WHERE TO LAND
A Reachability Based Forced Landing Algorithm for Aircraft Engine Out Scenarios

Jinu Idicula - NASA Armstrong
Kene Akametalu, Mo Chen, Claire Tomlin, Jerry Ding - UC Berkeley
Loyd Hook - UTulsa
11/17/2015
Outline

1. Where To Land (WTL)
2. WTL1 → WTL2
3. Engine Out Case
4. Aircraft Reachability
5. Cost Map Development
6. Dynamics Model
7. NASA TCM Model
8. Optimal Trajectory Generation
9. WTL2 C code
10. Test Cases
11. Hardware in the Loop (HIL) Simulation
12. Future Work
## WTL Team

| UC Berkeley       | Algorithm Design  
|                  | Reachable Sets    
|                  | Hybrid Mode Switching |
| NASA Armstrong    | WTL C Code        
|                  | S/W V&V           
|                  | HIL Simulation    |
| U. Tulsa          | NYC Cost Map      
|                  | S/W Requirements  |
Emergency Landings
Where To Land

- **Where To Land (WTL)** is an emergency forced landing algorithm developed by UC Berkeley.

- Inflight emergency → vehicle forced to land
  - What is the **optimal landing location** that will **minimize loss of life** and **minimize property damage** given a **set of constraints**?
  - What is the optimal trajectory required for the aerial vehicle to reach the optimal landing location?

- WTL attempts to mimic an expert pilot’s decision making and land the aircraft.

![WTL Concept of Operations](image)
WTL Algorithm

Pre-Planning - pre-compute trajectories using fault location, maps and reachable sets

Real Time Update – adapt emergency trajectory based on real time data (weather, occupancy, etc.)
Innovation

Prior Forced Landing Algorithms

- Simple dynamics model
- Assumes aircraft can return to runway
- Difficult to apply to autonomous vehicles
- Haven’t been flight tested

Where to Land Algorithm

- Provides safety guarantees for S/W V&V
- Higher fidelity aircraft model
- Fast computation
- Manned or unmanned vehicles
- Modular design
WTL1 Phase 1 Results

1. UC Berkeley Campus
   - No Fly Zone

2. Emergency Trajectories
   - Demo: MATLAB sim
   - Location: UC Berkeley
   - Vehicle: Quadrotor
   - Failure: 90% thrust

3. 2D Trajectory
   - Start Location
   - Final Location
   - Emergency Landing Location
Phase 1 \(\rightarrow\) Phase 2

- Reduce the scope of WTL
  - Simplify WTL \(\rightarrow\) Speed up software development
  - Find “real world” design/implementation issues
  - Get pilot feedback with HIL simulation
  - Collect data to improve future versions

- WTL1 \(\rightarrow\) WTL2
  - NASA TCM/B-757 aerodynamics model
  - No real time update \(\rightarrow\) compute trajectories during fault
  - No global cost map \(\rightarrow\) NYC/New Jersey area \(~100+\) miles
  - No Fault detection \(\rightarrow\) One predefined fault, dual engine failure
  - HIL 6DOF nonlinear aircraft simulation

---

**PHASE 2 GOALS**

- Demonstrate WTL in HIL simulation
- Develop tools to generate reachable trajectories
WTL Development Plan

**Phase 1 – WTL1**
- Demo: MATLAB Sim
- Location: UC Berkeley
- Vehicle: Quadrotor
- Failure: 90% reduction in thrust
- 2D Trajectory

**Phase 2 – WTL2**
- Demo: HIL Sim w/ FLS on embedded H/W
- Location: New York City +/- ~100 miles
- Vehicle: 757
- Failure: Loss of thrust
- 2D Trajectory

**Future Work**
- Demo: Flight test RC Aircraft w/ Pixhawk
- Location: Edwards, CA
- Vehicle: RC Aircraft
- Failure: Loss of thrust
- 2D Trajectory
WTL2 Architecture

Reachable Set → Cost Map → Path Planner

Aerodynamics → Aircraft Glide Equations

Inputs: Velocity, Weight, Position, Heading

Outputs: Waypoints, Glide Speed
WTL2 Algorithm

1. Get current aircraft state
   – Latitude/Longitude
   – Altitude/Heading/Velocity
2. Convert states to local frame
3. Compute maximum glide range
4. Window cost map with max range
5. Get reachable set for altitude
6. Scale and project reachable set over map with heading
7. Find best reachable landing location using 2D convolution
8. Generate trajectory using optimal path planner
9. Generate latitude/longitude waypoints
10. Generate target headings
HIL Simulation Architecture

- Nonlinear Aircraft Simulation
- WTL2
  - Embedded Linux Board
- HIL Simulation PC
- Display
- Pilot Controls

TCP/IP Communication
Engine Out Scenario

• Complete loss of thrust

• Engine out during takeoff is the most critical
  – WTL2 Operational Range: 1000 ft – 4000 ft
  – Less than 1000 ft → Can only land straight ahead
  – Greater than 4000 ft → Can often return to airport
  – Glide range will vary based on aircraft and configuration (i.e. weight, flaps)

• During failure → pilots must manage energy

• Flying at $L/D_{\text{MAX}}$ maximizes aircraft range

• $L/D_{\text{MAX}} \rightarrow \alpha_{\text{MAX}} \rightarrow$ gross weight $\rightarrow V_{\text{GLIDE}}$

• Flying at $V_{\text{GLIDE}}$ will maximize aircraft range
Reachability

Reachability - Given a dynamic system governed by some differential equation and input defined over some bounded state space. What are all the states visited by the trajectories of the system

- Reachability is a key technology for verifying safety critical systems
- Reachability assures that a system can reach a target state while remaining within a safety envelope
- Level Set Toolbox - computes reachable sets of hybrid systems with continuous dynamics using nonlinear ODE’s
- Grid based computation
Aircraft Reachability

Aircraft Reachability is gliding aircraft model with NASA TCM aerodynamics formulated as a PDE (HJ) and solved using the Level Set Toolbox. Aircraft trajectory has two modes. The two mode states are stitched together using a hybrid system model.

Mode 1 - Approach Mode
- TCM aerodynamics
- Glide equations
- Glide velocity
- Constant radius turns
- State constraints

Mode 2 – Landing Mode
- TCM 30° flap aerodynamics
- Landing velocity
- State constraints

States
- Aircraft position
- Velocity
- Flight path and heading angles

Control
- Angle of attack
- Bank angle
**Reachable Set**

Reachable sets are a set of initial states from which the system is guaranteed to remain inside a safe region while eventually reaching a desired target.

**State Constraints**

- $V$ – Stall avoidance
- $\alpha, \phi$ – Keeps aircraft within performance envelope
- Acceleration - structural load limits
Discrete Reachable Sets

- Reachable sets generated every 100 ft from 1000 ft - 4000 ft
- Grid size 10E4x10E4 ft
- Normalized and stored as a binary map
- Oriented onto global map using aircraft heading
Cost Map

- Hazard Map – constructed from population and geographical data
- Impact Map – constructed from density maps, land use maps, etc.
- Total Loss Map = Hazard Map + Impact Map
- Map Size: 7201x5401 pixels (3.5+ million pixels)
Gliding Aircraft Equations

- 3D motion of gliding aircraft over flat Earth
- Model assumes coordinated turns, no sideslip

\[
\begin{align*}
\dot{X} &= V \cos \gamma \cos \xi \\
\dot{Y} &= V \cos \gamma \sin \xi \\
\dot{Z} &= V \sin \gamma \\
\dot{V} &= -\frac{D(\alpha, V)}{m} - g \sin \gamma \\
\dot{\gamma} &= \frac{L(\alpha, V) \cos \phi}{mV} - \frac{g}{V} \cos \gamma \\
\dot{\xi} &= \frac{L(\alpha, V) \sin \phi}{mV \cos \gamma}
\end{align*}
\]

- Aircraft velocities
- Aircraft acceleration
- Flight path derivative
- Heading derivative

\[V_{\text{glide}} = \sqrt{\frac{2W}{\rho S \sqrt{C_D^2 + C_L^2}}}\]

- Optimum glide velocity
NASA TCM Model

- Nonlinear aircraft model developed by NASA Langley for NASA's Aviation Safety Program
- Transport Class Model (TCM) closely replicates B-757 aerodynamics
- For WTL2, TCM aerodynamics tables ($C_L$, $C_D$) are used
- On landing transition to 30° Flap aerodynamics
- Compute $L/D_{\text{MAX}}$ and $\alpha_{\text{MAX}}$
Optimal Landing Location

- Landing footprint is based on aircraft ground roll and impact area
- Optimal landing location = smallest total sum cost over landing footprint
- Found using 2D Convolution with FFT

### Diagram

- **Reachable Node**: ○
- **Searched Node**: ●
- **Landing Footprint**: ▶
Optimal Trajectory Generation

- Dubins trajectory – gives shortest path between two points
  - requires final location and final heading
  - target heading here is the heading required to reach final landing location

- Two basic maneuvers
  - Gliding (maximize glide range)
  - Turning (final orientation)

- Optimal turn radius – minimize energy loss with a constant radius turn
WTL2 C Code

• Dependencies
  – GSL (Numerical Library)
  – GDAL (GIS Library)

• Makefile
  – generates executable for ARM, x86 processors
  – ccompcert → safety critical C compiler

• V&V
  – Use JPL Flight S/W Best Practices (JPL DOCID D-60411)
  – Run code coverage tool
  – Memory debugging tool
  – Unit tests for critical functions
  – Test Cases
# Test Cases

<table>
<thead>
<tr>
<th>Test #</th>
<th>Altitude (ft)</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Initial Heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>40.70°</td>
<td>-73.8726°</td>
<td>270°</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>40.70°</td>
<td>-73.8726°</td>
<td>15°</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>40.85°</td>
<td>-73.70°</td>
<td>270°</td>
</tr>
<tr>
<td>4</td>
<td>4000</td>
<td>40.70°</td>
<td>-73.8726°</td>
<td>270°</td>
</tr>
<tr>
<td>5</td>
<td>4000</td>
<td>40.70°</td>
<td>-73.8726°</td>
<td>15°</td>
</tr>
<tr>
<td>6</td>
<td>4000</td>
<td>40.85°</td>
<td>-73.70°</td>
<td>270°</td>
</tr>
<tr>
<td>7</td>
<td>4000</td>
<td>40.85°</td>
<td>-73.70°</td>
<td>15°</td>
</tr>
<tr>
<td>8</td>
<td>3026</td>
<td>40.865°</td>
<td>-73.88°</td>
<td>220</td>
</tr>
</tbody>
</table>

- Altitude variation – Bounded by two altitudes
  - Altitude < 1000 ft → Can only land straight ahead
  - Altitude > 4000 ft → Should be able to return to airport
- Heading variation – Show effects of initial heading on trajectory
- Position variation – Show effects of initial position on trajectory
- Case #8 replicates US Airways 1549 failure
Results: Test Case 1

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>1000 ft</td>
</tr>
<tr>
<td>Heading</td>
<td>270°</td>
</tr>
<tr>
<td>Latitude</td>
<td>40.7000°</td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.8726°</td>
</tr>
</tbody>
</table>

Cost:
- Reachable location: 0
- Current location: +
- Landing location: ●
- Landing trajectory: ——

Cost Table:

<table>
<thead>
<tr>
<th>Cost</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
Results: Test Case 2

<table>
<thead>
<tr>
<th>Altitude</th>
<th>1000 ft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading</td>
<td>15°</td>
</tr>
<tr>
<td>Latitude</td>
<td>40.7000°</td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.8726°</td>
</tr>
</tbody>
</table>

Reachable location

Current location

Landing location

Landing trajectory

Cost

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.5</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
Results: Test Case 3

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>1000 ft</td>
</tr>
<tr>
<td>Heading</td>
<td>270°</td>
</tr>
<tr>
<td>Latitude</td>
<td>40.85°</td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.70°</td>
</tr>
</tbody>
</table>

Cost:
- 1.0
- 0.5
- 0.0

Reachable location
Current location
Landing location
Landing trajectory
## Results: Test Case 4

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>4000 ft</td>
<td></td>
</tr>
<tr>
<td>Heading</td>
<td>270°</td>
<td></td>
</tr>
<tr>
<td>Latitude</td>
<td>40.7000°</td>
<td></td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.8726°</td>
<td></td>
</tr>
</tbody>
</table>

![Map showing land at [980,1418]](image_url)

- **0**: Reachable location
- **+**: Current location
- **Landing location**: Green square
- **Landing trajectory**: Red line

### Cost

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Results: Test Case 5

<table>
<thead>
<tr>
<th>Altitude</th>
<th>4000 ft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading</td>
<td>15°</td>
</tr>
<tr>
<td>Latitude</td>
<td>40.7000°</td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.8726°</td>
</tr>
</tbody>
</table>

- **0**: Reachable location
- **+**: Current location
- ****: Landing location
- **----**: Landing trajectory

Cost Table:

<table>
<thead>
<tr>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>
Results: Test Case 6

<table>
<thead>
<tr>
<th>Altitude</th>
<th>4000 ft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading</td>
<td>270°</td>
</tr>
<tr>
<td>Latitude</td>
<td>40.85°</td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.70°</td>
</tr>
</tbody>
</table>

Reachable location

Current location

Landing location

Landing trajectory

Cost

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>
Results: Test Case 7

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altitude</td>
<td>4000 ft</td>
</tr>
<tr>
<td>Heading</td>
<td>15°</td>
</tr>
<tr>
<td>Latitude</td>
<td>40.85°</td>
</tr>
<tr>
<td>Longitude</td>
<td>-73.70°</td>
</tr>
</tbody>
</table>

Cost:
- Reachable location: 1.0
- Current location: 0.5
- Landing location: 0.0
- Landing trajectory:
Results: Test Case 8

Altitude: 3026 ft
Heading: 220°
Latitude: 40.86679°
Longitude: -73.9298°

US Airways 1549

Cost:
- 1.0
- 0.5
- 0.0
US Airways Flight 1549
January 15, 2009

Teterboro Airport

48th street

3:31 pm water landing into Hudson River

New York

Queens

US Airways 1549 ground track

NTSB accident reconstruction on Airbus simulator

3:26 pm takeoff from LaGuardia Airport
150 passengers
5 crew members

Longitude (deg)

Latitude (deg)

Altitude (ft)

40.88 40.84 40.82 40.8 40.78 40.76 40.74 40.72

0 500 1000 1500 2000 2500 3000

-74 -73.95 -73.9 -73.85 -73.8 -73.75 -73.7 -73.65 -73.6

40.75 40.8 40.85 40.9 40.95

Longitude (deg)

Latitude (deg)

Altitude (ft)
WTL2 HIL Simulation

<table>
<thead>
<tr>
<th>HIL Simulation Data Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTL State</td>
</tr>
<tr>
<td>Target V (kts)</td>
</tr>
<tr>
<td>Target Heading</td>
</tr>
</tbody>
</table>
| Waypoint #                 | ///</>

www.nasa.gov
Future Work

- “Online” WTL → Fast Estimator/Online Reachable Set
- “Adaptive” WTL → Dynamic trajectories
- WTL on Smartphones, Linux, PixHawk
- WTL + RTA (Run Time Assurance) framework
- WTL + Backward Reachable Controllers
Impact

• **General Aviation**
  – Pilots tend to less experienced
  – Mostly single engine aircraft

• **Commercial**
  – Pilots are experienced and well trained
  – Multi engine aircraft

• **Unmanned Vehicles**
  – Flight Termination Systems
  – Lost Link Mode

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Aviation</strong></td>
<td>Can improve odds of survival</td>
</tr>
<tr>
<td><strong>Commercial</strong></td>
<td>Gives pilots more options</td>
</tr>
<tr>
<td><strong>Unmanned Vehicles</strong></td>
<td>Can enable expanded UAS in the NAS</td>
</tr>
</tbody>
</table>
Distribution

1. WTL Design: AIAA Conference Paper
2. WTL2 Implementation: AIAA Conference Paper
3. WTL2 NASA Technical Memo
4. NASA NARI Presentation
References

3. Ding, J., Gillua, H., Huang, H., “Hybrid Systems in Robotics”
5. Rogers, D., “The Possible ‘Impossible’ Turn”
9. Hueschen, R., “Development of the Transport Class Model Aircraft Simulation from a Sub-Scale Generic Transport Model Model Simulation”