Acknowledgements

Spin off companies

- Trefo.ai
- Identified Technologies
- exyn technologies
- We Robotics

acquired by Qualcomm
acquired by Penn Engineering
applications to agriculture
aerial mapping
mining, asset mapping
health, disaster response
Non profit
A Brief History of Aerial Robotics

- International Aerial Robotics Competition (1991)
- Early work at GRASP (< 2000)
There are no new ideas, only good ideas!

Breguet-Richet Gyroplane No.1, 1907

1907, Paul Cornu: First to Hover?

George de Bothezat’s Quadrotor, 1922

Flying windmill, 2007

A Brief History of Aerial Robotics

• International Aerial Robotics Competition (1991)
• Early work at GRASP (< 2000)
Inertial Measurement Units

• Accelerometer, airbag sensors (Analog Devices), 1993
• MEMS gyros for electronic stability control (Bosch), 1997
• 3-axis accelerometers for Nintendo Wii, 2006
• 3-axis accelerometer, iPhone (Apple), 2007
• 3-axis accelerometer, 3-axis gyro, 3-axis magnetometer, iPhone 4 (Apple), 2010

Inflexion point
A Brief History of Aerial Robotics

- International Aerial Robotics Competition (1991)
- Early work at GRASP
- 2008-9 – small multi rotor aircrafts become practical
2012 – Swarm of 75 gm quadrotors

Daniel Mellinger

Alex Kushleyev
2015 – Drones everywhere!
Drone Racing

Drone Nationals, New York City, 2016

Yash Mulgaonkar
Beyond Quadrotors


David Saldana, Bruno Gabrich, Guanrui Li, Mark Yim, and Vijay Kumar, “ModQuad: The Flying Modular Structure that Self-Assembles in Midair,” 2018.
Gartner Hype Cycle
2019 (10 years later)

Market

$100BN
2016-2020

70%
military

13%
b2b

17%
consumer

Agriculture, Mining, Health

Goldman Sachs Research

Penn Engineering
Aerial Robotics Research (and Commercialization)

We are here! Safe Smart Speed Swarms

Small
... and at high speeds

The Falcon
Search and Rescue
Five Challenges

- Perception Action Loops for Autonomy
- State Estimation
- Navigation in Cluttered Environments
- Scaling Down in Size, Weight
- Perception Action Communication Loops for Swarms
1. Nested Perception/Action Loops

Nonlinear controllers on SO(3)/SE(3)

T Lee, M Leok, NH McClamroch, “Nonlinear Robust Tracking Control of a Quadrotor UAV on SE(3),” Asian Journal of Control 2012.
2. State Estimation (Stereo + IMU)

Model

\[
\begin{align*}
\dot{x}_I & = \left( I_G q^T, b_g^T, G v_I^T, b_a^T, G p_I^T, I q_C^T, I p_C^T \right)^T \\
\dot{I}_G q & = \frac{1}{2} \Omega(\omega)_G I_G q, \\
\dot{b}_g & = n_{wg}, \\
\dot{b}_a & = n_{wa}, \\
G \dot{v} & = C \left( I_G q \right)^T a + G g, \\
I \dot{q}_C & = 0_{3 \times 1}, \\
I \dot{p}_C & = 0_{3 \times 1}
\end{align*}
\]

Augmented State

\[
x_{C_i} = \begin{pmatrix} C_i q^T & G p_{C_i}^T \end{pmatrix}^T 
\]

\[x = \left( x_I^T \ x_{C_0}^T \ \cdots \ x_{C_{N-1}}^T \right)^T\]

Stereo Camera Measurement

\[
z_i = \begin{pmatrix} u_{i,1} \\ v_{i,1} \\ u_{i,2} \\ v_{i,2} \end{pmatrix} = \begin{pmatrix} \frac{1}{c_{i,1} z} & 0_{2 \times 2} \\ 0_{2 \times 2} & \frac{1}{c_{i,2} z} \end{pmatrix} \begin{pmatrix} C_{i,1} X \\ C_{i,1} Y \\ C_{i,2} X \\ C_{i,2} Y \end{pmatrix} + n_z, \quad c_{i,j} p_f = \begin{pmatrix} C_{i,j} X \\ C_{i,j} Y \\ C_{i,j} Z \end{pmatrix} \quad j \in \{1, 2\}
\]
Stereo Multistate Constraint Kalman Filter (S-MCKF)

Fast autonomous flight (Top speed at 18m/s)

Autonomous flight in unstructured environment
- Includes various scenes (warehouse, woods, open field, etc).
- Round trip over 700m
- Final drift under 0.5%

- **GPS (10 Hz)**
- **Laser Scanner (20 Hz)**
- **Altimeter (20 Hz)**
- **Stereo Camera (40 Hz)**
- **Downward Camera (40 Hz)**
- **IMU (1000 Hz)**

**Position**:
- **Laser odometry**
- **Altitude estimator**
- **Visual odometry**
- **Visual odometry**
- **Local Map (40 Hz)**

**Velocity**:
- **Controller (200 Hz)**

**Multi-Sensor Unscented Kalman Filter (200 Hz)**

- **Pose Graph SLAM**
- **Planner (20 Hz)**
- **Trajectory Generator (20 Hz)**
DARPA Fast Lightweight Autonomy (FLA)
3 Planning in Cluttered Environments

I. Optimal Control

\[
\begin{align*}
\min_{u(t), T} & \quad J(x(t), u(t)) + \rho T \\
\dot{x} & = Ax(t) + Bu(t), \ u(t) \in \mathcal{U}, \ \forall t \in [0, T] \\
x(0) & = x_0, \ x(T) \in X^{\text{goal}}, \ x(t) \in X^{\text{free}} \\
X^{\text{goal}} & \subset X^{\text{free}}
\end{align*}
\]

- Relative degree 4 (input and state constraints)
- Non convex
- Safe corridors in different homology classes
- Partially known environment (limited field of view sensors)


Planning in Cluttered Environments

2. Search-Based Planning with Motion Primitives

Minimum snap primitives

Search over induced discretization on state space

Results for different functionals

S. Liu, N. Atanasov, K. Mohta, V. Kumar, Search-based motion planning for quadrotors using linear quadratic minimum time control, IROS 2017
Resolution complete but …

\[ p_k(t_k) \]

limited field of view creates challenges

\[ v = 20 \text{ m/s}, \text{ max acceleration } 1 \text{ g} \]

\[ \text{Stopping time } \sim 2 \text{ s} \]

\[ \text{Stopping distance } \sim 20 \text{ m} \]
Safety Certificate

- Safety
- Completeness
- Suboptimality

\[ p_k(t_k) \]

\[ p_{k+1}(t_{k+1}) \]

\[ x_k(t_k) \]

\[ x_{k+1}(t_{k+1}) \]

\[ \Phi_k \]

\[ \Psi_k \]

\[ g' \]
Autonomous Flight in Unknown GPS-Denied Environment (5 m/s)
Search of Collapsed Buildings

Fully Autonomous Aerial Robot for Mine Inspection and Mapping

Stope Flight: Beyond Visual Line-of-Sight and Communications Range
4. Light Weight Autonomy

250 gram quadrotor (2018)
Qualcomm® Snapdragon Flight™
development board running Snapdragon Navigator™ flight controller and
Machine Vision (MV) SDK

1 kg quadrotor (2018)
Stereo camera synced with Vector NAV IMU, NVDIA Jetson TX2 +
FPGA (low-level pixel-wise operations) – OSRF
TOF 3-D camera, 6m range, 100x65 deg, 60 Hz – PMD technologies

2.5 kg quadrotor (2017)
Stereo camera synced with Vector NAV IMU, LiDar, Intel i7


Robustness to Collisions

250 gram quadrotor
Qualcomm® Snapdragon Flight™ board with Snapdragon Navigator™ flight controller

133 gram quadrotor capable of sustaining collisions
Qualcomm® Snapdragon Flight™ board with forward-facing stereo cameras, a downward facing camera for VIO, onboard WiFi and GPS

Autonomous Flight in Fukushima Daiichi Reactor Unit 1

Monica Garcia (SWRI), Richard Garcia (SWRI), Wataru Sato (TEPCO)
5. Aerial Robot Swarms

Perception — Action Loops

- observations inform actions
- move for better observations
5. Aerial Robot Swarms

Perception — Action — Communication Loops

Perception — Action

Task

Communication

• observations inform actions
• move for better observations

• messages inform actions
• move to communicate better

• share private perceptions
• communicate to make better observations
Decentralized Multi-Robot Teams

- Centralized methods are not practical for real-world robot deployments (Turpin, '14)
  - Partial observability by individual agents
  - Limited communication
- Decentralized, correct-by-construction policies available only for very simple cases
  - Simple communication and sensing models
  - Edge or cloud computation
  - Point robots

Tanner, Pappas and Jadabaie, 2004


Sensor Coverage
Cortes, 2004

Belta and Kumar, 2005
Distributed Learning: PAC Loops

Key Ideas:
- Learn communication policies
- Learn action policies
- Learn planning policies

(with Professor Alejandro Ribeiro)
Graph Neural Networks

- Robots act on relative position and velocity information
  - Must stay close to each other
  - Must avoid collisions
  - Must “align” themselves

Aggregates information at each node from neighboring node using graph adjacency properties

CNNs

GNNs

Penn Engineering
Aggregate over belief states of neighbors

\[ z = \sum_{k=0}^{K} h_k S^k x^{(k)} = H(S)x \]
Flocking

GNN-learned control policy (K=3)

\[
\mathbf{u}_i^+ = - \sum_{j \in \mathcal{N}_i} (\mathbf{v}_i - \mathbf{v}_j) - \sum_{j \in \mathcal{N}_i} \nabla_{r_i} U(r_i, r_j).
\]

Tanner, Pappas and Jadbabaie, 2004

(a) Average difference in velocities
(b) Average minimum distance to a neighbor
(c) Flock positions using the GNN
(d) Flock positions using the local controller

In practice, following the optimal policy to collect training data results in a distribution of states that is not representative of those seen at test time. To resolve this we use the Dataset Aggregation (DAgger) algorithm and follow the learner's policy instead of the expert's with probability 1 when collecting training trajectories [33]. The probability of choosing the expert action while training is decayed by a factor of 0.993 after each trajectory to a minimum of 0.5.

5 Results

We report results comparing (11) and (10) for point masses with fully controllable accelerations in Section 5.1. This simple setting allows for an exploration of the effect of different system parameters such as initial velocity or communication radius, to determine experimentally the scenarios on which the aggregation GNN offers good performance. In Section 5.2 we study the case of transfer learning, where we train the model in one network but test it in another (for example, with different number of agents), and also by exporting the trained architecture to other physical models beyond the point-mass model, as shown in the AirSim simulator (Sec. 7.2).
Graph Policy Gradients

- Train GNNs on a small number of robots
  - Information from k-hop neighbors is aggregated by each robot
  - Local controllers are learned by each robot
  - Centralized reward used to train the robots

- Extend to swarms with larger numbers
  - Transfer of policies to larger groups with similar “local” graph properties

Arbaaz Khan, Vijay Kumar, and Alejandro Ribeiro, Graph Policy Gradients for Large Scale Unlabeled Motion Planning with Constraints, IEEE International Conference on Robotics and Automation, submitted (2020)
Graph Policy Gradients for Large Scale Formation Control
Conclusion

• Autonomy using smartphone grade processors/sensors
• 10x improvement in performance/price
• Applications to search and rescue and precision agriculture
• Integration of model-based and data-driven methods

AI 1.0  ➔  AI 2.0  ➔  AI 3.0