Intelligent Motion Video Guidance for Unmanned Air System Ground Target Surveillance

John Valasek
Kenton Kirkpatrick
James May

Aerospace Engineering Department
Texas A&M University

Infotech@Aerospace 2012
21 June 2012
Garden Grove, CA
Overview

- Identified Need and Research Objectives
- Learning Agent
- Problem Definition and Representations
- Environment Modeling and Simulation
- Results
- Flight Test
- Extensions
2011 – 2012 Research Team
Identified Need
How It Is Done Now (1)

Payload Vehicle Operator
Air Vehicle Operator
Mission Command / Intelligence “Gatherers”
How It Is Done Now (2)
Various Approaches

- Some recent and not-so-recent approaches for this specific problem
  - Sinipoli et al (2001)
  - Stolle & Rysdyk (2003)
  - Wang et al (2005)
  - Stepanyan and Hovakimyan (2006)
  - Ma et al (2007 x 2)

- Most use some type and amount of gimballing to control the camera, and focus on path planning approaches

- Strap-Down camera, feedback, based guidance without path planning
  - Saunders & Beard (2011)
Research Objectives

1. Identify a Preferred Concept for controlling a UAS and camera
   a. Keep a selected target visible in the camera field of view.
      • Frees a human supervisor to focus on selecting viable targets and analyzing the images received.

2. Develop a methodology for determining a **combined UAS / image sensor control policy using a novel approach**
   a. Guides UAS with a **fixed-mounted image capturing device** to track targets.
   b. Control policy shall track a pre-designated target in the field of view of the image capturing device.
   c. **Type of image capturing device non-specific.**

3. Track stationary and moving targets in wind, independent of road net / features / terrain data
UAS System Types Considered

Steer Vehicle to Orient Video Image
Learning Agent
Inference Mechanism Candidates

- **Rules-Based Expert System**
  - Requires prior expert knowledge

- **Genetic Algorithms**
  - Not state-based domain
  - Does not save information learned between initial and final points

- **Fuzzy Logic**
  - Not state-based domain
  - Requires insights into behaviors that may not be possible

- **Artificial Neural Networks (ANN)**
  - Requires existence of good training actions data

- **Reinforcement Learning (RL)**
  - State-based domain
  - Requires reward information, but not training actions data
  - Model Free Method*

\[ o(x_1, \ldots, x_n) = \begin{cases} 
1 & \text{if } w_0 + w_1x_1 + \ldots + w_nx_n > 0 \\
-1 & \text{otherwise} 
\end{cases} \]
Reinforcement Learning 1

- Sequential decision making
  - Knowledge is based on experience and interaction with the environment, not on input-output data supplied by an external supervisor

- Achieves a specific **goal** by learning from interactions with the environment.
  - Considers **state** information $s \in S$
  - Performs sequences of **actions** $a \in A$, observing the consequences
  - Attempts to maximize **rewards** ($r$) over time $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \sum_{i=0}^{\infty} \gamma^i r_{t+i}$
    - These specify what is to be achieved, **not how to achieve it**

  - Constructs a **state value function** $V(s)$ or **action-value function**, $Q(s,a)$
    - Memory is contained in the state value function, $V^*(s) = \max_\pi V^\pi(s)$, or action-value function, $Q^*(s,a) = \max_\pi Q^\pi(s,a)$

  - Learns a control **policy**, $\pi$, where $\pi : S \rightarrow A$
Reinforcement Learning: Q-Learning

- Off-policy Method
  - Learned action-value function, $Q$, directly approximates the optimal solution, $Q^*$
    - Independent of the policy being followed
      \[
      Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]
      \]
  - Policy determines state-action pairs

- Mathematically proven convergence to optimal policy

\[
Q(s_t, a_t) \to Q^*(s_t, a_t) \text{ as } N_{\text{visits}} (s_t, a_t) \to \infty
\]

but

\[
Q^\pi(s_t, a_t) \approx Q^*(s_t, a_t) \text{ if } N_{\text{visits}} (s_t, a_t) < \infty \text{ for } N_{\text{visits}} \text{ "large"}
\]

Watkins 1989
Problem Definition and Representation
Problems Considered

- Stationary target with pan angle
  - Determine “best” pan angle for fixed camera

- Stationary target (Case 1)

- Moving target with pan angle
  - Linear, constant speed, determine “best” pan angle for fixed camera

- Moving target (Case 2)
  - Linear, constant speed

- Stationary target in wind (Case 3)
  - Constant wind at
Tracking as an RL Problem

- **States** of System ($s$)
  - **Target** x-position in image frame
  - **Target** y-position in image frame
  - UAS bank angle
  - $s = [X \ Y \ \phi]^T$
Tracking as an RL Problem

- **Goal** of Learner ($g$)
  - Move *target* to center of image
  - Once *target* reaches goal state, hold it there
  - Only *target* x-position and y-position considered for goal achievement
  - $g = [0 \ 0 \ \phi]^T$
Tracking as an RL Problem

**Action Selection Rationale**
- No control over target global position.
- **Only way to track target in image frame is to steer the UAS itself.**
- Bank angle is the UAS state that is controllable and has the greatest effect on target image position.
- To lower number of state-action pairs to explore, change in commanded bank angle is used rather than simply commanded bank angle.

**Actions \((a)\)**
- -2 degrees bank angle
- +0 degrees bank angle (need to be able to hold current bank angle)
- +2 degrees bank angle \[ a_2: \Delta \varphi = 0^\circ \]
- \[ a = \begin{bmatrix} -2 & 0 & 2 \end{bmatrix}^T \]
- \[ a_1: \Delta \varphi = -2^\circ \]
- \[ a_3: \Delta \varphi = 2^\circ \]
Tracking as an RL Problem

- **Rewards** \( (r) \)
  - Rewards given to the RL agent are used to update the Q-matrix.
  - Q-Matrix dimensions are \((s \times a)\)
    - Current discretization = maximum dimensionality of \((114,688 \times 3)\) for stationary targets.

- Reward structure for UAS Tracking problem
  - Target reaching center of image \((r = +20)\)
  - Target hitting image boundary \((r = -5)\)
  - Target leaving image frame \((r = -20)\)
  - Neutral reward for every other situation \((r = 0)\)
Episodic Learning

- **Episode length and number of episodes are design parameters that must be chosen to make sense for the particular problem**

- **End of an episode is defined by 2 possible conditions:**
  - 500 actions performed (due to time step of 1 sec per action, this is 500 sec)
  - Target leaving the image frame (breaching state constraints)

- **The initialization of the next episode can be done by:**
  - Continuing from the state where the last episode ended
  - Initialize the system to a specified state
  - Initialize the system to a random state (within the bounds)
Environment Modeling and Simulation
Simulation

- **Aircraft**
  - Kinematic Model
    - Position and orientation only
      - x, y, z
      - airspeed
      - φ, θ, ψ
    - Important aircraft specifications
      - Cruising speed
      - Operating altitude

- **Camera**
  - roll, tilt, pan
  - Aspect ratio
  - Zoom
    - FOV angle
Simulation: Assumptions

- Constant radius steady, level, turns
- Discrete jumps in $\phi$
  - Action timespan $>>$ time required to reach commanded $\phi$
- Constant altitude
- Constant velocity
- Fixed camera pose

Balance of Forces

\[
L \cos \phi = W = mg
\]
\[
L \sin \phi = a_n = m \psi^2 R_i = m \psi U_1
\]
\[
\tan \phi = \frac{m \psi U_1}{mg} = \frac{\dot{\psi} U_1}{g} \rightarrow \psi = g \tan \phi \approx g \frac{\phi}{U_1}
\]

Pan

Tilt
Simulation

\[ d = \frac{(X_{\text{img}} - X_{\text{target}}) \cdot \hat{n}_{\text{img}}}{(X_{\text{camera}} - X_{\text{target}}) \cdot \hat{n}_{\text{img}}} \]

- **Global simulation space**
  - Aircraft pose
  - Target pose

- **Learning space**
  - Target position in image frame
  - \( \varphi \)
Case 1: Stationary Target

- Camera tilt at -20° from left wing
- Altitude = 152 m
- Target Speed = 0 m/s
- Cruise Speed = 27 m/s
- Reflects 10M learning episodes
Case 1: Stationary Target
Case 2: Linear Moving Target

- Target moves in straight line
- Camera tilt at -20° from left wing
- Altitude = 152 m
- Target Speed = 27 m/s
- Cruise Speed = 27 m/s
- Reflects 5M learning episodes
Case 2: Linear Moving Target
Wind Effects

- Wind acts as a disturbance that affects guidance solutions and must be accounted for
  - Pushes target out of the sensor field of view
  - Imposes additional kinematic constraints, solutions are harder

- Constant target motion and constant wind have the same effect on relative dynamics
  - Stationary target & const. wind represent both moving

- Recent approaches
  - Thomasson (1998)
  - Rysdyk (2006)
  - Hamel & Mahoney (2007)
  - Saunders & Beard (2011)
Accounting for Wind

- Incorporating wind in state-space
  - Wind is described by 2 parameters:
    - Wind Speed, $v_w$
    - Wind Direction (or Wind Heading Angle), $\psi_w$
  - Assumptions:
    - Wind speed and direction are known
    - Wind is approximately constant during the short testing period
  - Wind speed and direction are randomly initialized at start of each episode
  - New learning state-space must include wind characteristics:

  $$ \mathbf{s} = [X \ Y \ \phi \ v_w \ \psi_w]^T $$
Case 3: Stationary Target w/ Wind

- Camera tilt at -20° from left wing
- Altitude = 152 m
- Target Speed = 0 m/s
- Cruise Speed = 27 m/s
- Reflects 1M learning episodes
- Wind for this test case:
  - 13 mph
  - 45° from global x-axis
Case 3: Stationary Target w/ Wind
What The Test Cases Show

- Closed-loop tracking control laws, including gains, were developed using an episodic learning process.

- Controller is improved by shaping rewards, and increasing the number of learning episodes.

- Proper representation of actions and rewards is essential for good results.

- Posing the problem to minimize the number of states and actions leads to faster and more efficient learning, faster and more efficient operation.

- The learning agent can account for wind if it is included in the state-space during the learning process.
Test & Evaluation
Wii Remote Infrared Sensor

- Characteristics
  - Manufacturer: PixArt Imaging
  - Resolution: 128x96
  - Field of View: ±33° H x ±23° V (determined experimentally)
  - Spectrum: 940nm IR
  - Refresh Rate: 100 Hz
  - Multi-Object Tracking engine
    - 8x subpixel analysis
    - 1024x768 virtual resolution
    - On-line configurable sensitivity
    - Blob tracking for up to four points.
  - Interface: 400kHz Fast I²C
ConSInt Testbed
Pegasus

- **Features:**
  - Large fuselage internal volume
  - Variable static stability
  - Modular construction

- **Geometry:**
  - Wing Span: 12 ft
  - Wing Area: 18 ft²
  - Length: 10.6 ft

- **Weight:**
  - Empty: 30 lb
  - Maximum Take-Off: 52 lb
  - **Payload:** 20 lb
  - Fuel: 2 lb

- **Performance:**
  - Maximum Speed: 87 knots
  - **Stall Speed (MTOW):** 26 knots
  - Endurance: 1 hr

Design Airframe Life
100 cycles minimum
Summary

- Method synthesizes control laws for complex interaction, hard to model, possibly poorly understood systems.

- Good for systems where prior engineering knowledge or training data does not exist.

- “Model-free” approach
  - Model not used in synthesizing control laws
  - Model does not appear in the control laws
  - Vehicle/environmental models needed for learning via simulation

- Matured controller has “some” ability to work on different systems
  - Changes in the original system
  - Similar but different system

- Learning is lifelong
  - The more the controller is used, the “better” it gets
Conclusions (1)

- Algorithm learning
  - Static case convergence with fixed camera is rapid
    - Low number of state-action pairs.
    - Number of learning episodes required to converge to a “good” solution varies with the particular case/scenario being learned,
    - All results show a point of diminishing returns on learning episodes.

- Camera installation and orientation
  - Initial position of target relative to initial position of aircraft strongly influences results.
  - Controller usually attempts to drive target into second quadrant
    - Allows aircraft to turn ahead of target, keep in image frame in the future.
  - Right hand side camera orientation steers target to first quadrant.
Conclusions (2)

- **Stationary target cases**
  - For feasible initial conditions, controller keeps target in the image frame for simulation duration.
  - Broader tracking goal of keeping target in image frame for a “useful” period of time is generally met.

- **Moving target cases**
  - show that this method has promise for learning to track a moving target, and merits further investigation.

- **Stationary target with wind cases**
  - Using wind measurements in state-space shows promise, merits further investigation on moving target cases.
Future Work

- **Fidelity**
  - Improved Image Data & Realistic Camera Characteristics
  - Variable Discretization: Adaptive Action Grid (AAG)

- **Capabilities**
  - Moving Target – Randomized Motion
    - Increases learning state space
    - Complex target movements require more numerous or complex states
  - Momentarily Obscured Target – Tunnels, etc.
  - Multiple Targets – Switching Between Targets On The Fly
  - Non-planar dynamics with velocity as an action (control)
  - Incorporate wind adjustments into all future scenarios
Future Work

- Greater Control Authority
  - Gimbaled Camera
    - Allows for partial decoupling of tracking with UAS banking
    - Increases action space and learning the state-space
  - Increase UAS control inputs from RL agent
    - Altitude changes
    - Velocity changes
    - Allow for tracking of more complex moving target trajectories

- Flight Test
  - Track a stationary target using actual Pegasus-class UAS
  - Track a moving target using actual Pegasus-class UAS

Acknowledgements

This work was sponsored (in part) by the

- Raytheon Intelligence and Information Systems Division under contract C10-00904.
  - Technical Monitor: Michael Moan

  - Technical Monitor: Dr. Fariba Fahroo

Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Raytheon or the U.S. Air Force.
Point of Contact

- Director, Vehicle Systems & Control Laboratory
  John Valasek
  Aerospace Engineering Department
  Texas A&M University
  3141 TAMU
  College Station, TX 77843-3141

  (979) 845-1685
  valasek@tamu.edu

- Web Page
  - [http://vscl.tamu.edu/valasek](http://vscl.tamu.edu/valasek)