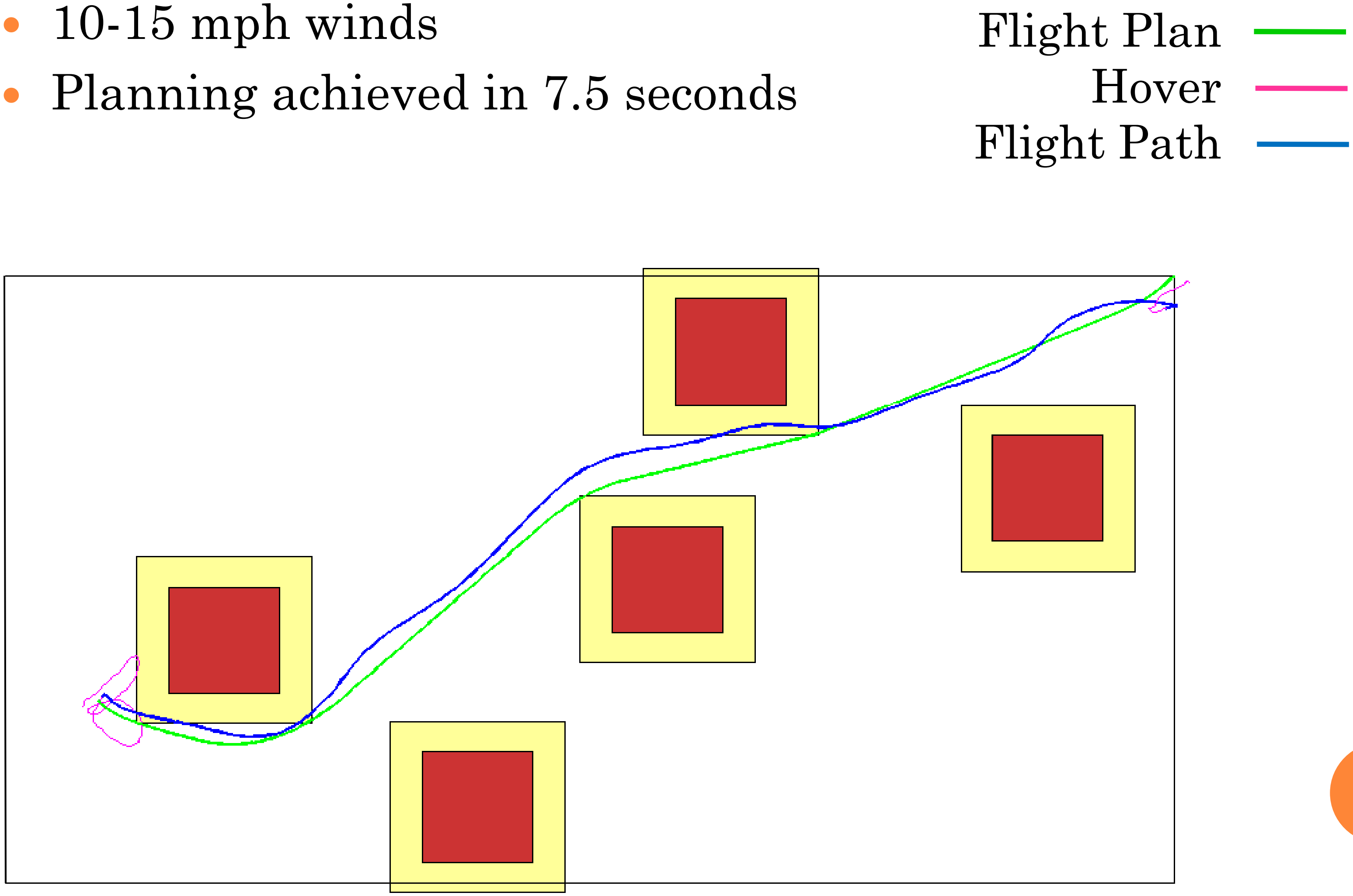




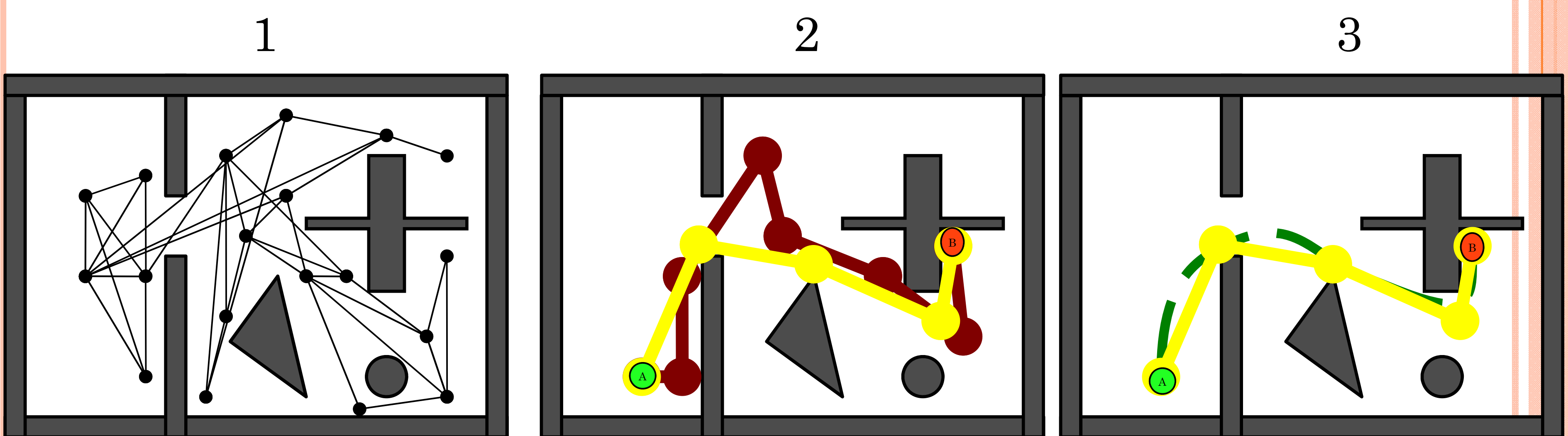
Tunnel-MILP on STARMAC

TUNNEL-MILP ON STARMAC

- Outdoor flight test results
  - 10-15 mph winds
  - Planning achieved in 7.5 seconds



Randomized Exploration Optimal Trajectory Planning

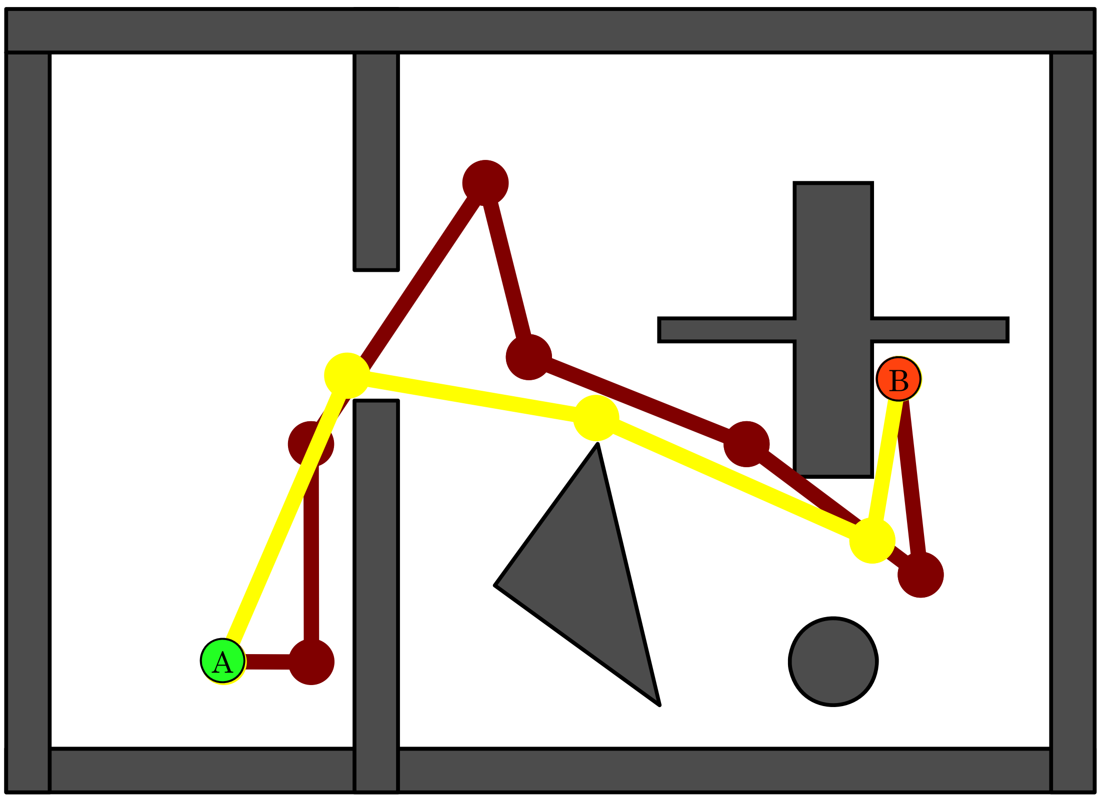


Generate PRM

Query PRM and Improve Path

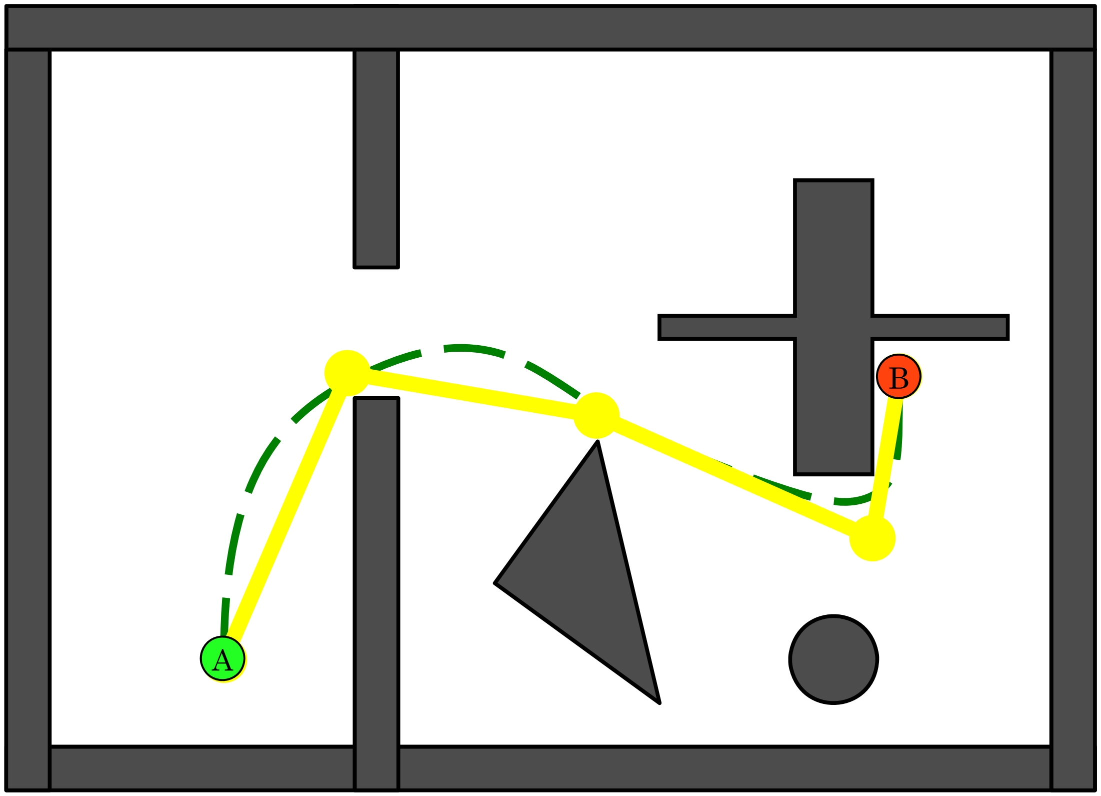
Find Dynamically Feasible Trajectory

PRM Path Improvement



- Origin A, Destination B
- Attach A and B to roadmap (same method)
- Use discrete planning techniques (Dijkstra, A*, etc) to find shortest path from A to B
- **Refinement: Check for better lines of sight along the original shortest path**

Step 3: Formulate and Solve NLP



- Path from PRM query is:
- Not dynamically feasible
- Not optimal
- **<u>But</u>**, provides good initial guess for NLP solver

Step 3: NLP Formulation

$$\text{minimize } f(x)$$

subj. to:

$$c(x) \leq 0$$

$$h(x) = 0$$

$$x_{min} \leq x \leq x_{max}$$

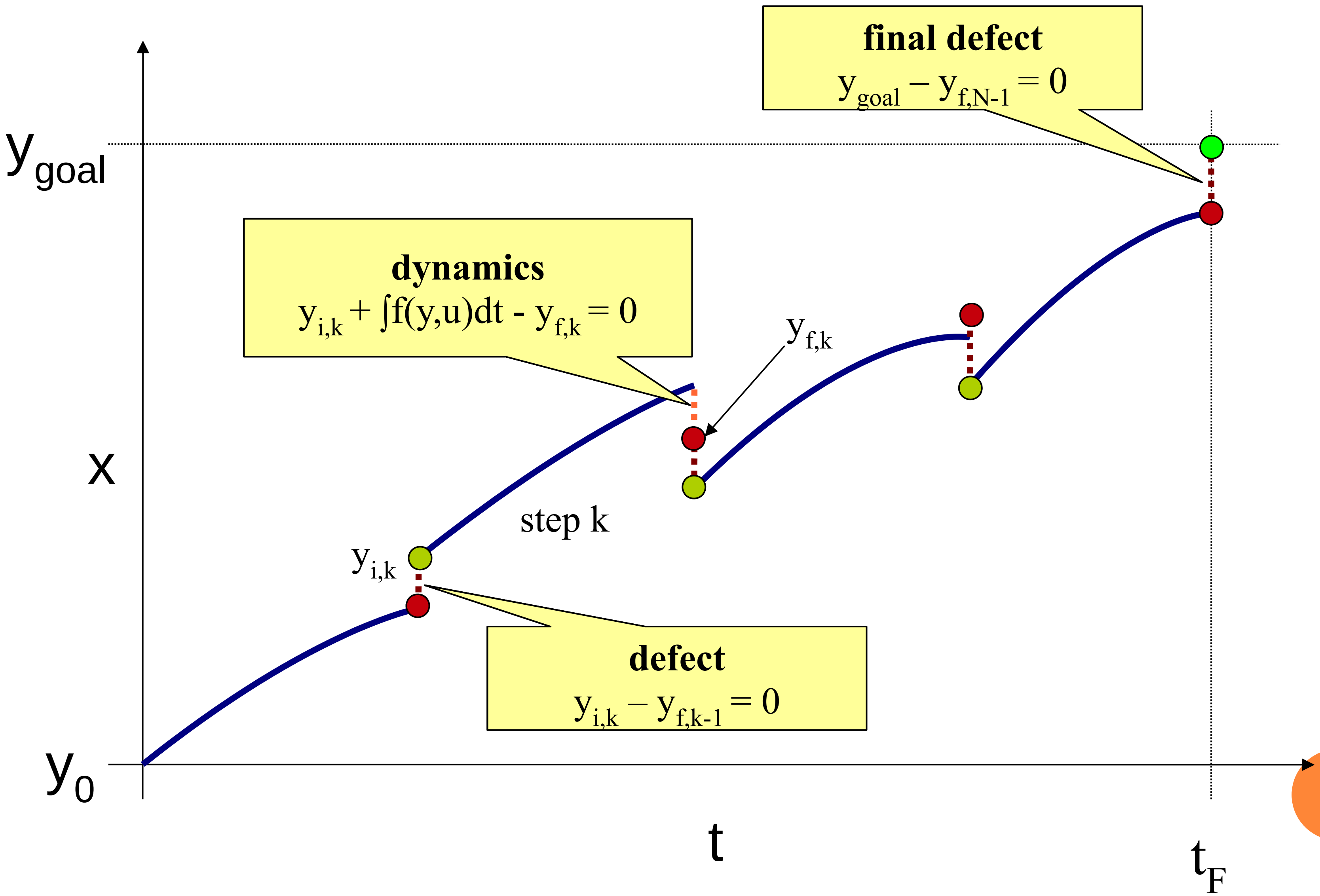
Control cost

Collision constraints

- Dynamics
- State consistency
- Goal

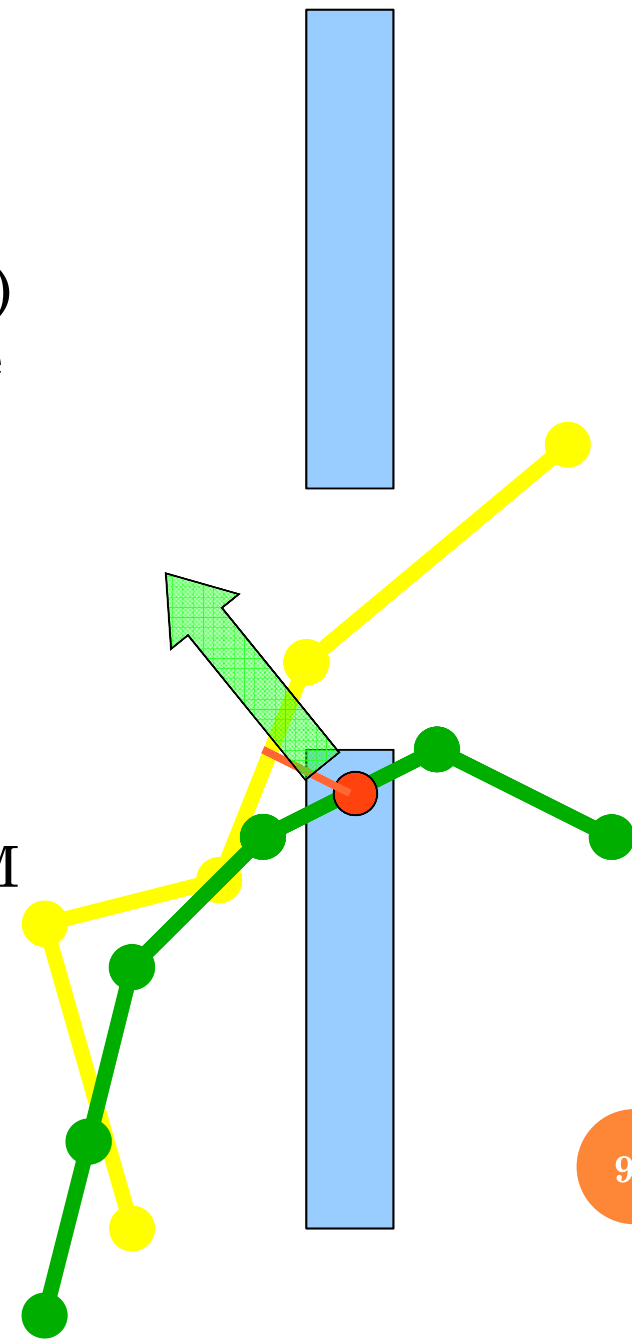
Simple bounds on state and control variables

Multiple Shooting Formulation



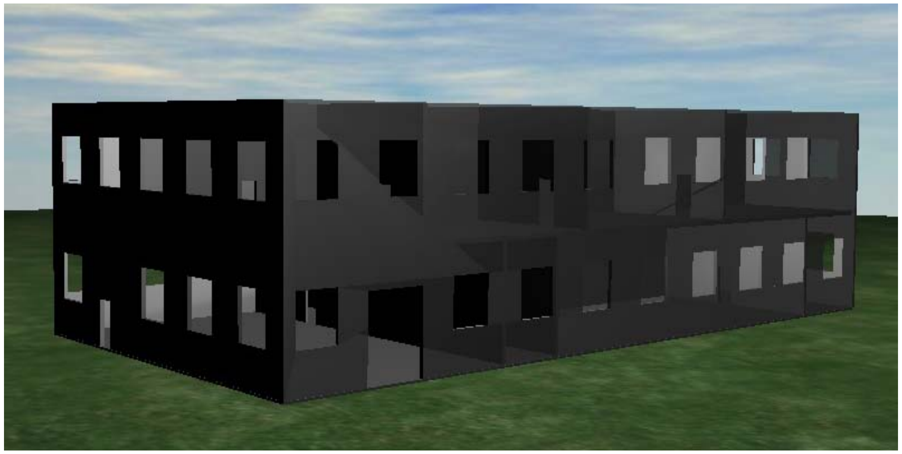
Collision Constraint

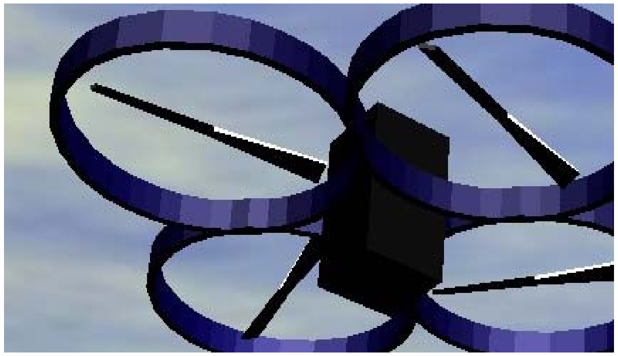
- Proximity Query Package (PQP) used to check distance from line segments connecting initial and final points to obstacles
- PQP returns 0 for collisions
  - No depth-of-penetration info
  - Use approximate heuristic:
  - Take distance from midpoint of colliding line segment to the PRM path
  - Issues
    - Discontinuity of c(x)
    - Pathological cases



Scenario

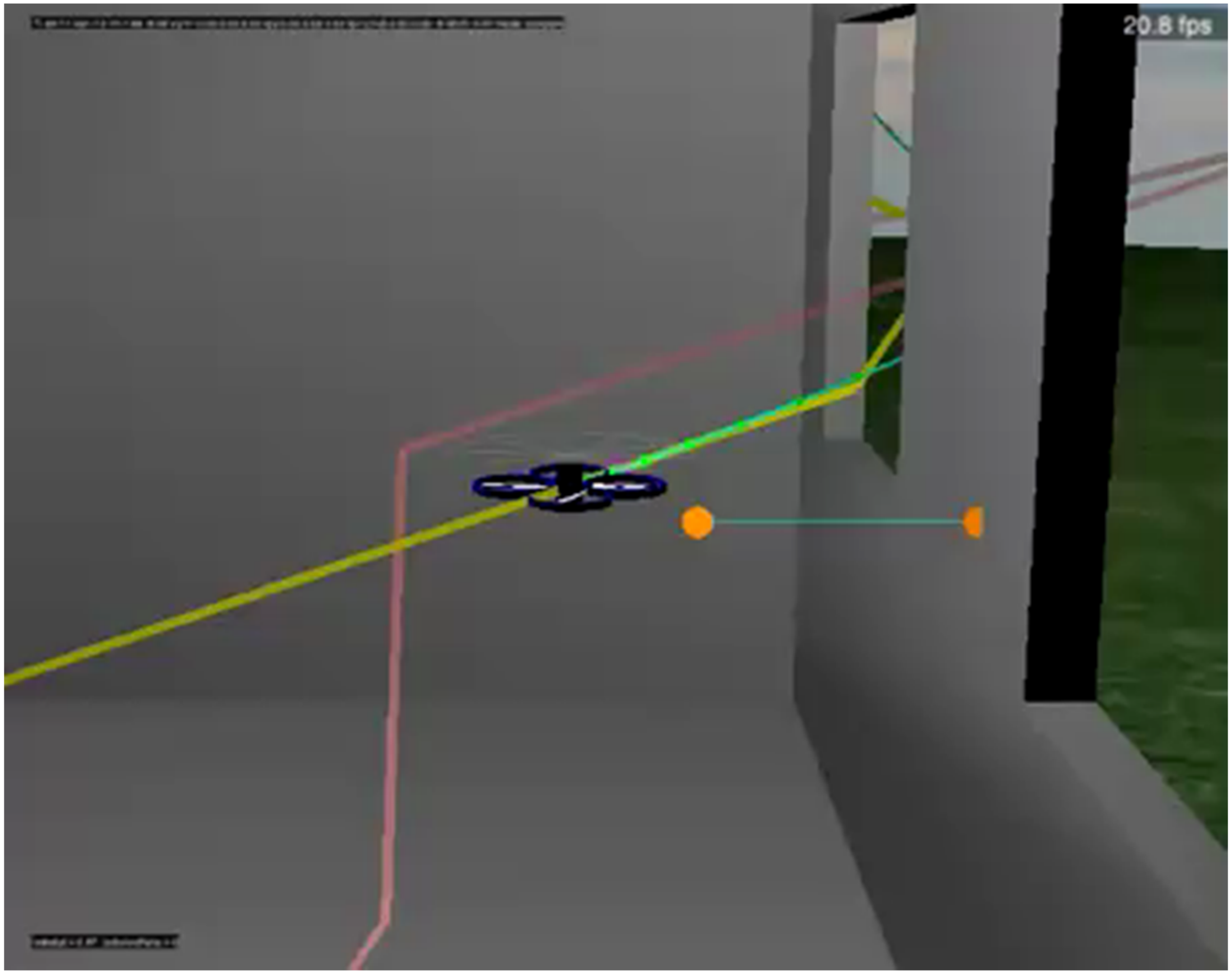
- Building
  - 20 m x 40 m x 10 m
  - 2,402 triangles
  - 2 floors
  - doors, windows
  - 12 rooms, 2 corridors





- Quadrotor
  - ~ 80 cm wide
  - 1.9 kg

EXAMPLE PRM IN 3D

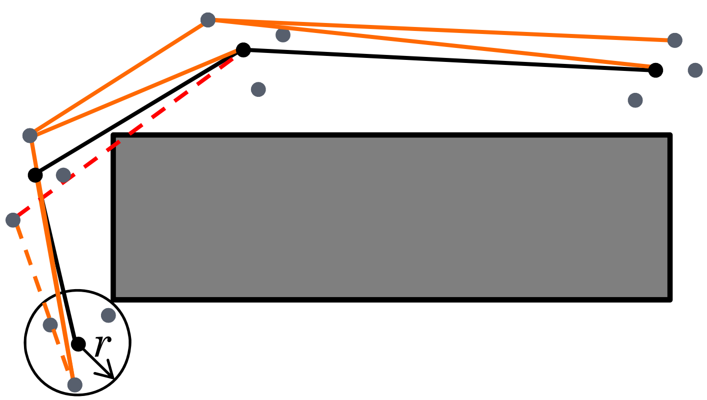


Heuristic Re-sampling Algorithm

- Maximize heuristic function using iterative greedy approach

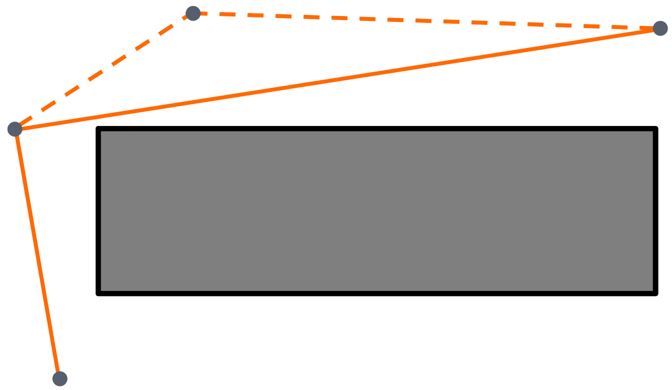
$$\frac{1}{N}\sum_{i=0}^{N} \min(\min\_dist(node\ i, O), d_{max})$$

- Step 1:



Heuristic Re-sampling Algorithm

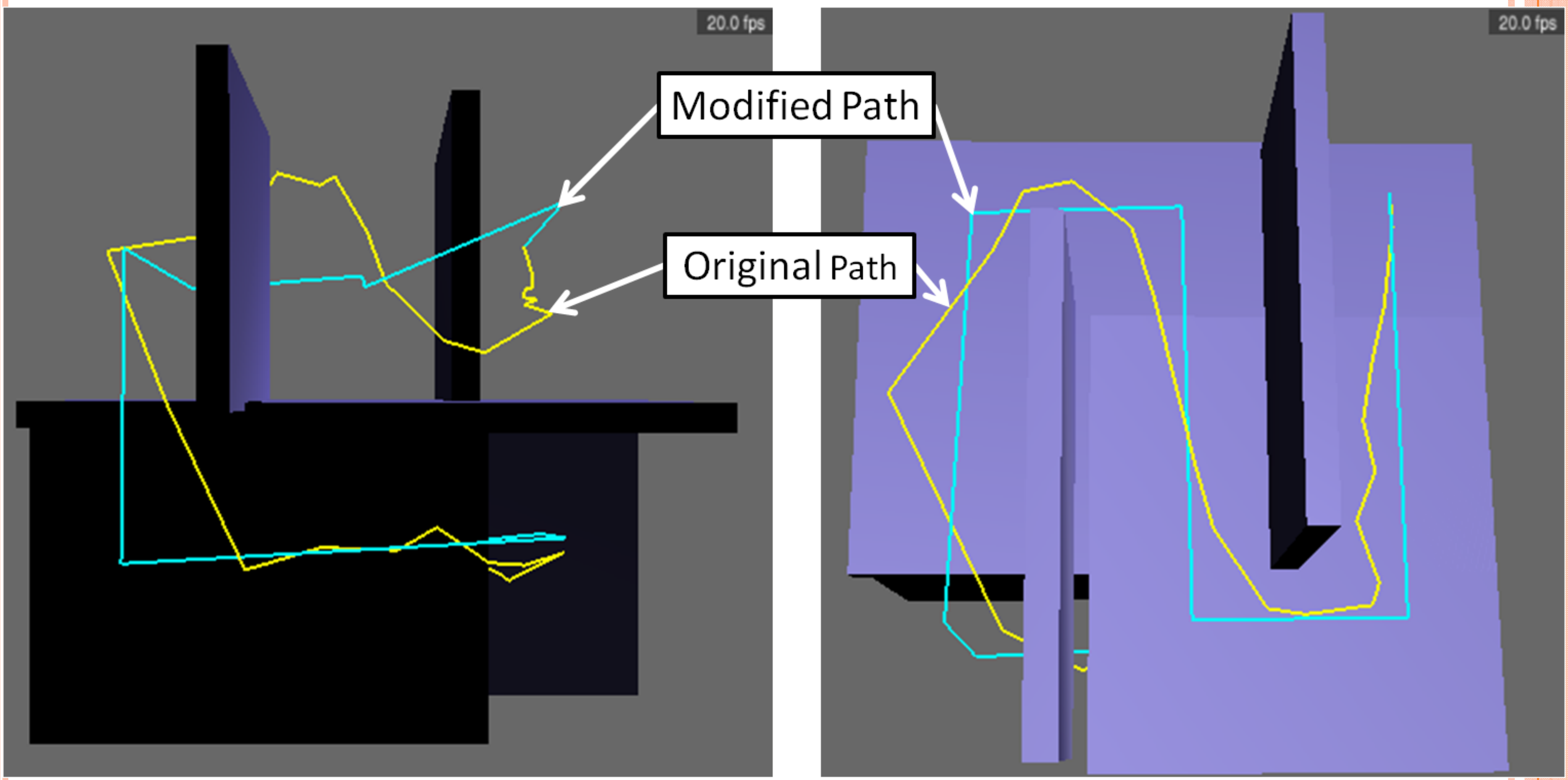
- Step 2:



- Repeat steps 1 and 2 until improvements in heuristic value is less than ε for several iterations

HEURISTIC RE-SAMPLING EXAMPLE

- Runtime = 0.68 seconds, total iterations = 4



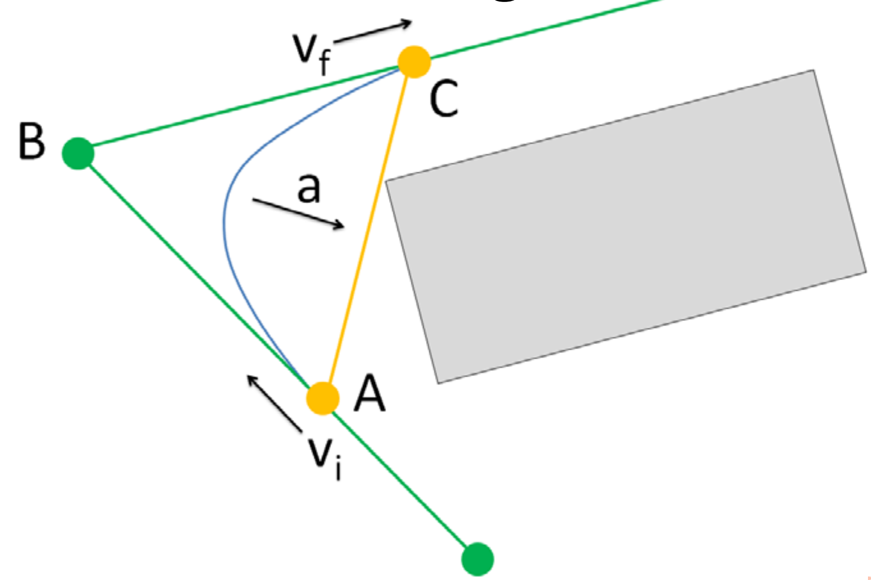
Side view

Top view

DEFINING MOTION PRIMITIVES

- Find A and C such that triangle ABC is as large as possible and obstacle free
- Worst case scenario: AB = BC = 0
  - Use trivial solution
- Straight segments are collision free (PRM)
- Goal: contain corner motion inside triangle ABC

1. $\hat{v}_i \cdot \widehat{AB} = \hat{v}_f \cdot \widehat{BC} = 1$
2. $\dfrac{\|v_f\|_2}{\|v_i\|_2} = \dfrac{\|BC\|_2}{\|AB\|_2}$



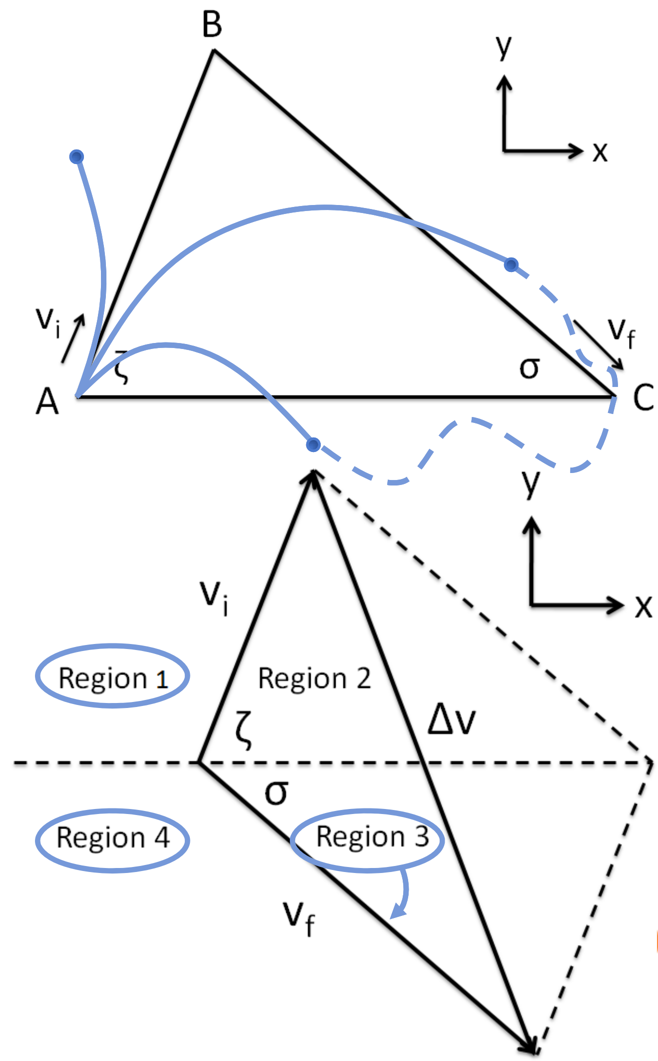
Corner Motion: Feasible and Bounded

- There exists an unique acceleration such that
  - Position: A → C
  - Velocity: v_i → v_f

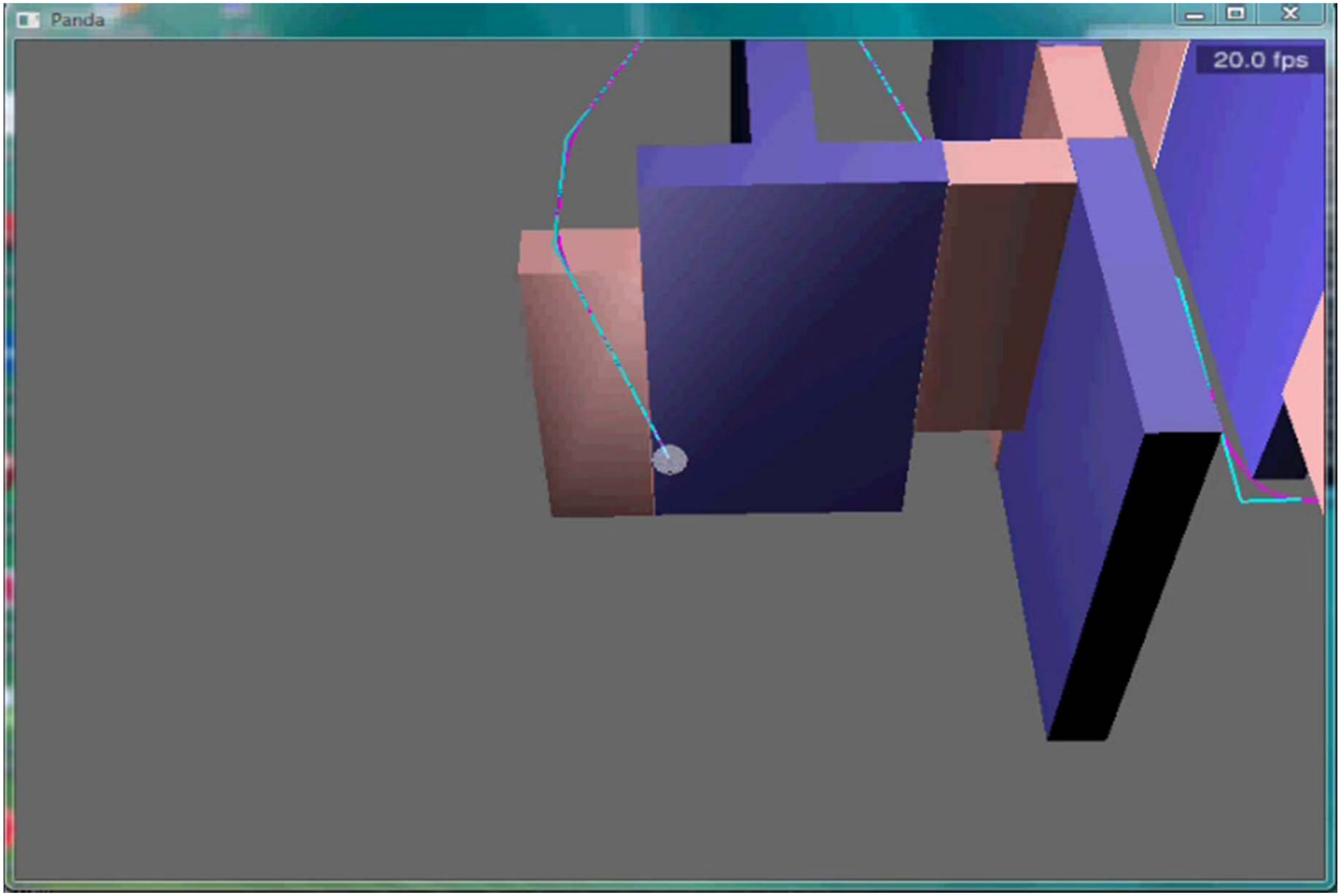
$$\Delta p_x = \frac{1}{2}(v_f \cos\sigma + v_i \cos\zeta)t = \frac{K}{a}$$

$$\Delta p_y = \frac{1}{2}(v_f \sin\sigma + v_i \sin\zeta)t = 0$$

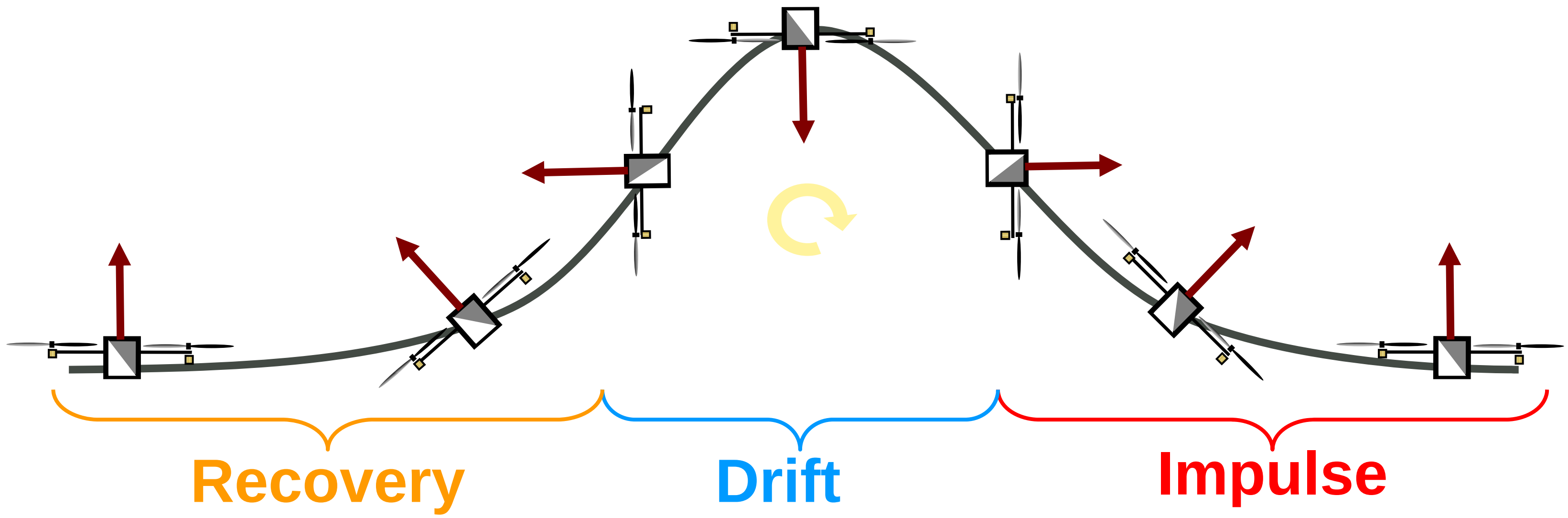
- Velocity direction dictates gradient of the path
- Direction of velocity
  - Region 2 → Region 3



Simulation Example



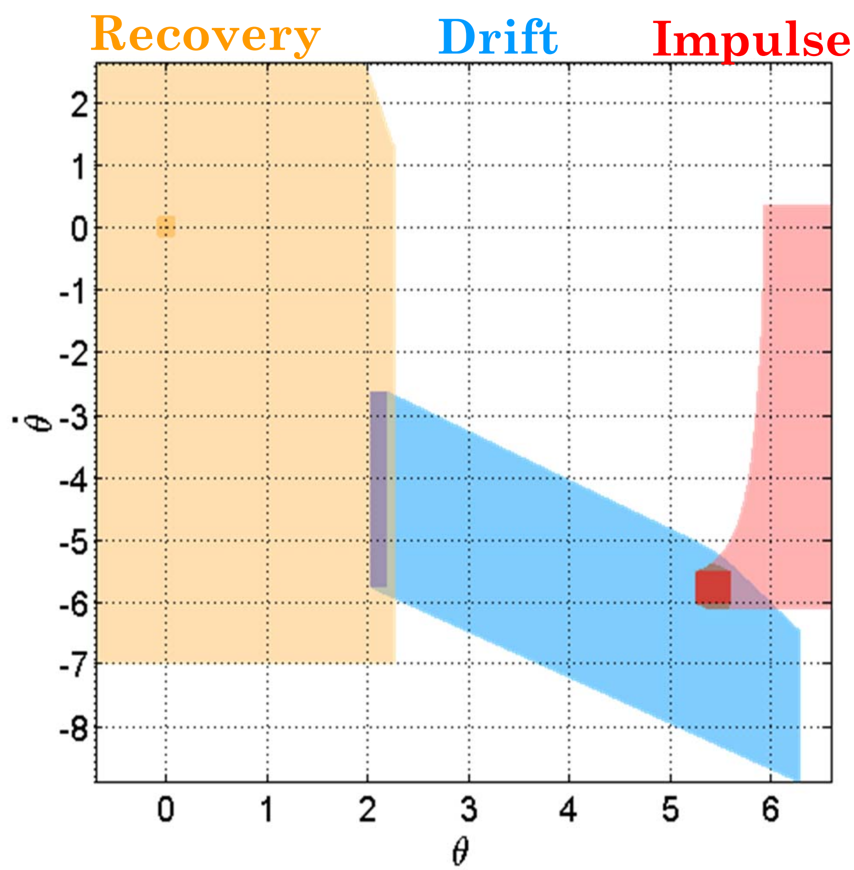
Motion Planning: Quadrotor Back-Flip



- Divide flip into three modes
- Difficult problem:
  - Hitting some target sets while avoiding some unsafe sets
- Solution:
  - Analyze rotational dynamics and vertical dynamics separately

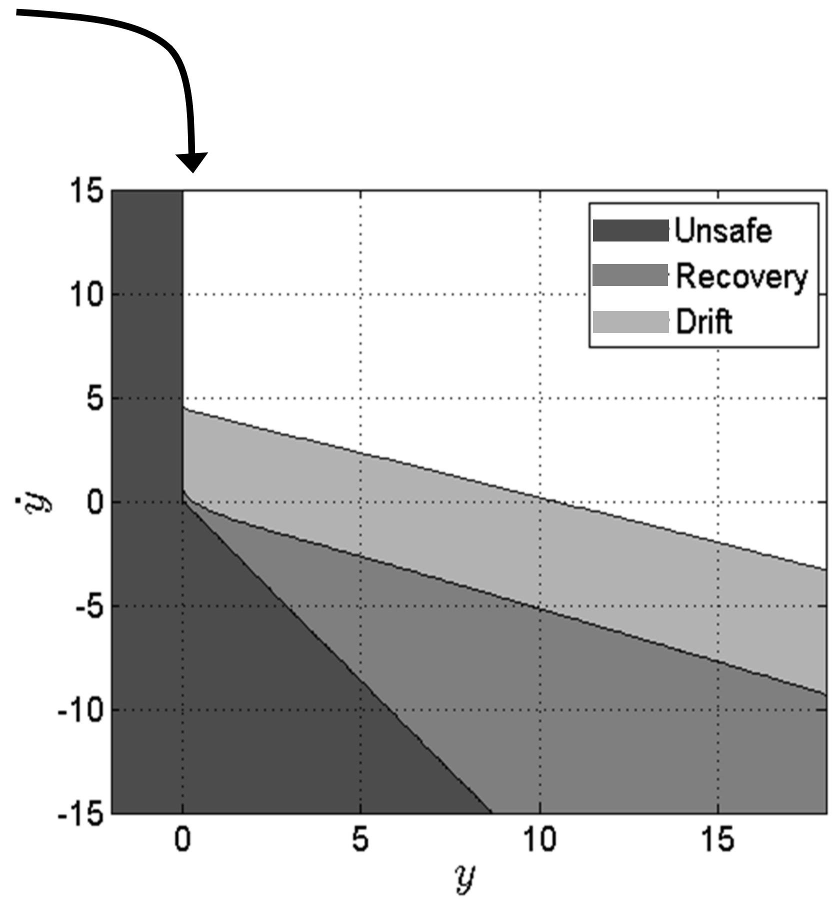
BACK-FLIP: METHOD (1)

- Identify ***target*** region in rotational state space for each mode
- Use reachable sets to calculate ***capture basin*** for each target
  - Dynamic game formulation accounts for worst-case disturbances
- Verify that target of each mode is contained by capture basin of next mode

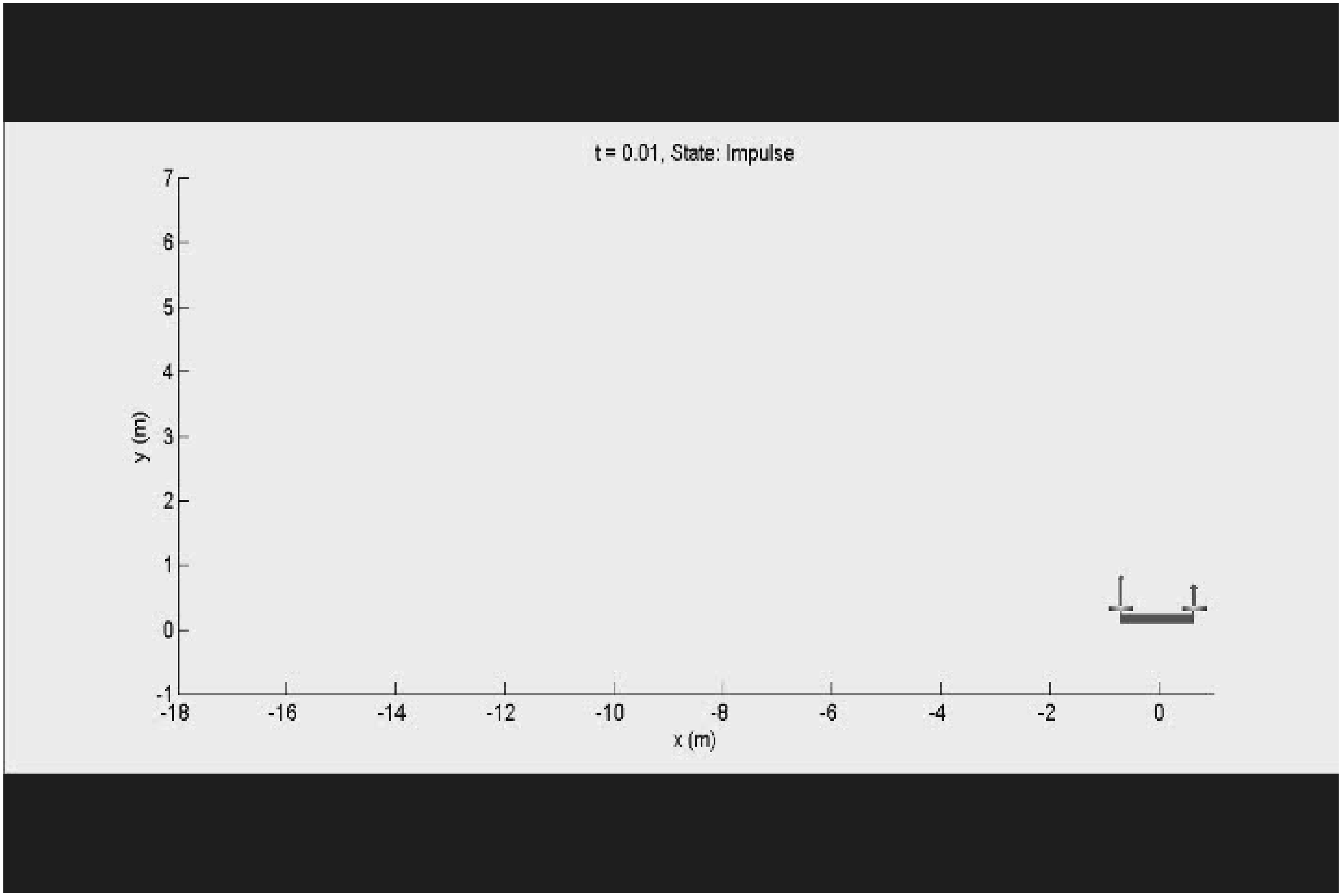


Back-flip: Method (2)

- Identify ***unsafe*** region in vertical state space for final mode
- Use reachable sets to propagate ***unsafe set*** for each mode
  - Dynamic game formulation accounts for worst-case disturbances
- Verify that control keeps state out of unsafe set



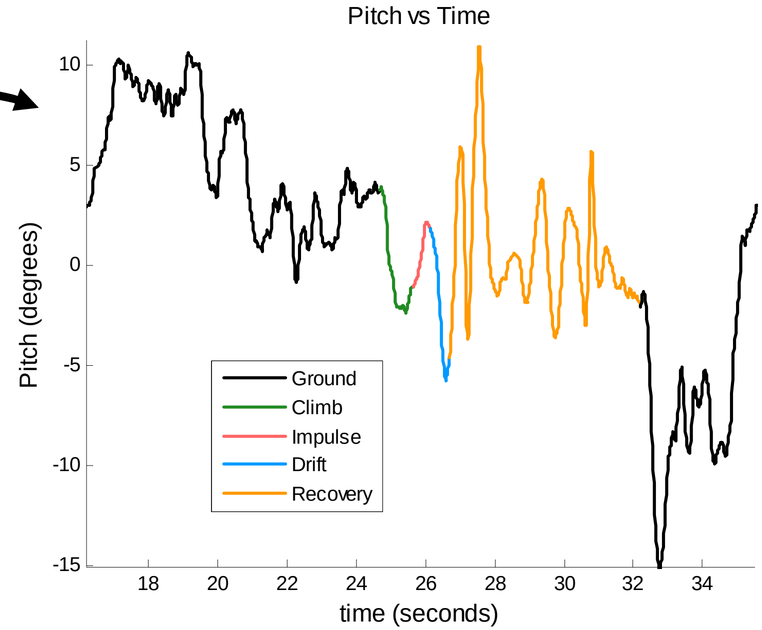
Back-Flip: Results

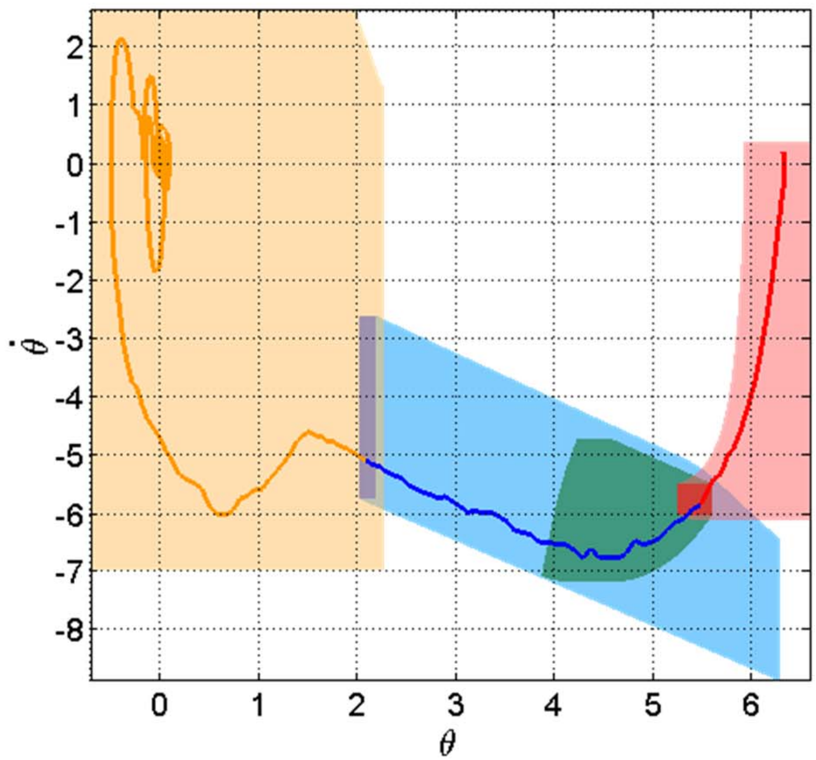


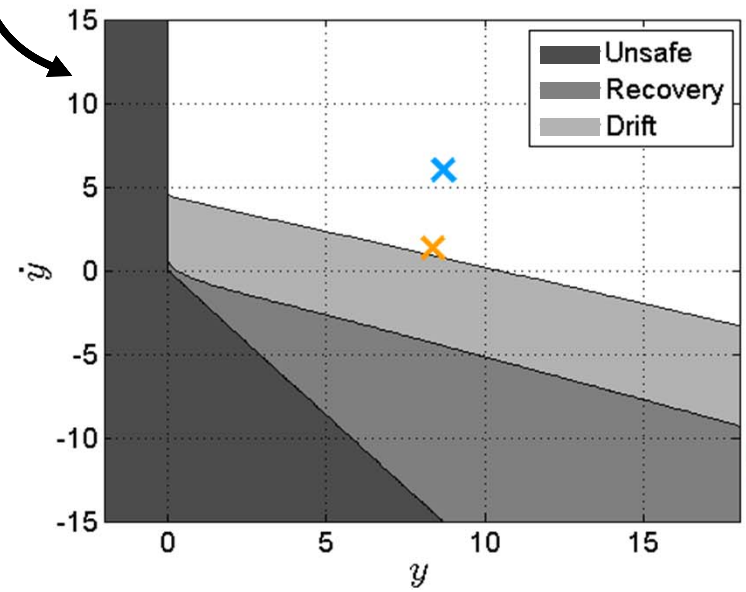
Back-Flip: Results

BACK-FLIP: RESULTS

- Assumptions Validated
- Safety Guaranteed
- Reachability Demonstrated

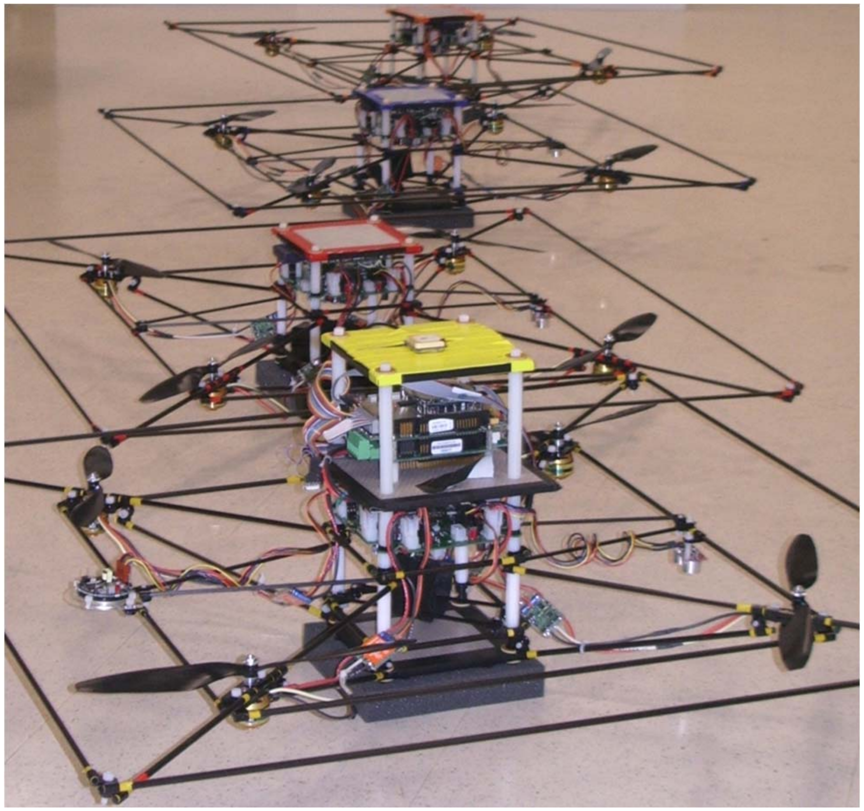






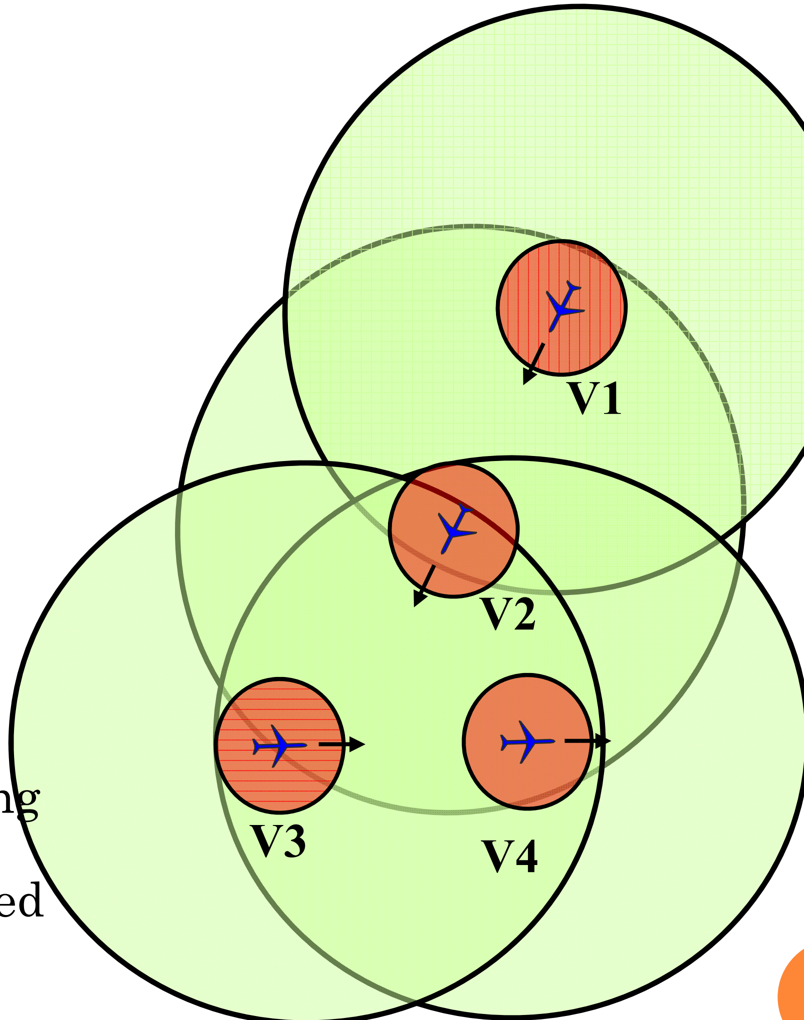
Outline

- Introduction
- Platform Development
- Vehicle modeling
- Estimation and Control
- Mapping
- Motion Planning
- Multi-Vehicle Coordination
  - Collision Avoidance
  - Information Seeking Control



Decentralized Collision Avoidance

- Maximum communication range
  - Vehicle neighborhood
- Minimum separation constraints
  - Only interconnection between vehicles
- Desired trajectory
  - Based on local information
- Penalty method for enforcing interconnected constraints
  - Gradually increase cost of violating constraints
  - No initial feasible solution required



System Cost Metric

- Efficient solution
  - Comparison: marginal cost

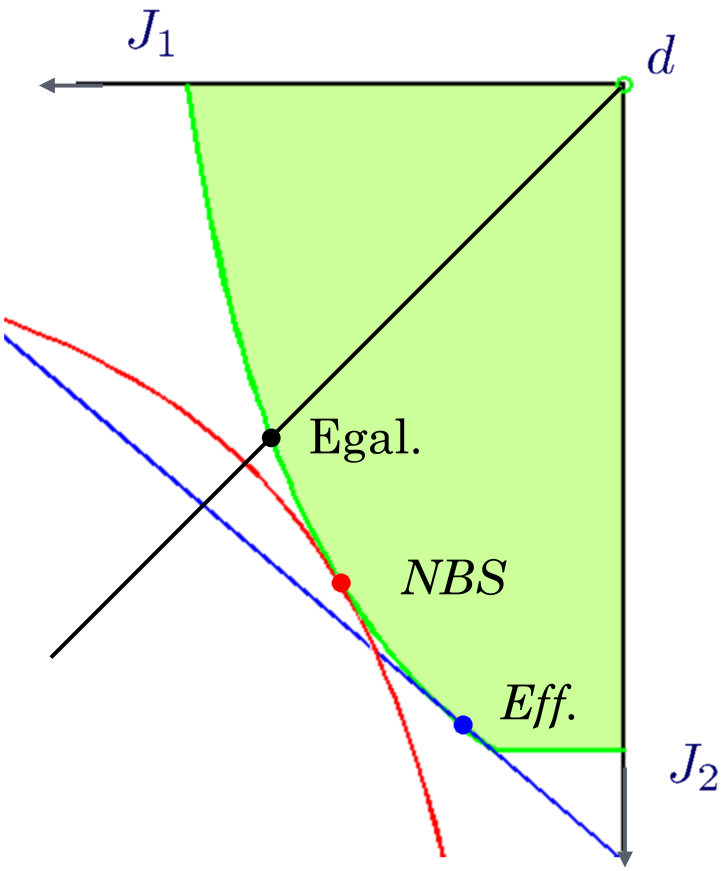
$$J_{eff} = \sum_{j \in \mathcal{J}} J_j$$

- Egalitarian solution
  - Comparison: cost

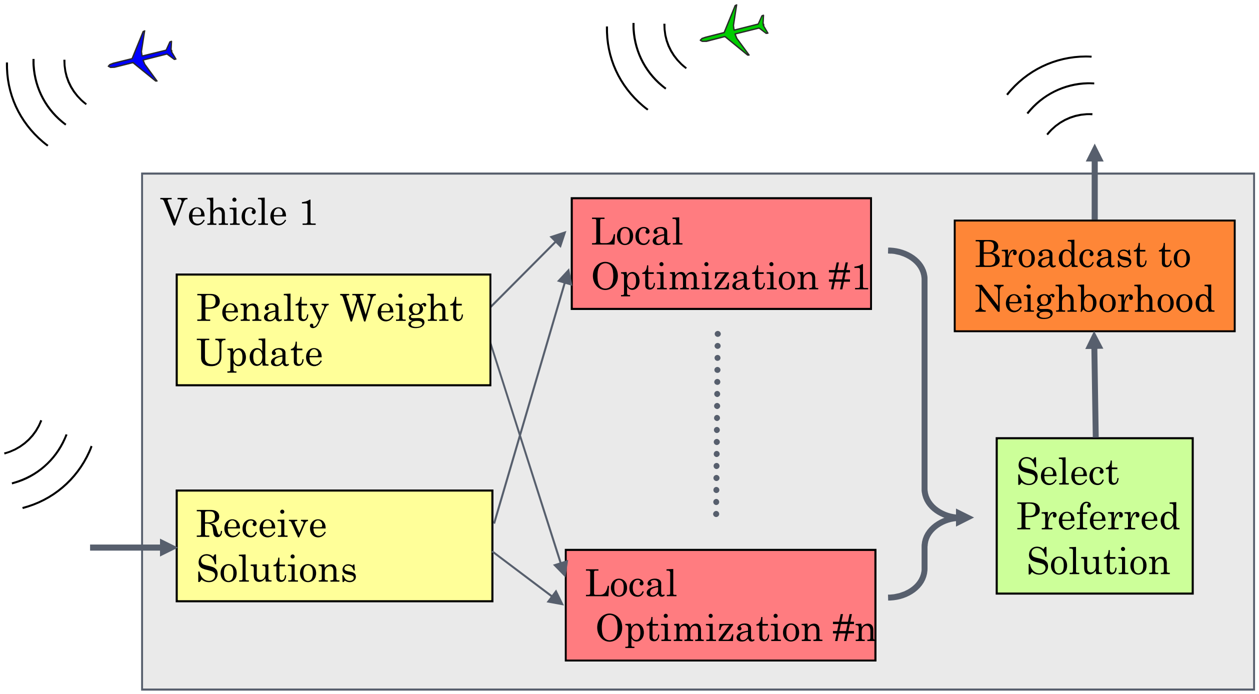
$$J_{eg} = \max_{j \in \mathcal{J}} (J_j - d_j)$$

- Nash Bargaining solution
  - Comparison: percent change in marginal cost

$$J_{NBS} = -\prod_{j \in \mathcal{J}} (d_j - J_j)$$



DECENTRALIZED ALGORITHM



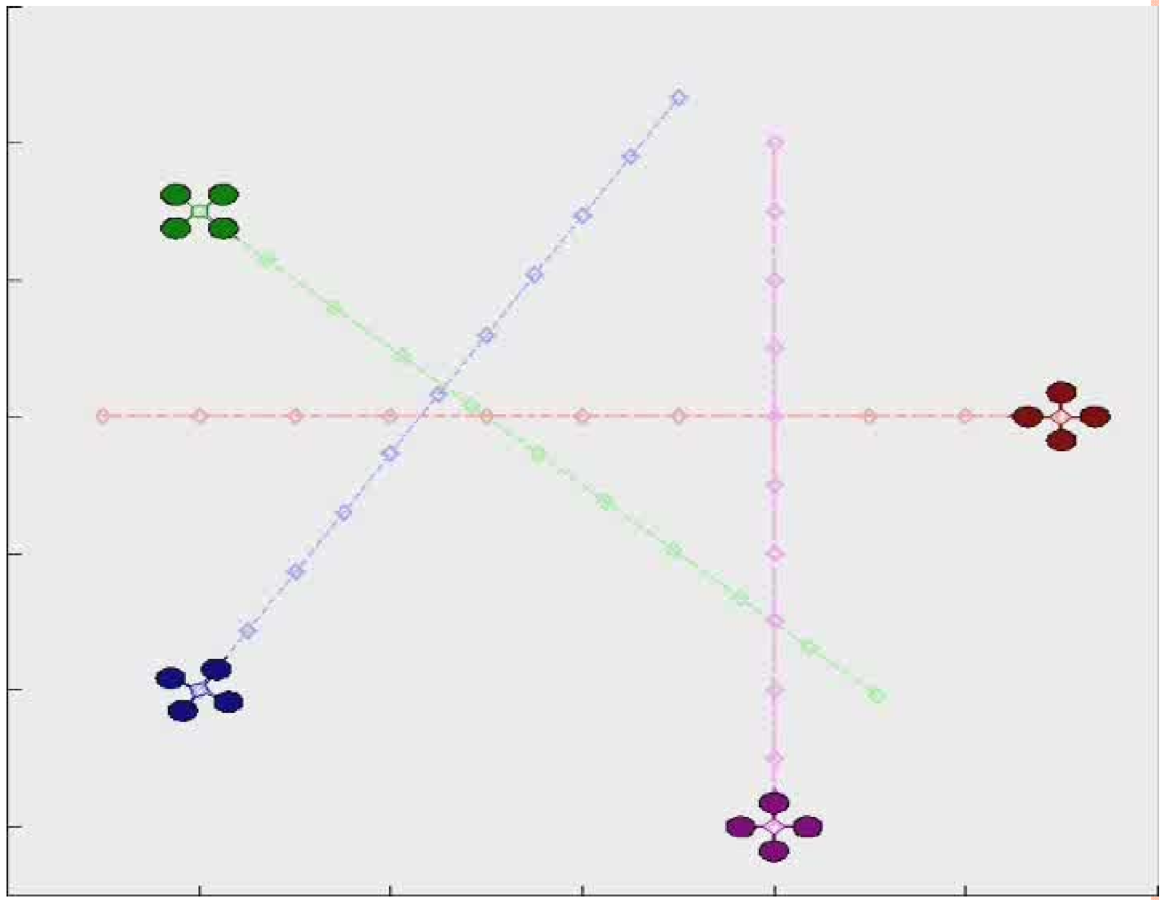
- Proposition [Convergence to Nash Bargaining solution]:

  The penalty method formulation of the decentralized optimization problem converges to a solution that satisfies the necessary conditions for optimality of the Nash Bargaining solution to the centralized optimization problem.

[Waslander, Inalhan, Tomlin, 2003]

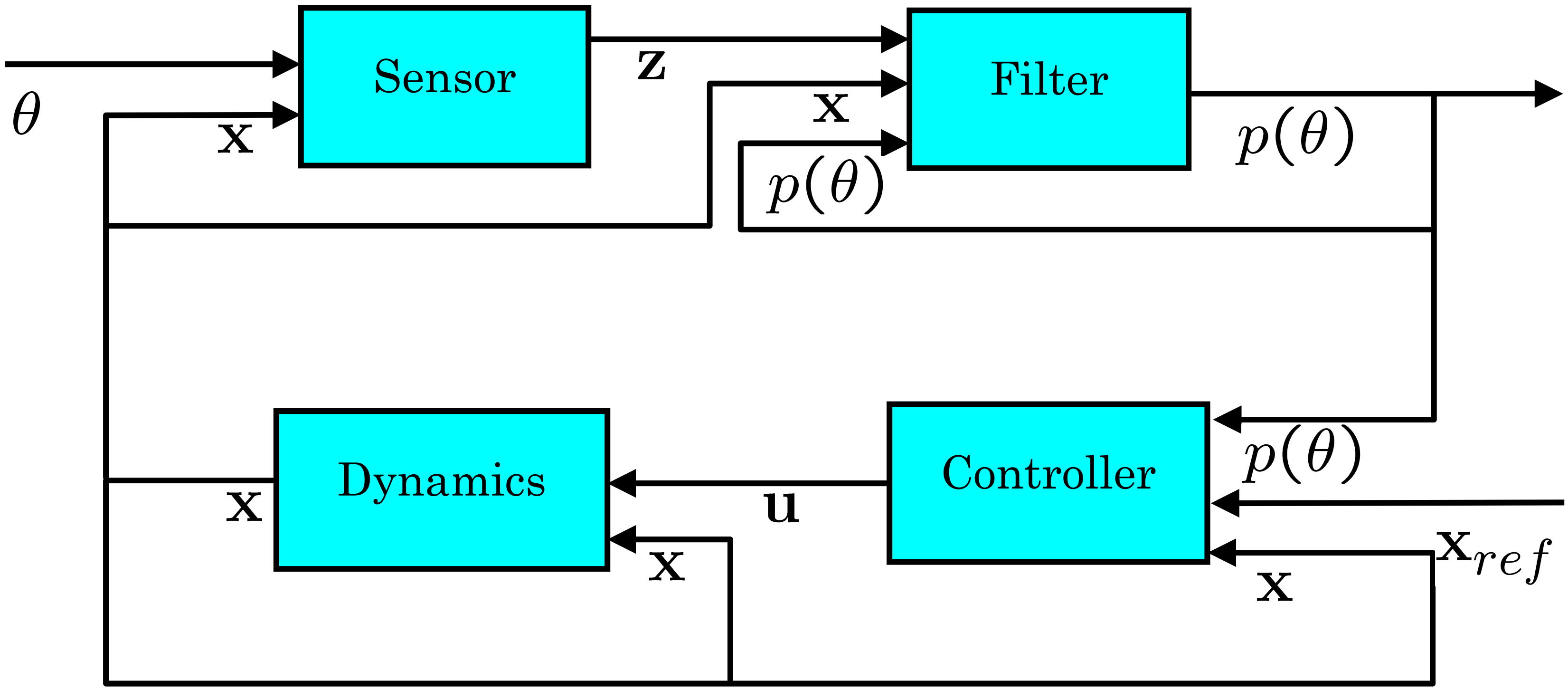
QUADROTOR SCENARIO

- Onboard computation
  - 2 m separation
  - Planning at 0.1 Hz onboard for 10 step horizon
- Implementation requirements
  - Finalize trajectory tracking control
  - Test in-flight message passing



STARMAC Flight Tests

INFORMATION-SEEKING PROBLEM FRAMEWORK



Controller goal is to minimize the uncertainty of $p(\theta)$

There may be *no reason* to move towards the target

Modeling Uncertainty to Increase Knowledge

Target State: θ
Observations: $\mathbf{z}_{t+1}$
Target State Model: $p(\theta)$

Vehicle States: $\mathbf{x}_t$
Control Inputs: $\mathbf{u}_t$
Motion Model:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$$

Sensor model: $p(\mathbf{z}_{t+1}|\theta; \mathbf{x}_{t+1})$

Use Bayes' Rule to update the target state model,

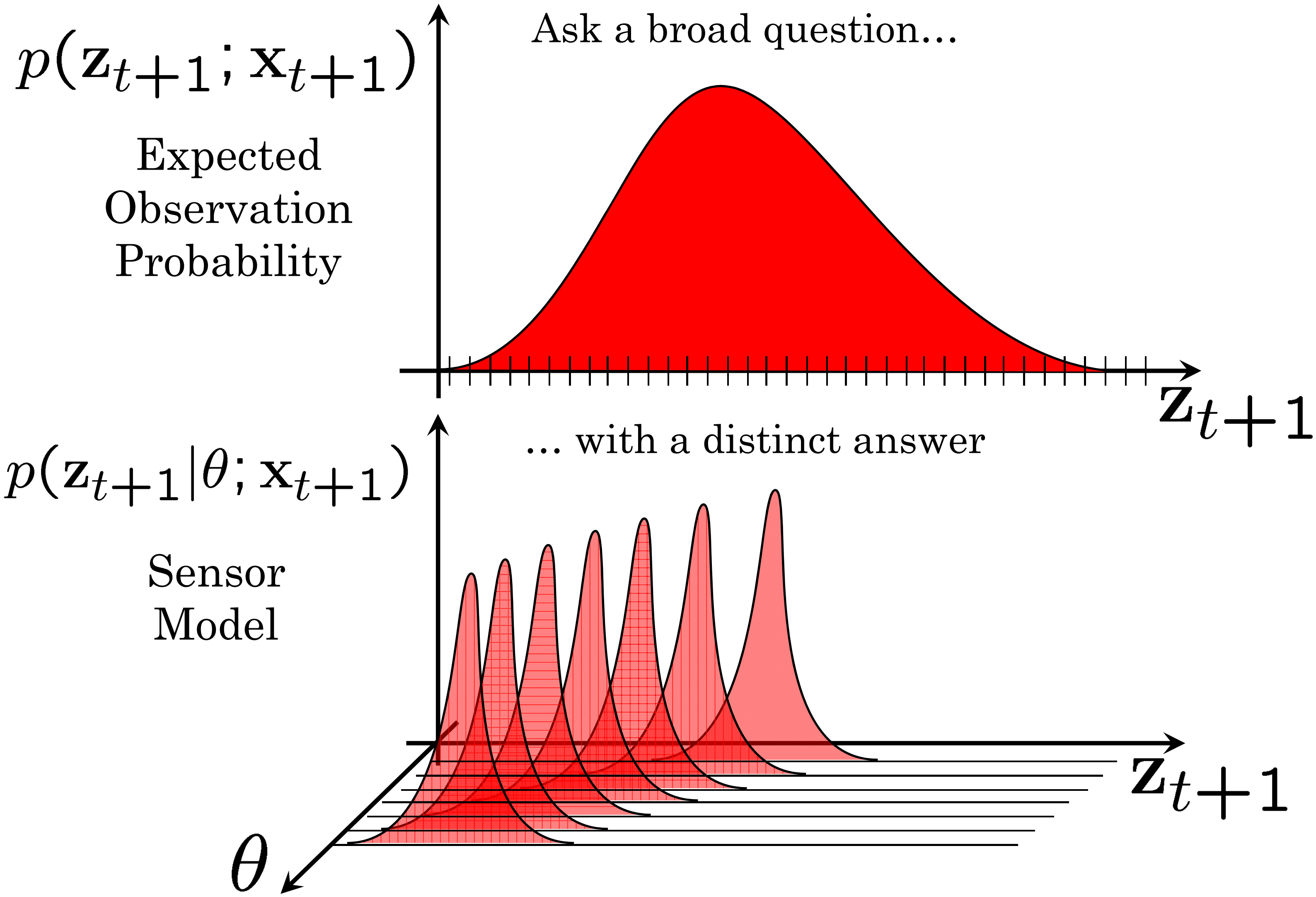
$$p(\theta|\mathbf{z}_{t+1}; \mathbf{x}_{t+1}) = \frac{p(\theta)p(\mathbf{z}_{t+1}|\theta; \mathbf{x}_{t+1})}{p(\mathbf{z}_{t+1}; \mathbf{x}_{t+1})}$$

Minimize the expected future uncertainty,

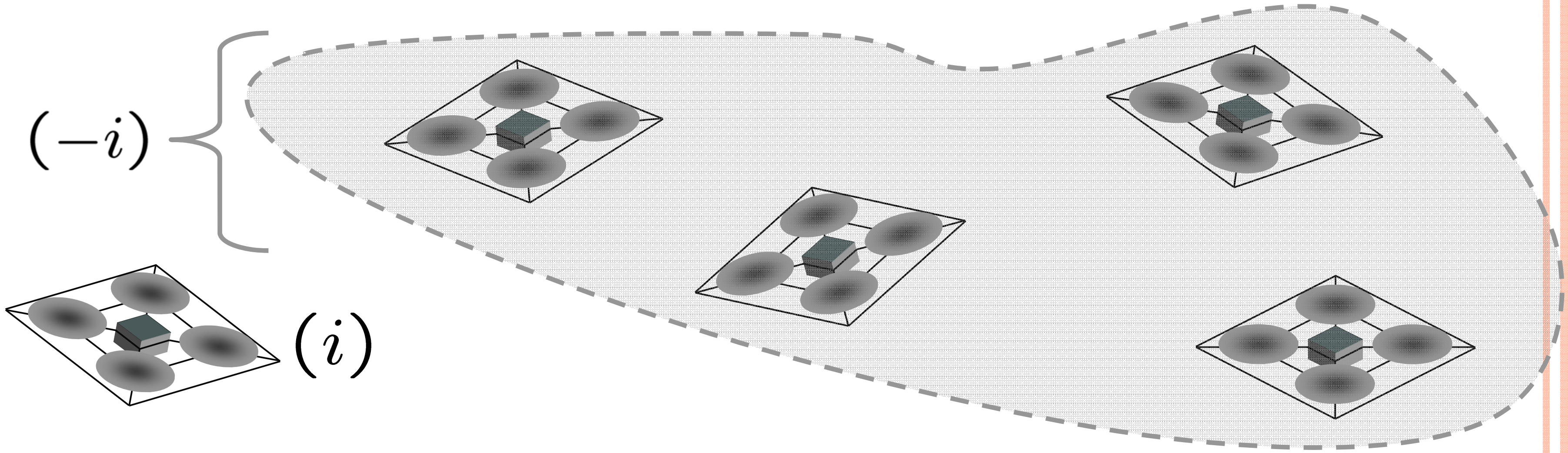
$$H(\theta|\mathbf{z}_{t+1}) = H(\theta) - I(\theta; \mathbf{z}_{t+1})$$

Maximizing Information

$$I(\mathbf{z}_{t+1}; \theta_t^{(i)}) = H(\mathbf{z}_{t+1}) - H(\mathbf{z}_{t+1} | \theta_t^{(i)})$$



Distributed Optimization Program



$$\underset{\mathbf{u}_t^{(i)} \in U^{(i)}}{\text{minimize}} \quad -I^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}, \theta_t^{(i)} | \mathbf{x}_t^{(-i)}, \mathbf{u}_t^{(-i)}) + \frac{1}{\beta} P(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)} | \mathbf{x}_t^{(-i)}, \mathbf{u}_t^{(-i)})$$

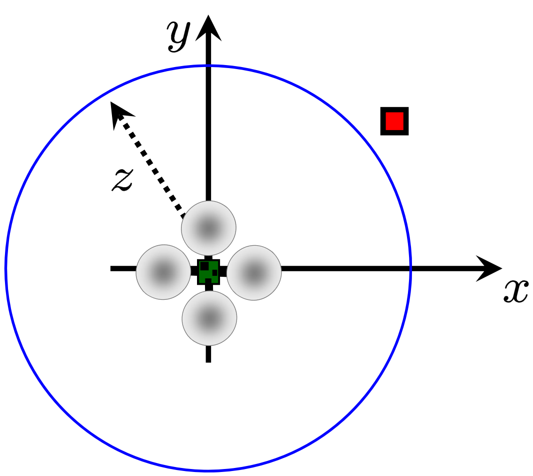
$$\text{subject to} \quad \mathbf{x}_{t+1}^{(i)} = f_t^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)})$$
$$\mathbf{z}_{t+1}^{(i)} = h_t^{(i)}(\mathbf{x}_{t+1}^{(i)}, \theta_t^{(i)}, \eta_t^{(i)})$$

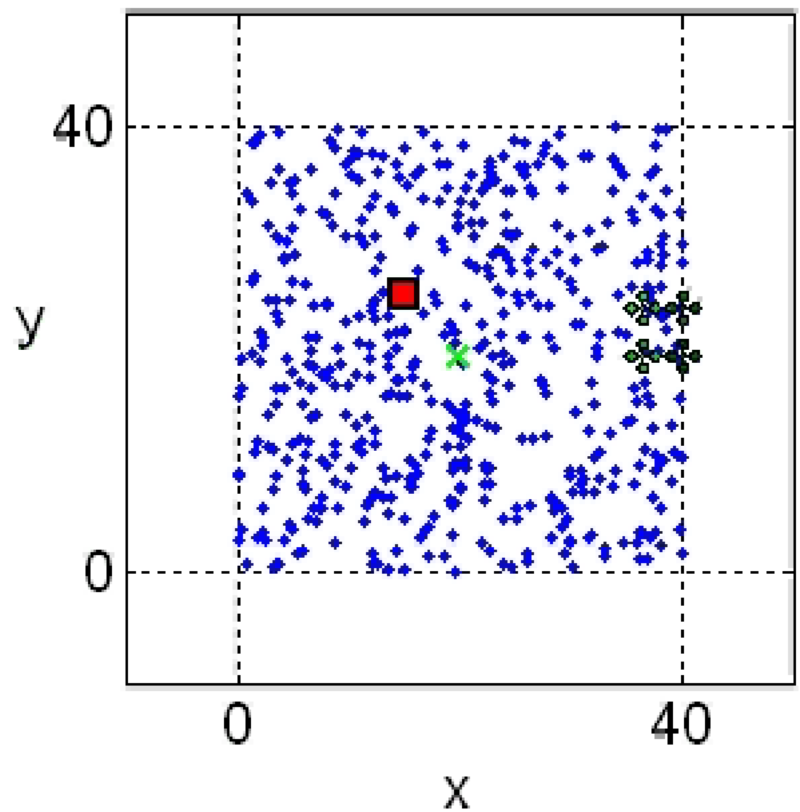
Ihalhan, Stipanovic, Tomlin, *CDC*, 2002

Hoffmann, Waslander, Tomlin, *CDC, 2006*

Range-Only Example

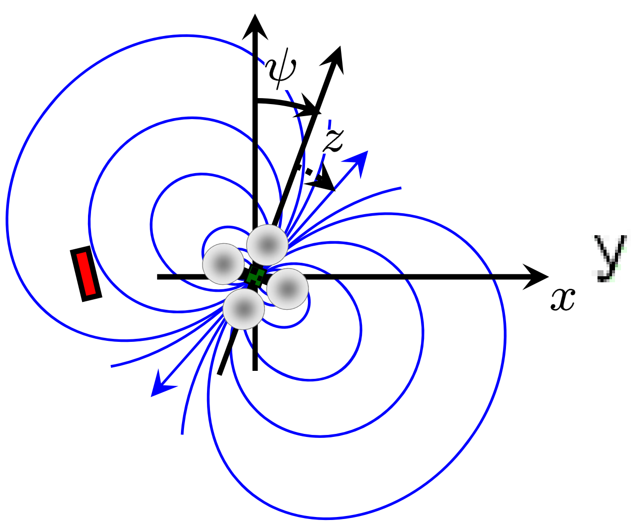
Measure the distance to the target

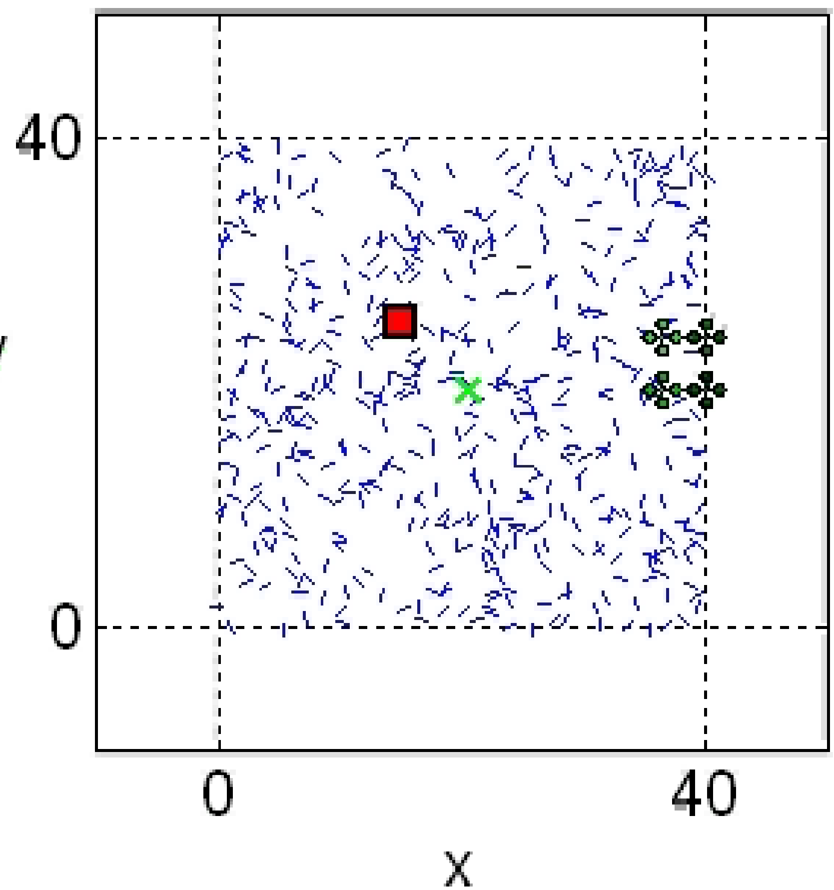




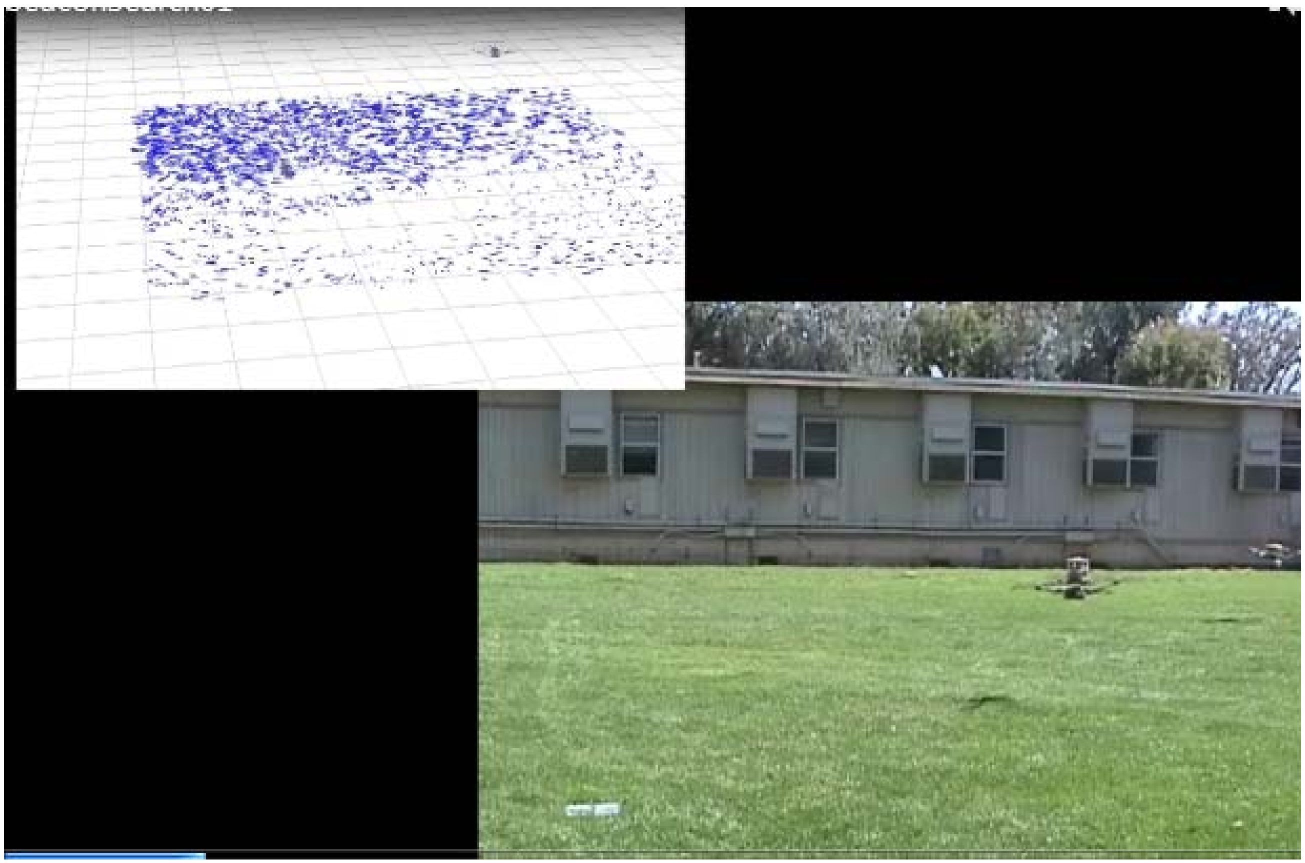
BEACON FIELD EXAMPLE

Measure the field line orientation





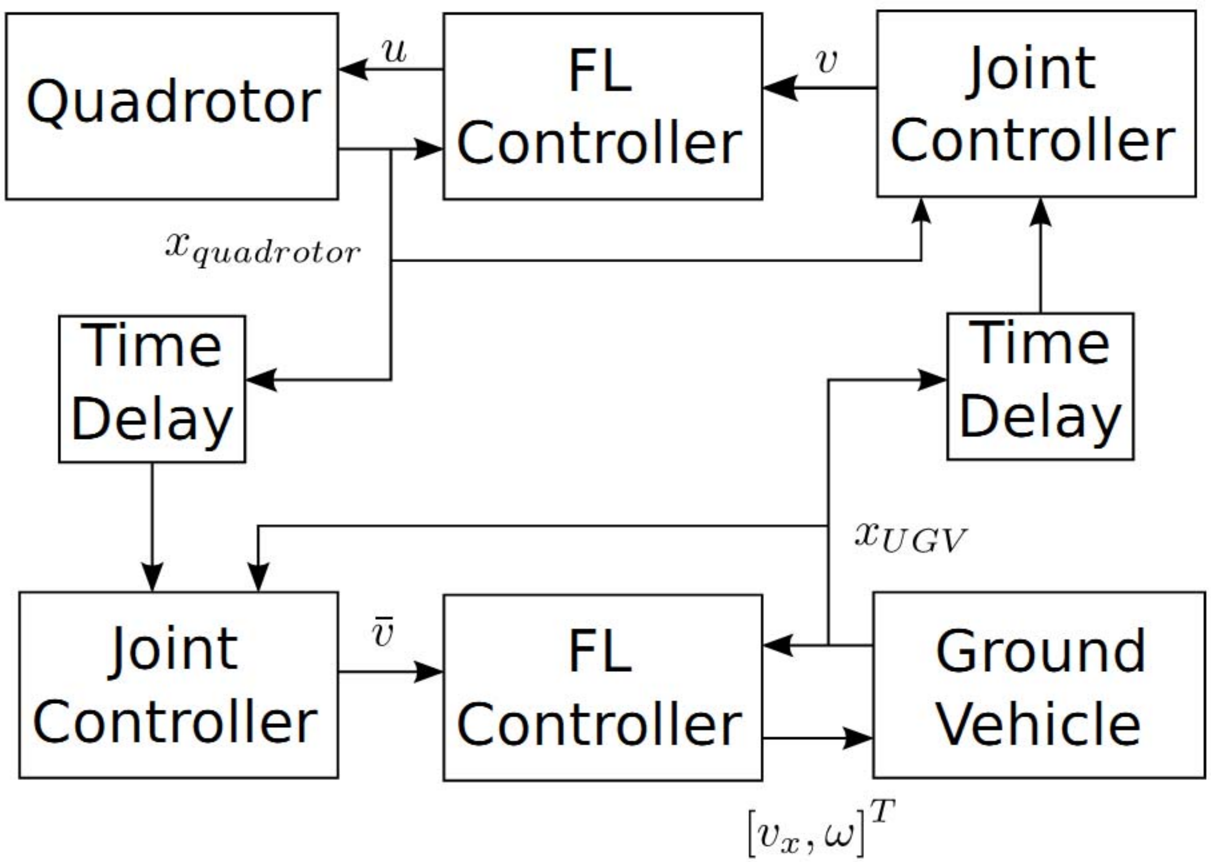
Beacon Search Field Example



UAV/UGV Coordination

- Autonomous coordinated landing
  - Use precision motion of ground vehicle, manoeuvrability of aerial vehicle
  - Design landing controller
    - Low computation requirements

Control Architecture



Delay Margin

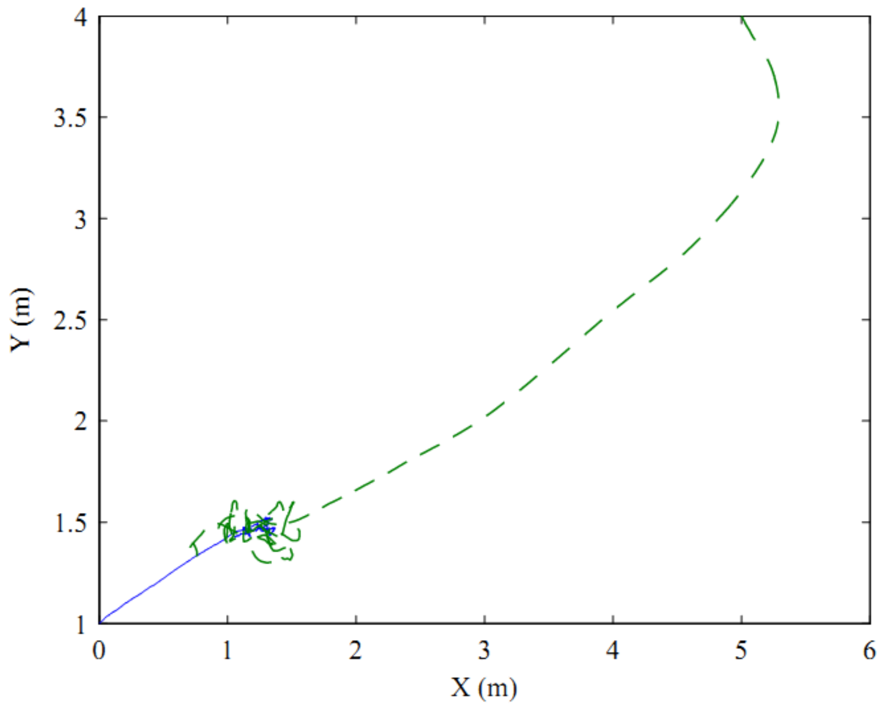
- Using results in [1], we can numerically determine delay margin
  - Joint controller designed to give closed loop poles at $\{-1, -2, -3, -4\}$ yields delay margin $\tau = 0.52327$ seconds.
  - Joint controller designed to give closed loop poles at $\{-0.1, -0.2, -0.3, -0.4\}$ yields delay margin $\tau = 6.1237$ seconds.

- As CL system becomes more sluggish, delay margin increases
  - Intuitively satisfying

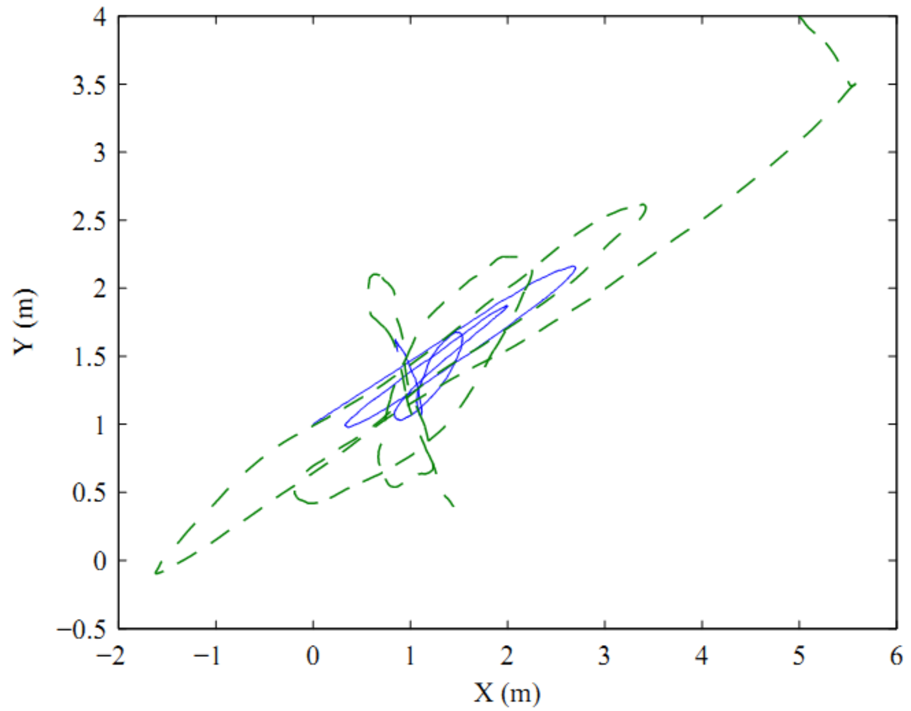
[1] K. Gu, V. L. Kharitonov, and J. Chen, Stability of Time-Delay Systems. New York: Birkhauser Boston, 2003.

SIMULATION VERIFICATION

$\tau = 1s$



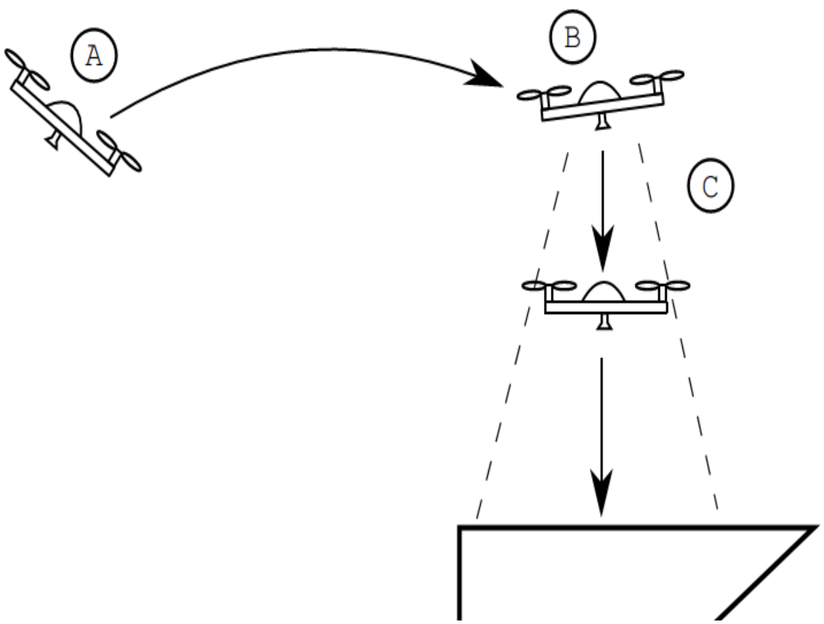
$\tau = 6.2s$

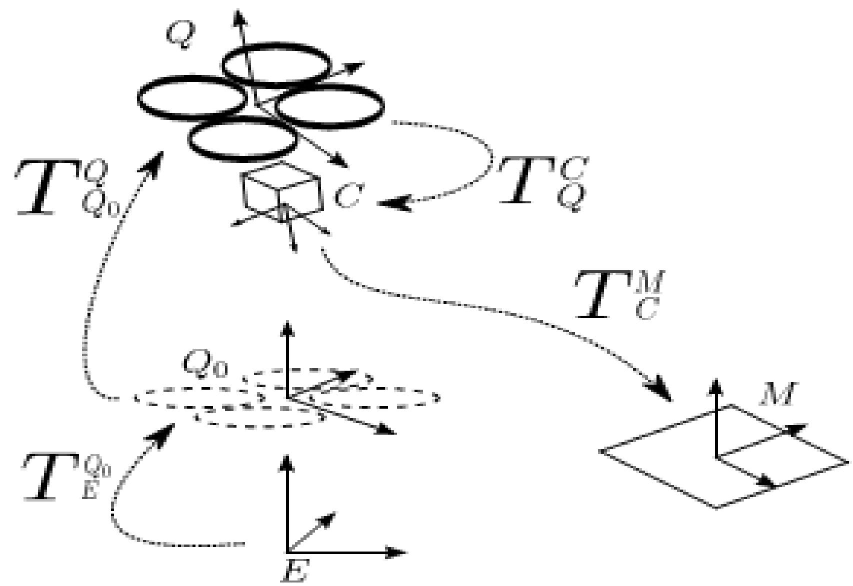


Flight Demonstration

Visual Docking for UAV/UGV teams

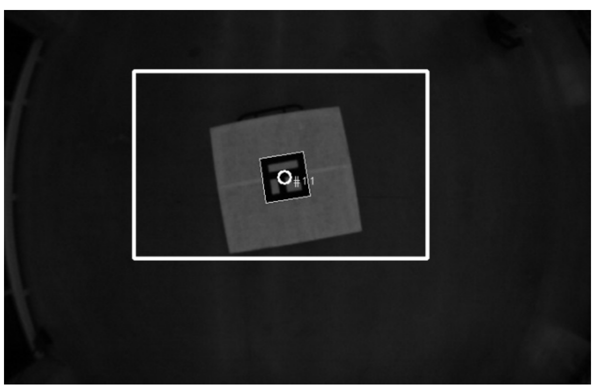
- Three phase process: rendezvous, tracking then descent
- Focus on careful selection of which sensors to rely on
  - GPS, IMU for rendezvous – 5-10 m positioning error
  - Vision only for acquisition, descent – 10 cm relative error without wind.

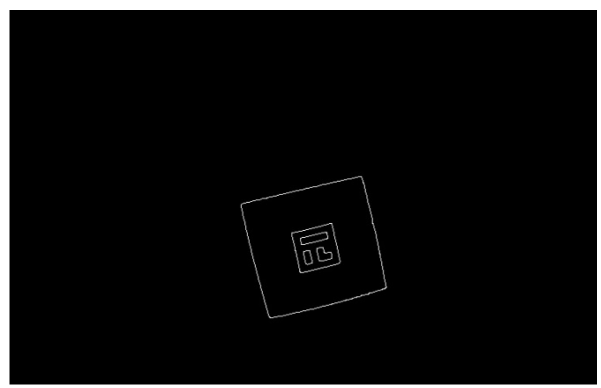




2DOF Relative Pose Estimation

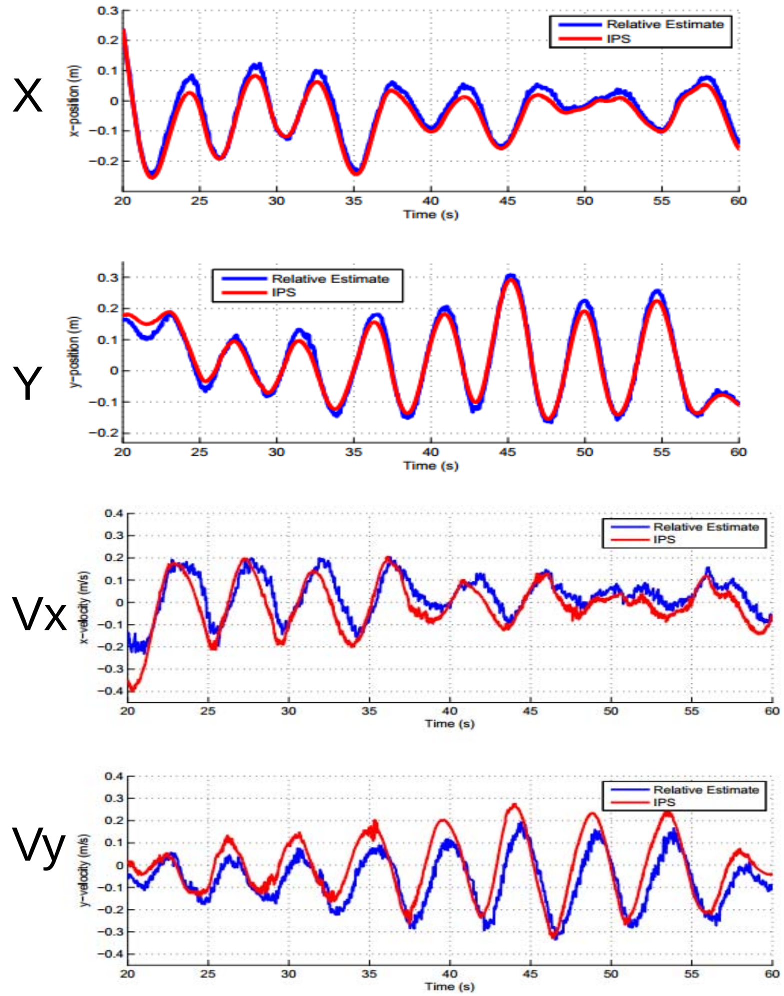
- Based on April Tags code, simplified and sped up
  - Only need x, y, vx, vy estimates
  - Windowed tracking based on constant velocity model
  - Canny edges simplify image processing
  - Lots of illumination/reflection issues
    - Felt target, mimimum exposure time (low blur)
  - Attained 25 Hz update on Intel Atom boar





2DOF Tracking Performance

- Tracking performance compared to indoor positioning data very reassuring
  - Delay in velocity measurement visible
  - Low noise overall, as long as target tracking is maintained



Controller designs

- Rendezvous in GPS Inertial Frame, cross track/along track

$$u_{at} = K_{at,d}\dot{e}_{at} + K_{at,i}\int \dot{e}_{at}dt$$

$$u_{ct} = K_{ct,p}e_{ct} + K_{ct,d}\dot{e}_{ct} + K_{ct,i}\int e_{ct}dt$$

- Acquisition in the same frame, but now relative pose regul

$$u_{E,x} = K_{E,p}e_{E,x} + K_{E,d}\dot{e}_{E,x} + K_{E,i}\int e_{E,x}dt$$

$$u_{E,y} = K_{E,p}e_{E,y} + K_{E,d}\dot{e}_{E,y} + K_{E,i}\int e_{E,y}dt,$$

- Final desc igned with target

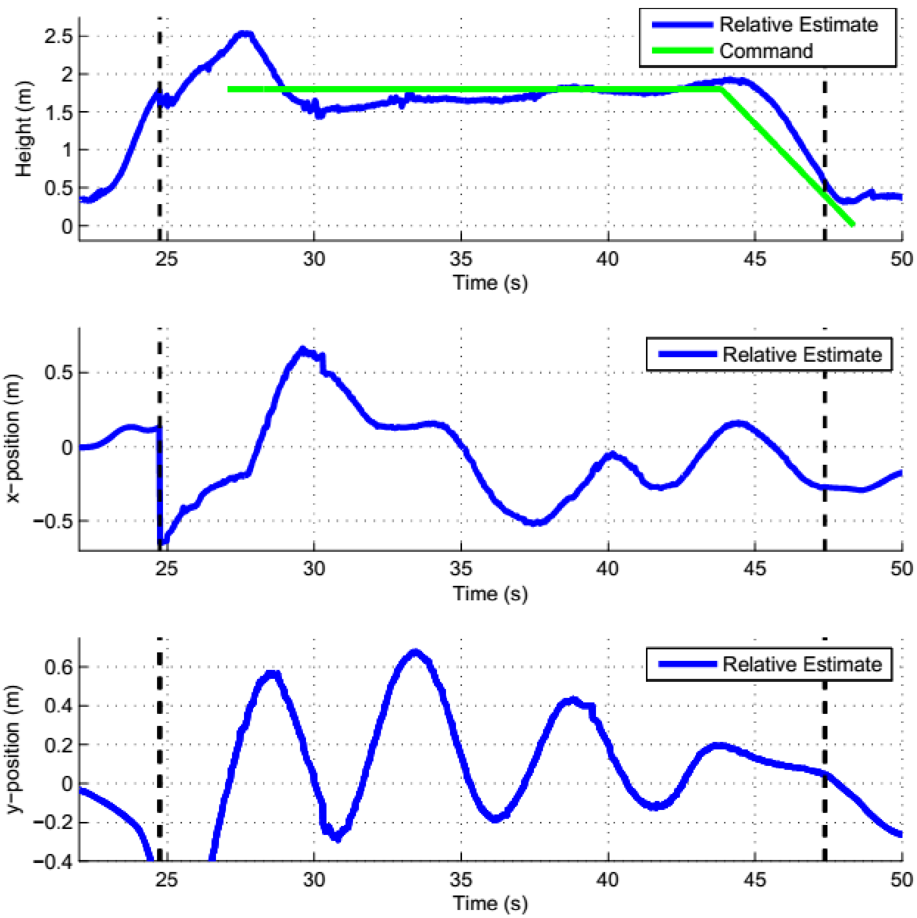
$$u_{r,x} = K_{r,p}e_{r,x} + K_{r,d}\dot{e}_{r,x} + K_{r,i}\int e_{r,x}dt$$

$$u_{r,y} = K_{r,p}e_{r,y} + K_{r,d}\dot{e}_{r,y} + K_{r,i}\int e_{r,y}dt,$$

VISUAL DOCKING FOR UAV/UGV TEAMS

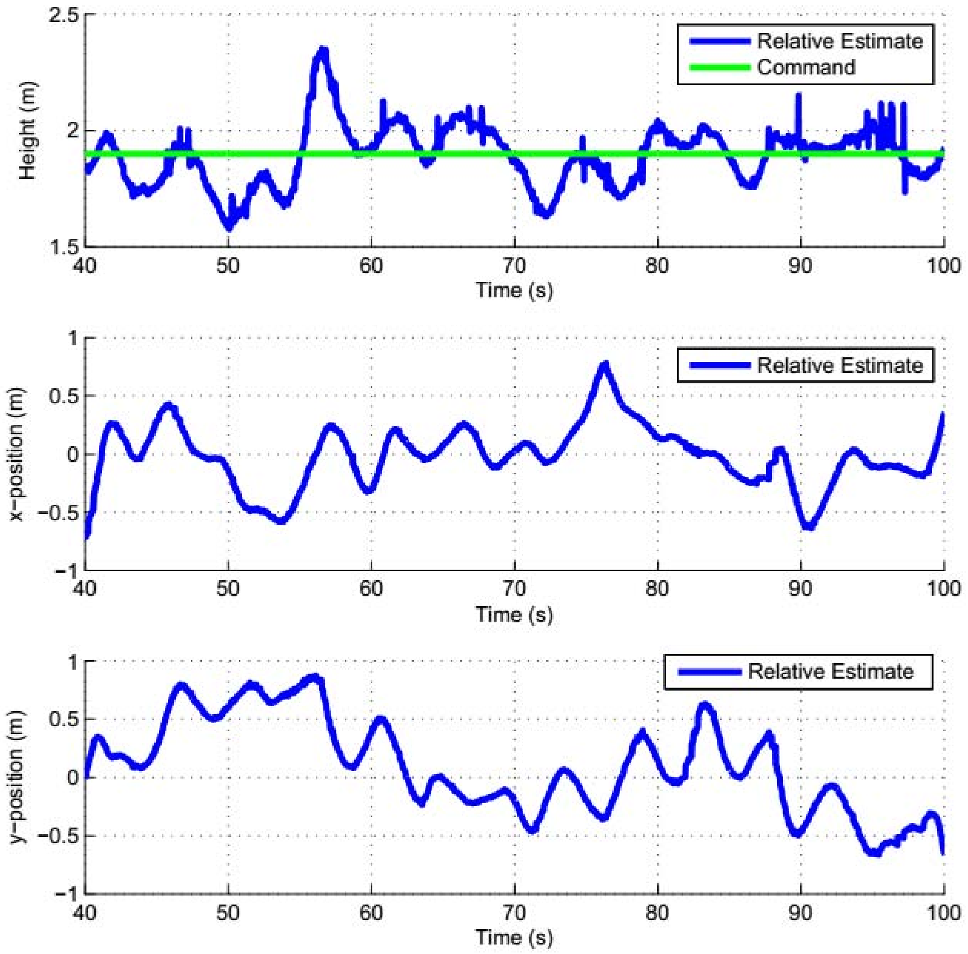
Visual Docking for UAV/UGV teams

- Indoor landing relative position results



Visual Docking for UAV/UGV teams

- Outdoor tracking relative position results



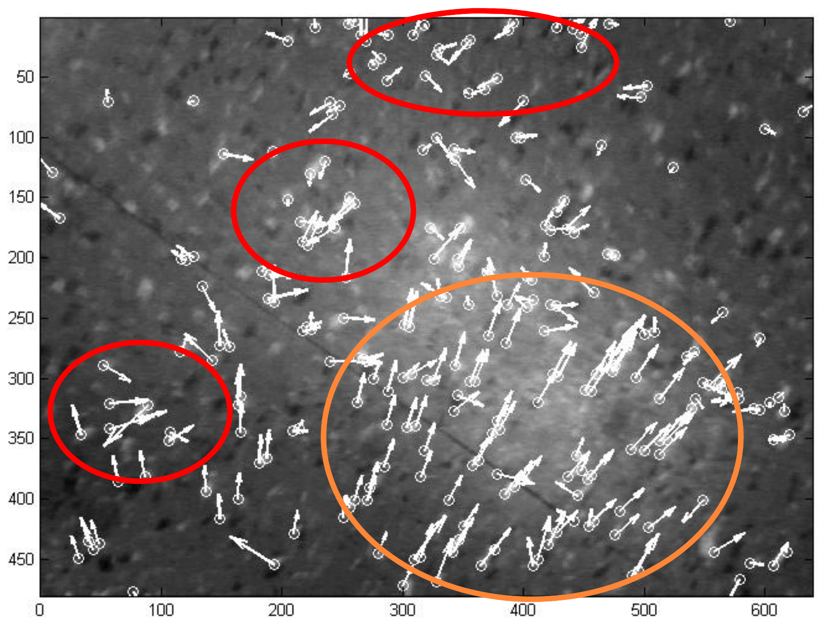

QUESTIONS?

Extra Slides

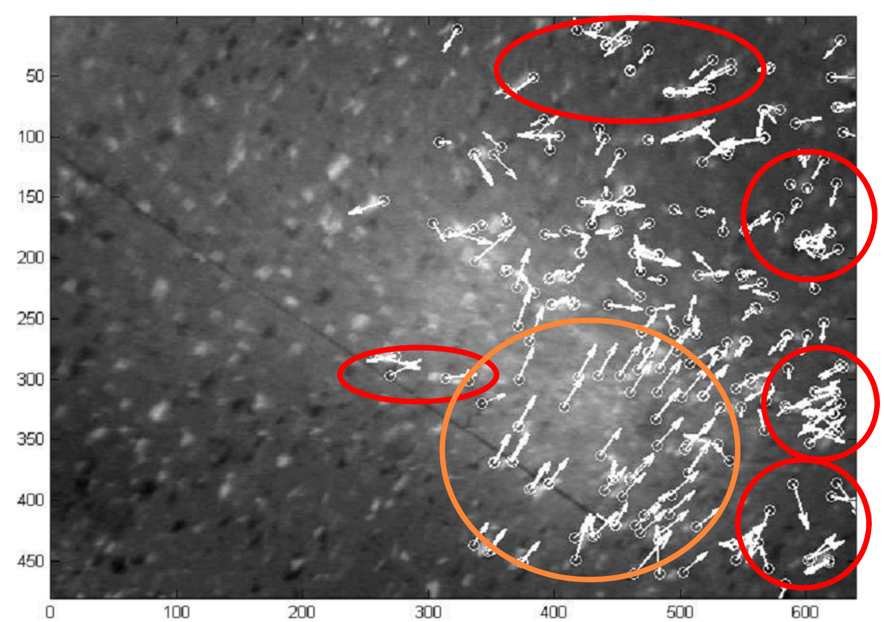
Visual Motion Estimation

Visual Motion Estimation

- Biggest Issue is low quality of features
- Apply RANSAC to low quality feature
  - Embrace existence of outliers (reject accordingly)



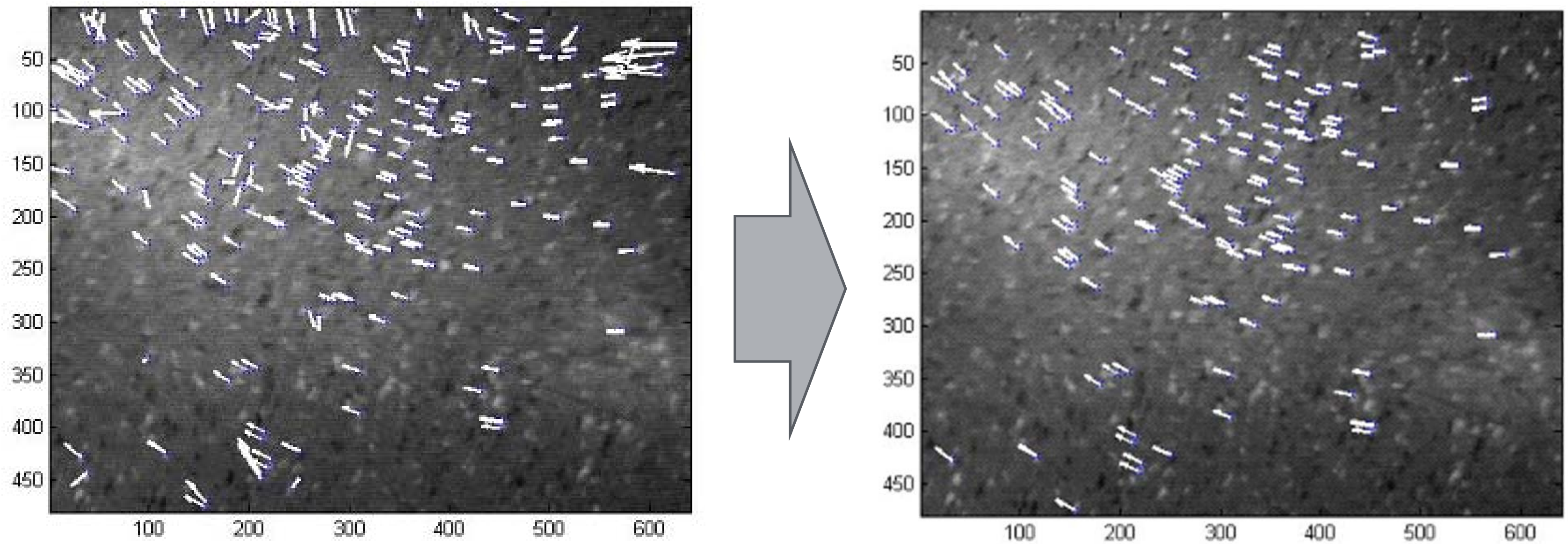
Typical case – Shi & Tomasi



Typical case – SURF

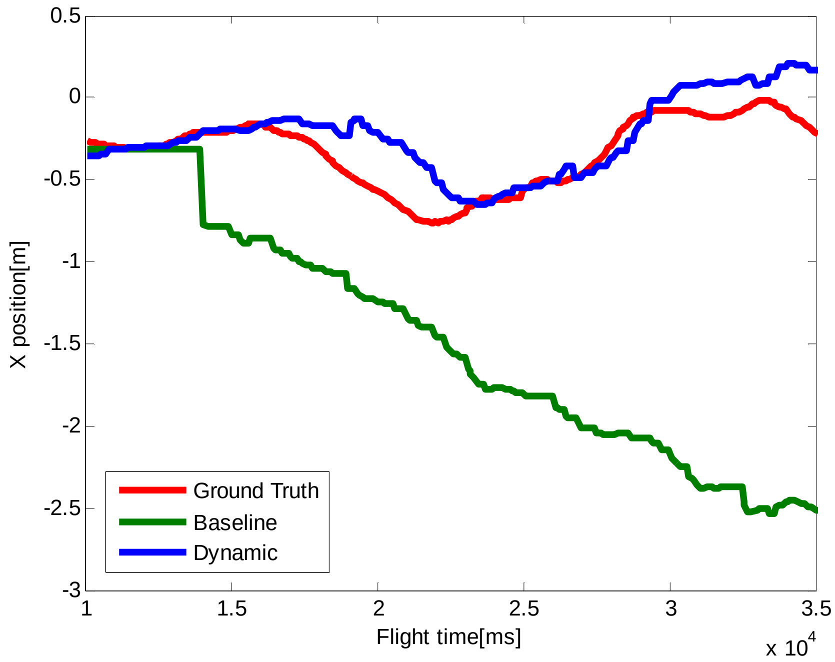
Optical Flow Comparison

- Before and After RANSAC

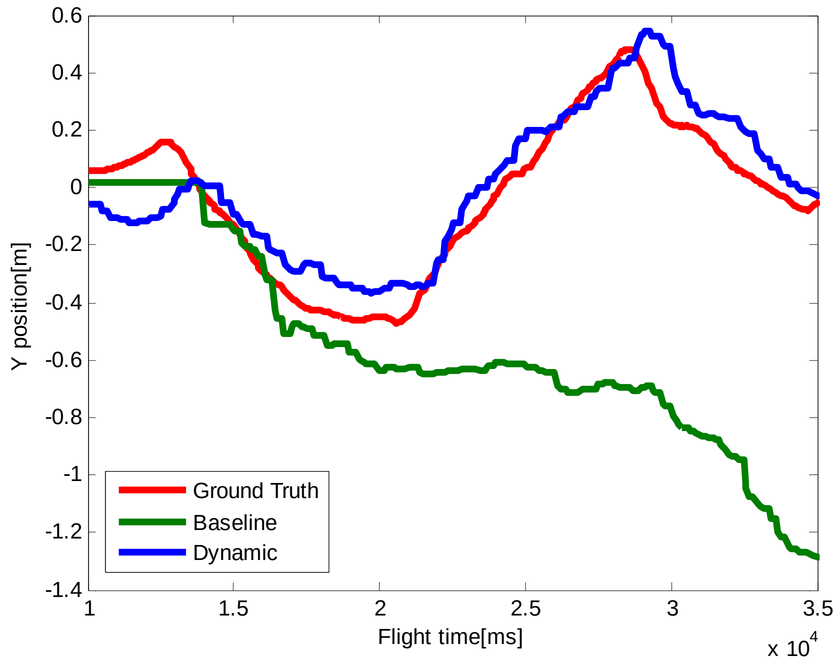


- Once a consistent set of measurements are found
  - EKF to estimate velocity and position, heading
  - Control accordingly

Results – Estimation

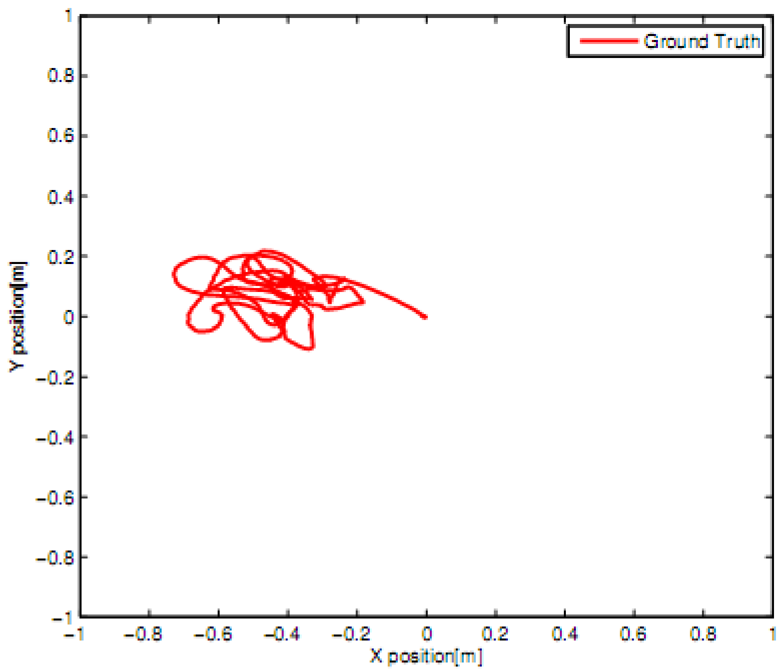


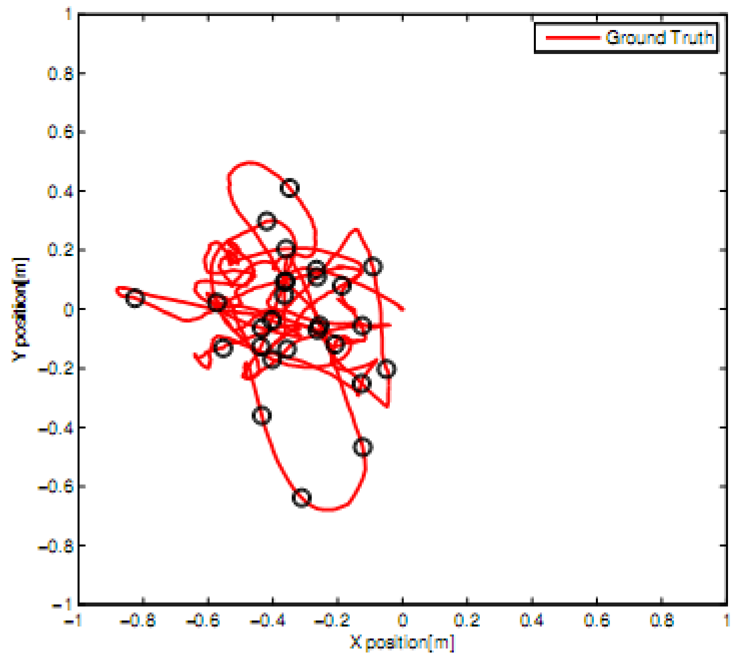
(a) X position over time



(b) Y position over time

RESULTS – CONTROL





Parrot Drone