Enabling Multimodal Sensing, Real-time Onboard Detection and Adaptive Control for Fully Autonomous Unmanned Aerial Systems

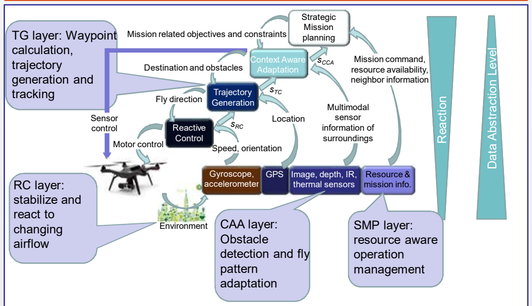
[1]Qinru Qiu, [3]Yanzhi Wang, [1]Senem Velipasalar, [1]Jian Tang, and [2]Amit Sanyal

[1]Department of Electrical Engineering & Computer Science, Syracuse University, NY, USA

[2]Department of Mechanical & Aerospace Engineering, Syracuse University, NY, USA

[3]Department of Electrical and Computer Engineering, Northeastern University, Boston, USA

Four Levels of Autonomy



Challenges and Solutions (1)

- Challenges in solving the control problem
  - Extremely large state and solution space
  - Uncertainty in the environment
  - Delayed reward/penalty
  - Solutions: deep reinforcement learning (DRL)
- Challenges in sensing and detection
  - Multiple sensors with quasi-synchronous reading
    - ✓ Requires synchronization and data fusion capabilities
  - Lack of training data
    - ✓ Most existing CNN are trained using front view of objects
  - Solutions: transfer learning and multi-modal data association

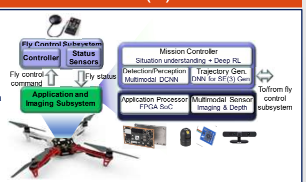
Challenges and Solutions (2)

- Challenges in onboard computing
  - Limited computing power

  Most of the on-board flight controllers use ARM processors
  - Limited battery power

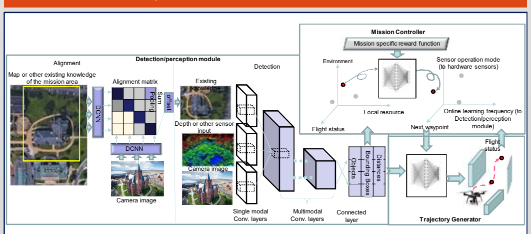
  Typical drone battery ranges from 2000mAh to 5000mAh
  - High complexity of DCNN

  E.g. one forward pass of AlexNet requires ~1.4GFOPS



- Diversified computing model for sensing, detection and control requires general purpose processor
  - Low performance and energy efficiency
- Solutions: Unified computing model and custom designed embedded system based on FPGA,
  - Circulant weight matrix for lower computation and storage complexity
  - FFT based operation for efficient hardware implementation

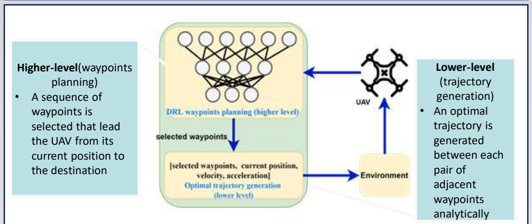
Proposed Framework



Challenges in Trajectory Generation & Control

- Trajectory generation and control for safe and effective UAV operations requires:
  - Obstacle avoidance
  - Stability
  - Energy efficiency
- Multi-rotor unmanned aerial vehicles have high maneuverability in three-dimensional motion
  - Rigid body model imposes challenge in stability
  - Need to consider long distance flight in a complex environment
  - Complex relation between force, torques and UAV aerodynamic status

Proposed Two-level Trajectory Optimization

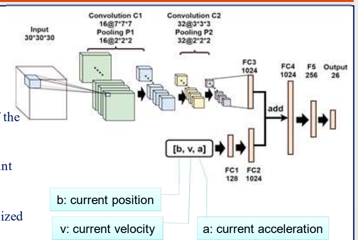


Trajecotry Generation: Problem Formulation

- Discretized Environment
  - $N \times N \times N$ "grids"
- Find the set of waypoints w_i, $0 \le i \le N-1$ that minimizes $\boldsymbol{F} = \sum_{i=0}^{N-2} f(w_i, w_{i+1})$, ($f(w_i, w_j)$→ UAV control thrust cost to follow trajectory between waypoints w_i and w_j) subject to:
  - w_0 is the starting location w_{N-1} is the destination
  - w_i is not an obstacle, $0 \le i \le N-1$
- The minimum energy trajectory between (w_i, w_{i+1}), $0 \le i \le N-2$, does not overlap with obstacles

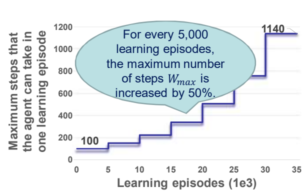
Network Structure of Waypoints Planning

- State:
  - $N \times N \times N$ grids
  - [b, v, a] of the UAV
- Action:
  - A UAV can choose any of the $3 \times 3 \times 3$ grids around its current location as waypoint
- Reward:
  - Position reward + Normalized control thrust cost



Progressive Learning in Controlled Environment

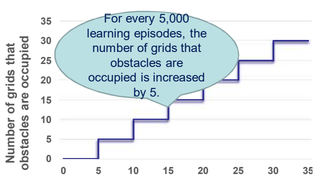
- Progressive learning:
  - Start learning with very low UAV mobility (i.e. the maximum number of steps that the UAV can take) and gradually increase it as the learning processes



Goals:

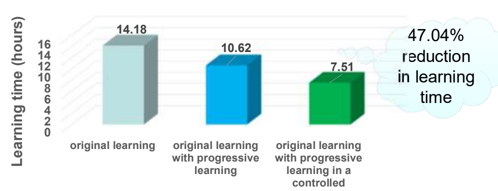
1. The agent can get quick feedback by traveling short distance with limited mobility
2. The agent can gain knowledge about the position reward faster

- Controlled environment
  - To start the learning with a free space and gradually increase the number of obstacles in the environment



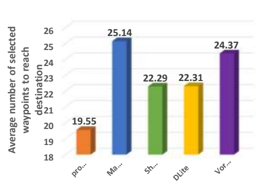
Goals:

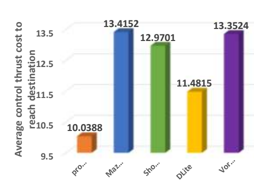
1. The agent can discover basic environment first
2. The agent can learn the interaction between the upper and lower layer optimizer faster
3. The agent can learn how to avoid obstacles more quickly with the knowledge of its capability



Comparison with Existing Methods

- On average 16.67% less number of waypoints, 21.28% control thrust reduction



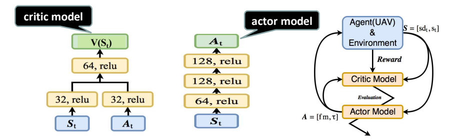


Trajectory Control: Problem Formulation

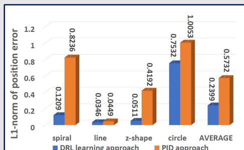
- Target UAV system: Quadrotor fixed-wing UAVs
- Input: Desired trajectory C_d2 generated through a set of predefined waypoints
- Environment state (s):
  - Current UAV status (18 variables) (***Pose***)
  - [position (p), velocity (v), acceleration (a), attitude (R)]
  - Desired UAV status (18 variables) from C_d2 (***Pose***)
  - Desired [position (pd), velocity (vd), acceleration (ad), attitude (Rd)]
- Action (a):
  - One degree of translation motion
  - Three degrees of rotation motion (roll, pitch, yaw)

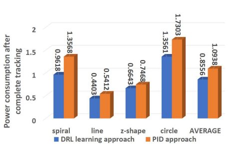
Network Structure

- Objective function: To find the policy π that minimizes $\sum_{i=0}^{t-1}(\boldsymbol{Pose}_{dt} - \boldsymbol{Pose}_t)$
  - Use actor model to predict At
  - Update V(st+1) using received reward, calculate $\boldsymbol{\Delta}$ V(st+1)
  - Back propagate and update the critic model; predict V(st) and V(st+1)
  - Calculate the advantage function $\boldsymbol{\Delta}$ At using received reward, V(st) and V(st+1)
  - Back propagate $\boldsymbol{\Delta}$ At and update the actor model



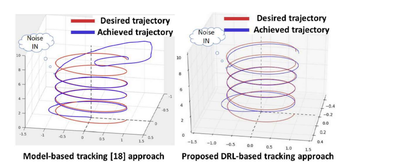
Results of Trajectory Tracking





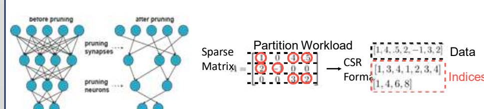
Position Error

Control Thrust



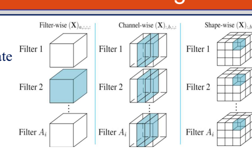
Limitations of Existing DNN Compression

- The non-structured weight pruning – arbitrary weight can be pruned
  - Limited actual deployment
  - Limited weight pruning rate in CONV layers (2.7X for AlexNet)
  - Indices are required for sparse format – speed degradation in GPU/CPU



ADMM Based Structured Pruning

- Incorporating "structures" in DNN weight pruning to facilitate hardware implementations
- Filter-wise, channel-wise, and shape-wide structured sparsity



Structured pruning, with no accuracy loss

Method	Top1 Acc Loss	Statistics	conv1	conv2	conv3	conv4	conv5	conv2-5
SSL [Wen *et al.*, 2016]	0%	column sparsity	0.0%	20.9%	39.7%	39.7%	24.6%	33.3%
		pruning rate	1.0×	1.3×	1.7×	1.7×	1.3×	1.5×
our method	0%	column sparsity	0.0%	70.0%	77.0%	85.0%	81.0%	79.2%
		GPU1×	1.00×	2.27×	3.35×	3.64×	1.04×	2.58×
		GPU2×	1.00×	2.83×	3.92×	4.63×	3.22×	3.65×
		pruning rate	1.0×	3.3×	4.3×	6.7×	5.3×	4.8×

Structured pruning, with 2% accuracy loss

Method	Top1 Acc Loss	Statistics	conv1	conv2	conv3	conv4	conv5	conv2-5
SSL [Wen *et al.*, 2016]	2.0%	column sparsity	0.0%	63.2%	76.9%	84.7%	80.7%	84.4%§
		row sparsity	9.4%	12.9%	40.6%	46.9%	0.0%	
		pruning rate	1.1×	3.2×	7.7×	12.3×	5.2×	6.4×
our method	0.7%	column sparsity	0.0%	63.9%	78.1%	87.0%	84.9%	86.3%§
		row sparsity	9.4%	12.9%	40.6%	46.9%	0.0%	
		CPU×	1.05×	2.82×	6.63×	10.16×	5.00×	6.16×
		GPU1×	1.00×	1.28×	4.31×	1.75×	1.52×	2.29×
		GPU2×	1.00×	2.34×	6.85×	6.99×	4.15×	5.13×
		pruning rate	1.1×	3.1×	7.3×	14.5×	6.6×	7.3×
our method	2.0%	column sparsity	0.0%	87.5%	90.0%	90.5%	90.7%	93.7%§
		row sparsity	9.4%	12.9%	40.6%	46.9%	0.0%	
		CPU×	1.05×	8.00×	14.68×	14.22×	7.71×	11.93×
		GPU1×	1.00×	2.39×	5.34×	1.92×	2.04×	3.15×
		GPU2×	1.00×	4.92×	12.55×	8.39×	6.02×	8.52×
		pruning rate	1.1×	9.2×	16.8×	19.8×	8.4×	15.0×