NARI
NASA AERONAUTICS RESEARCH INSTITUTE

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

NASA Armstrong Flight Research Center
Controls & Dynamics
jinu.t.idicula@nasa.gov

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

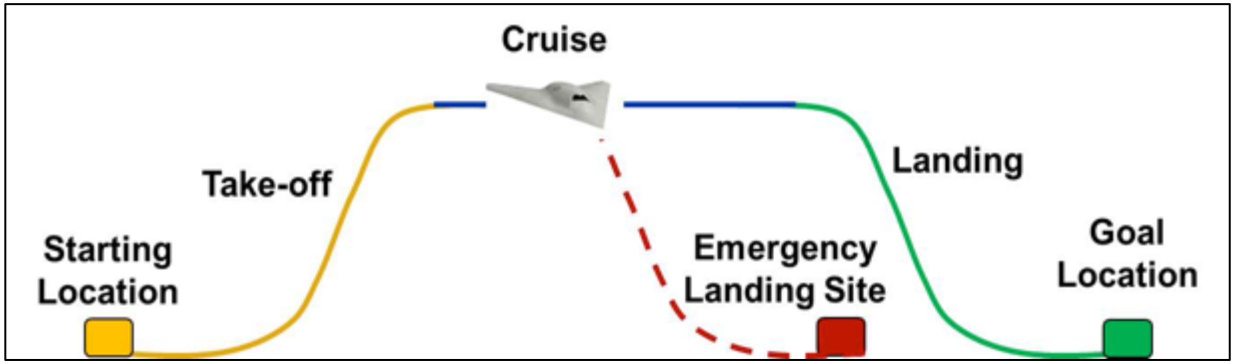
WTL Team

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

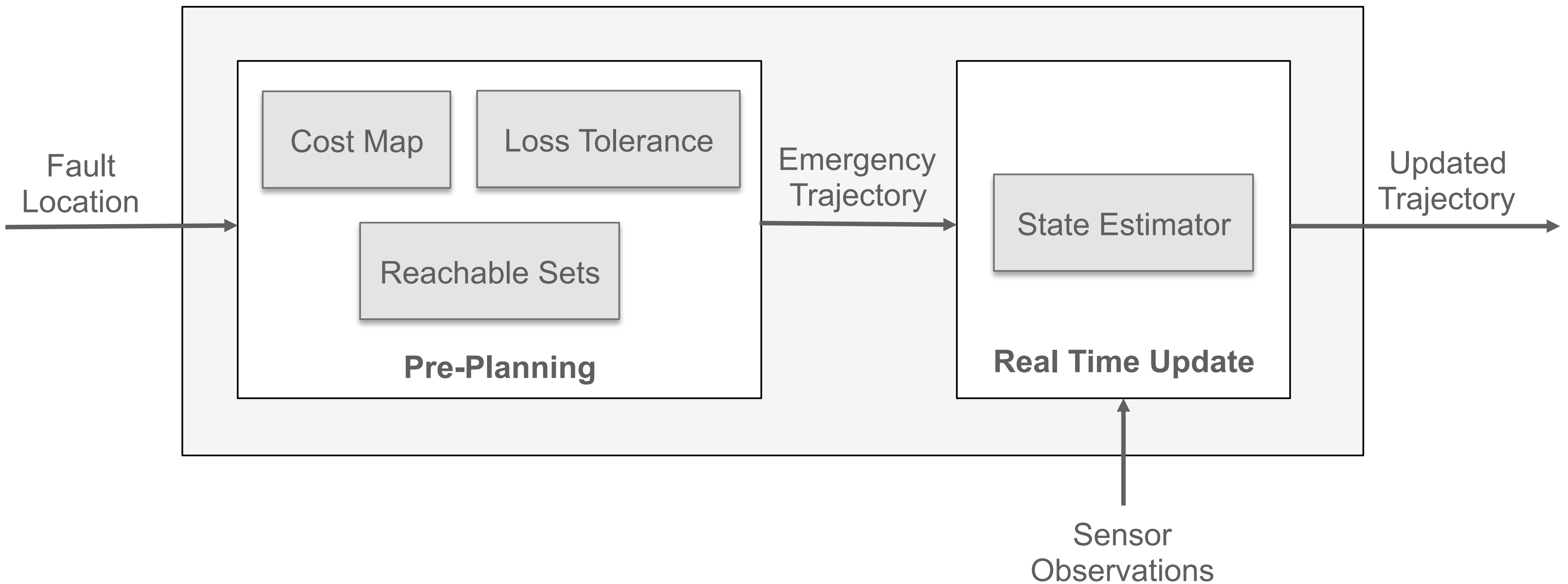
Where To Land

- **Where To Land (WTL)** is a 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 optimal landing location?
- WTL attempts to mimic an expert pilot's decision making and land the aircraft



WTL Concept of Operations

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

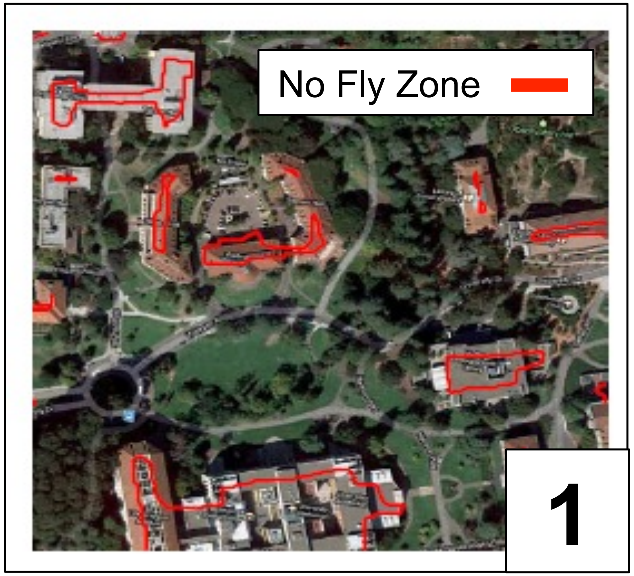
Prior Forced Landing Algorithms

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

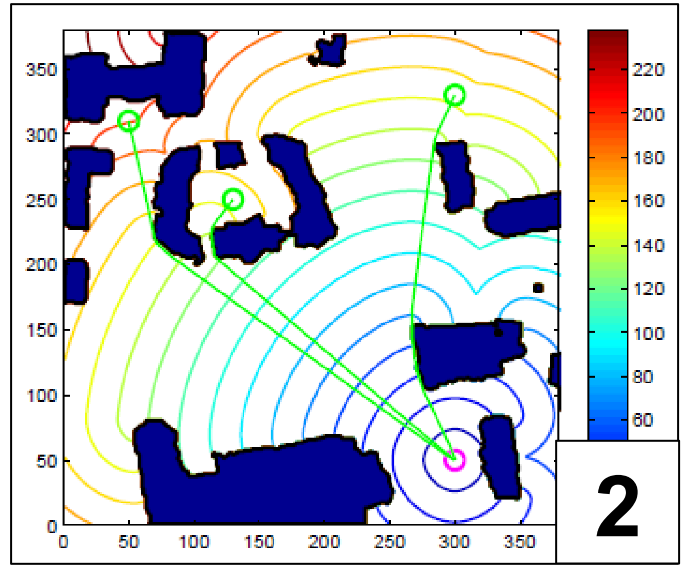
Where to Land Algorithm

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

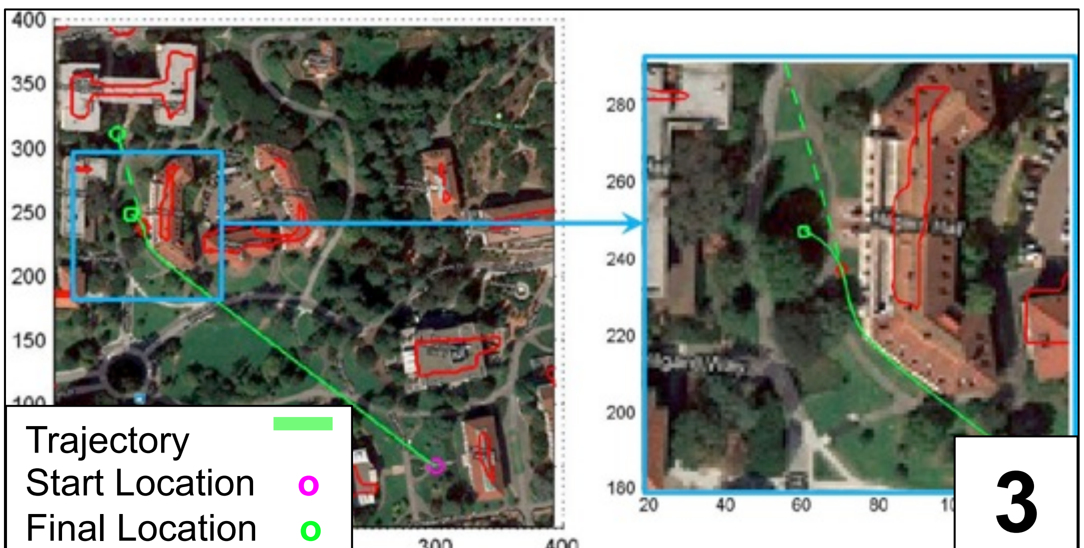
WTL1 Phase 1 Results



UC Berkeley Campus



Emergency Trajectories



Emergency Landing Location

Demo: MATLAB sim
Location: UC Berkeley
Vehicle: Quadrotor
Failure: 90% thrust
2D Trajectory

Phase 1 → Phase 2

- **Reduce the scope of WTL**
  - Simplify WTL → Speed up software development
  - Find "real world" design/implementation issues
  - Get pilot feedback with HIL simulation
  - Collect data to improve future versions
- **WTL1 → WTL2**
  - NASA TCM/B-757 aerodynamics model
  - No real time update → compute trajectories during fault
  - No global cost map → NYC/New Jersey area ~100+ miles
  - No Fault detection → 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

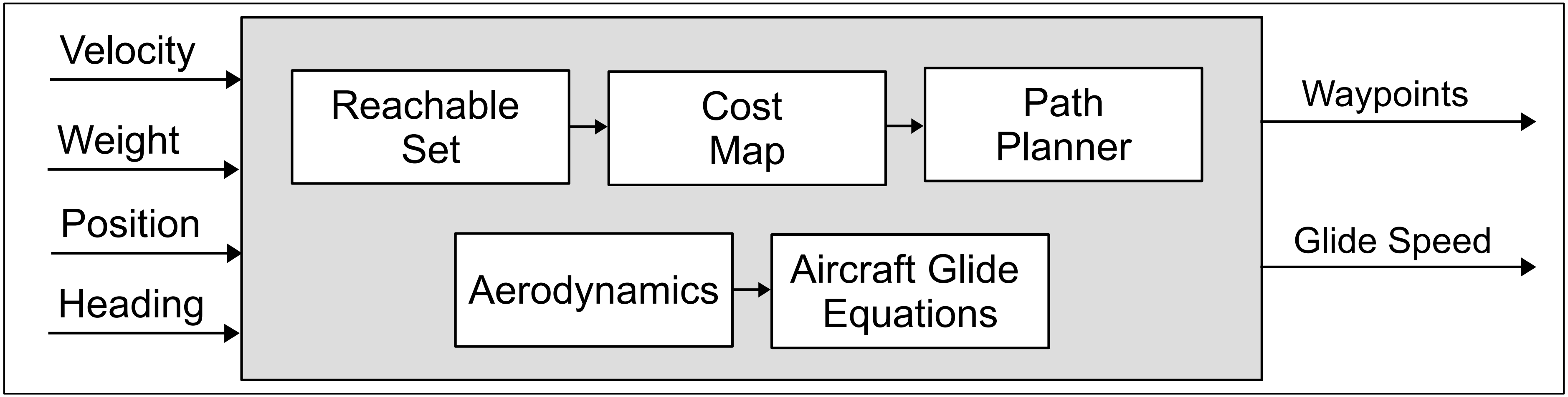
Phase 2 – WTL2

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

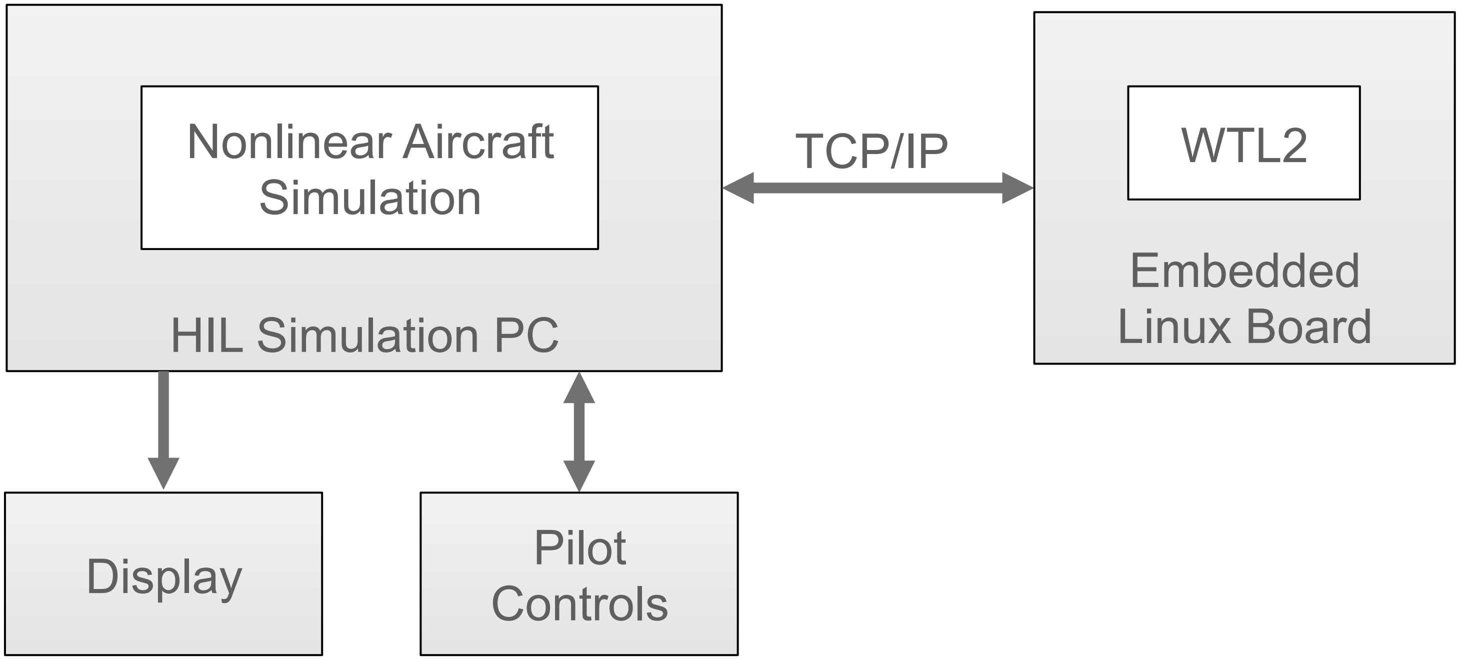
WTL2 Architecture



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



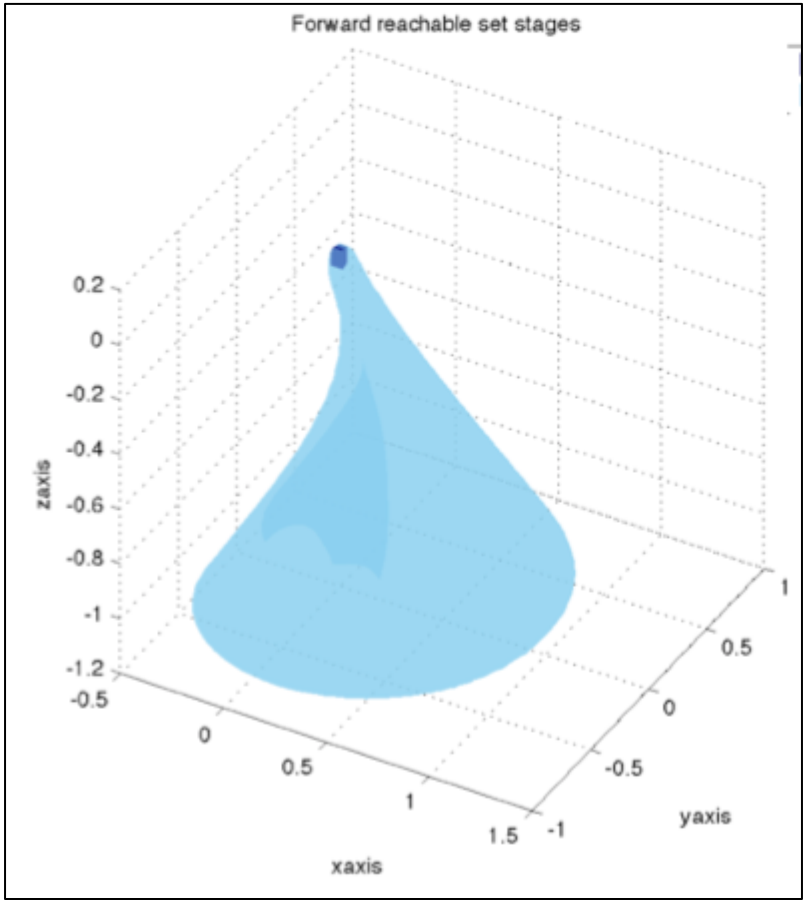
Engine Out Scenario

- 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_{MAX} maximizes aircraft range
- L/D_{MAX} → α_{MAX} → gross weight → V_{GLIDE}
- Flying at V_{GLIDE} will maximize aircraft range

Reachability

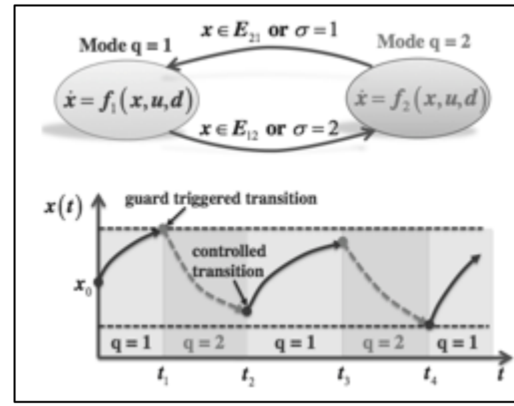
Reachability - Given a dynamic system governed by some differential equation and input defined over some bounded state space. What are all the states visited by the trajectories of the system

- Reachability is a key technology for verifying safety critical systems[7]
- Reachability assures that a system can reach a target state while remaining within a safety envelope[7]
- Level Set Toolbox - computes reachable sets of hybrid systems with continuous dynamics using nonlinear ODE's[3]
- 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

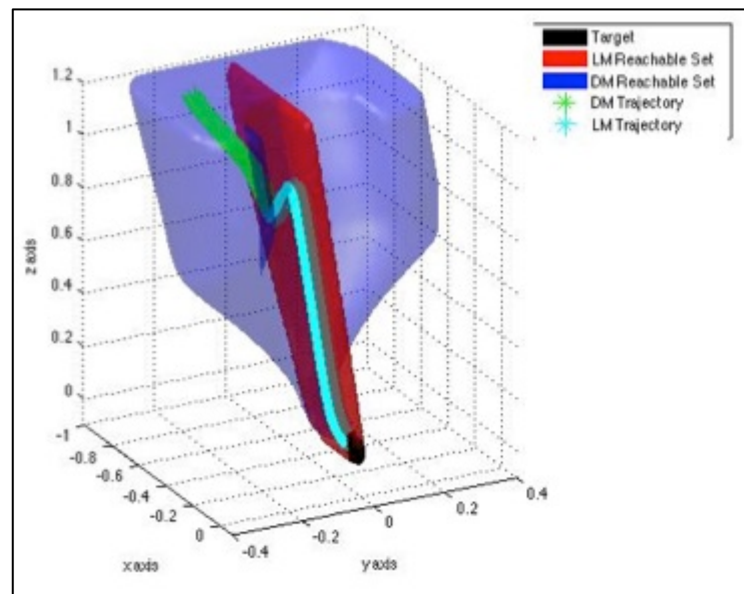
- TCM 30° flap aerodynamics
- Landing velocity
- State constraints

States

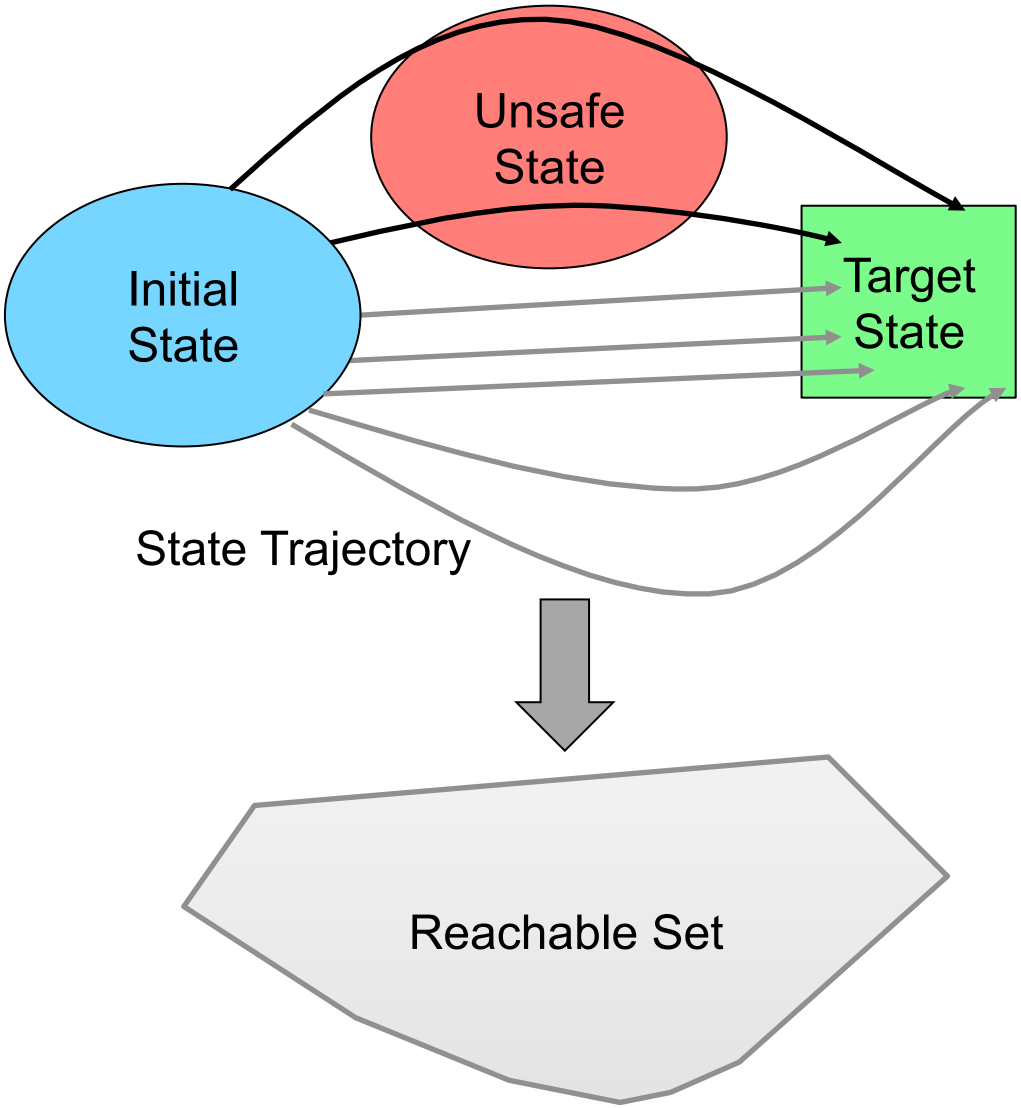
- Aircraft position
- Velocity
- Flight path and heading angles

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[3]

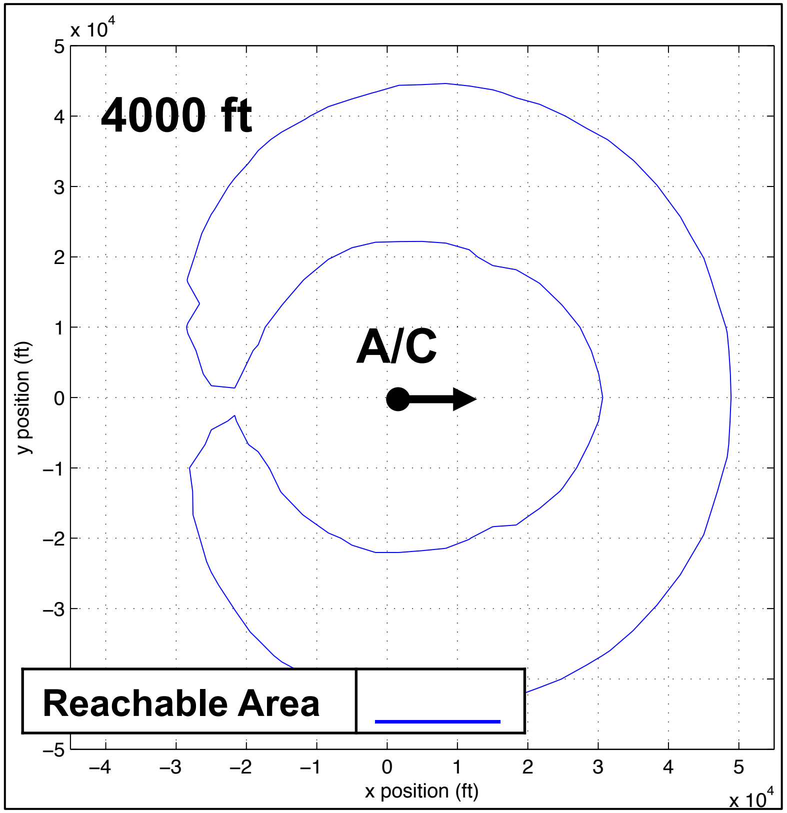
State Constraints

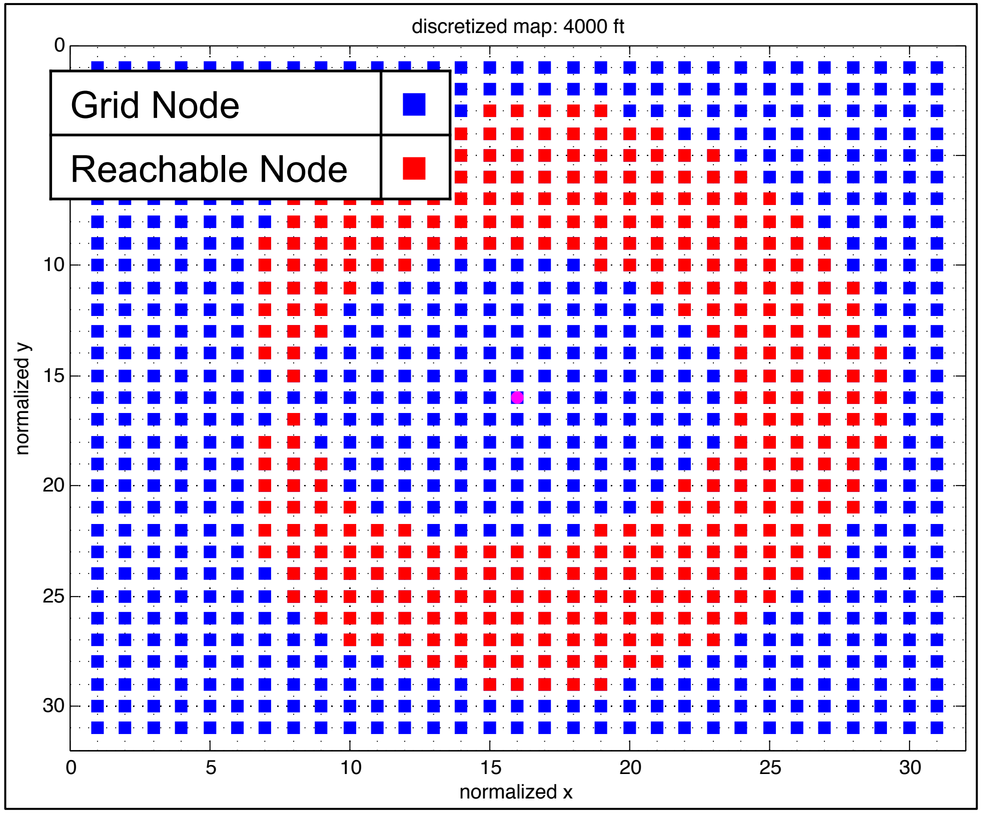
V – Stall avoidance

α, ϕ – Keeps aircraft within performance envelope

Acceleration - structural load limits

Discrete Reachable Sets

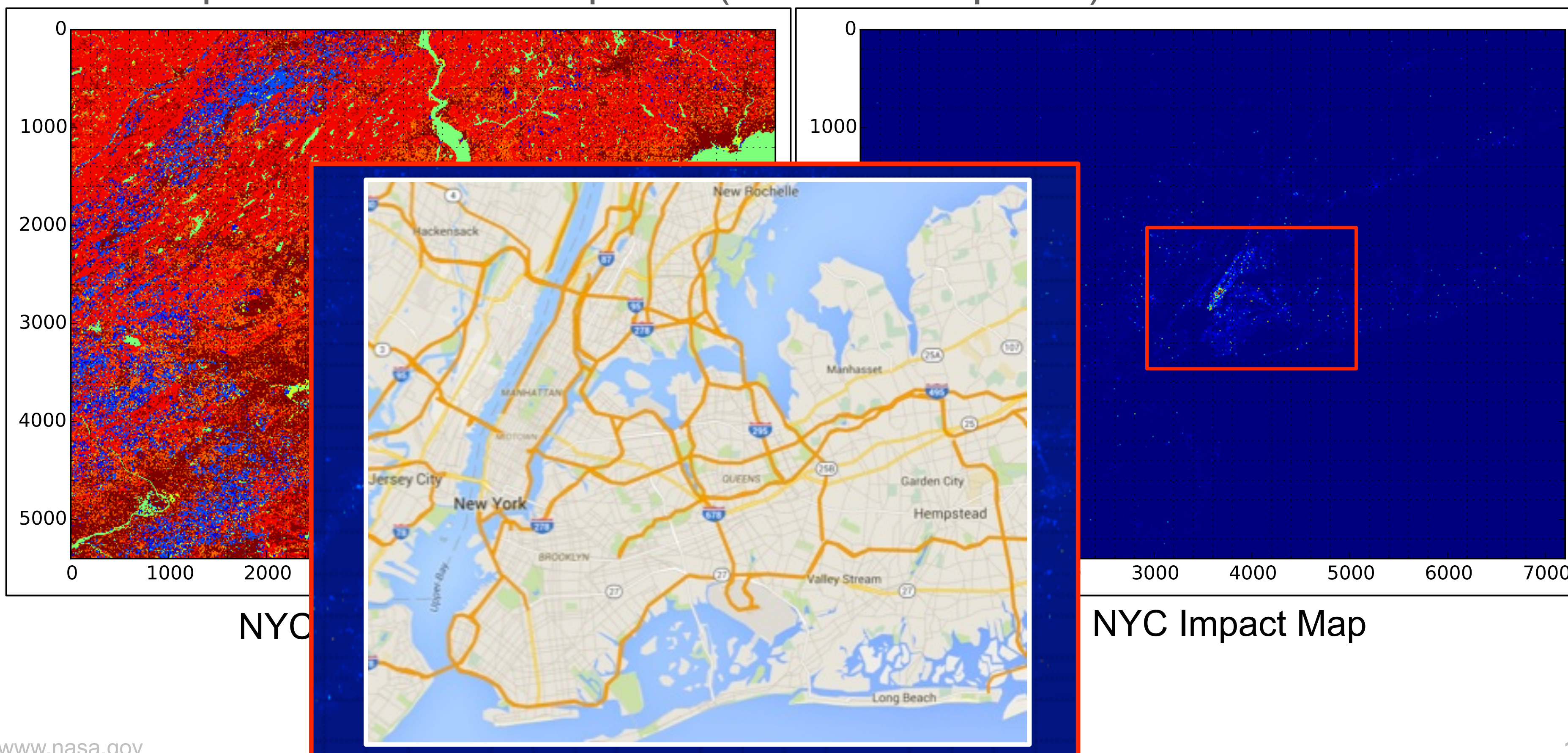




- Reachable sets generated every 100 ft from 1000 ft - 4000 ft
- Grid size 10E4x10E4 ft
- Normalized and stored as a binary map
- Oriented onto global map using aircraft heading

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)



NYC

NYC Impact Map

Gliding Aircraft Equations

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

$$\dot{X} = V \cos\gamma \cos\xi$$
$$\dot{Y} = V \cos\gamma \sin\xi$$
$$\dot{Z} = V \sin\gamma$$

Aircraft velocities

$$\dot{V} = -\frac{D(\alpha, V)}{m} - g \sin\gamma$$

Aircraft acceleration

$$\dot{\gamma} = \frac{L(\alpha, V)\cos\phi}{mV} - \frac{g}{V}\cos\gamma$$

Flight path derivative

$$\dot{\xi} = \frac{L(\alpha, V)\sin\phi}{mV\cos\gamma}$$

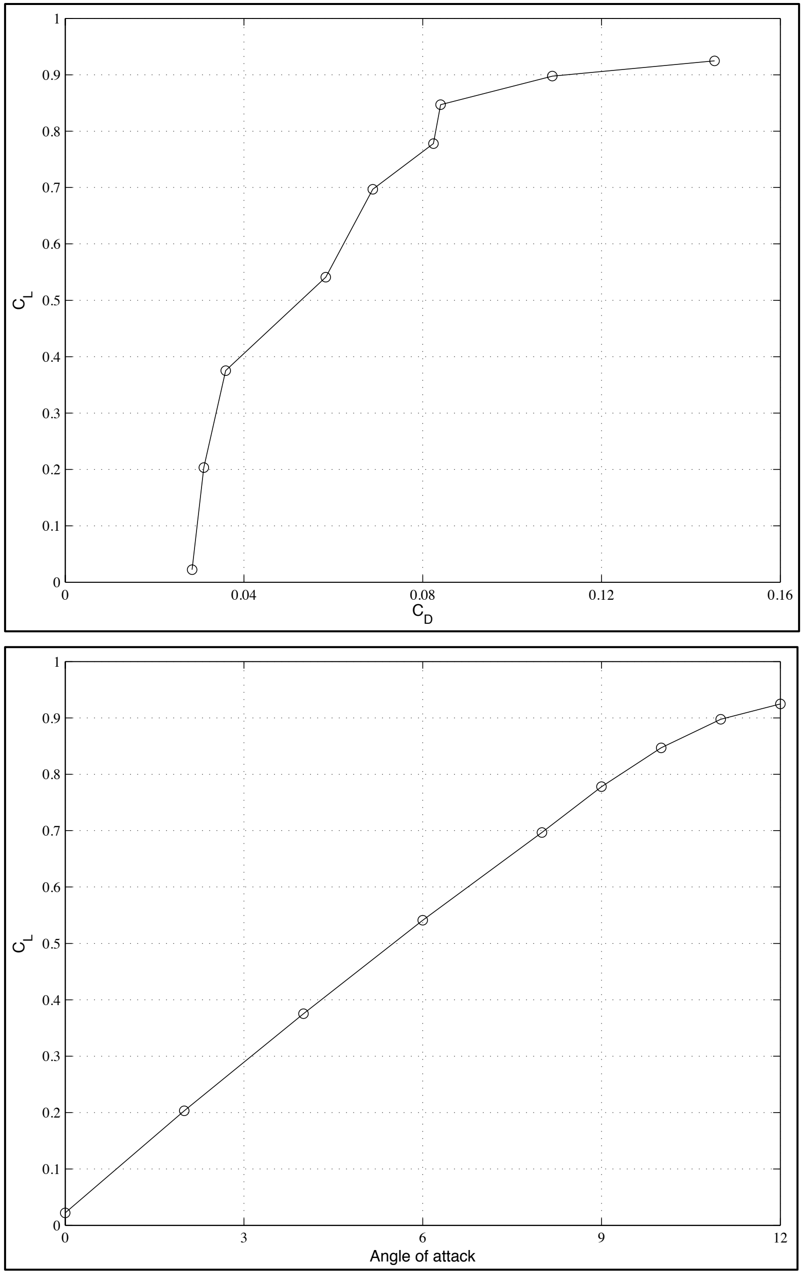
Heading derivative

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

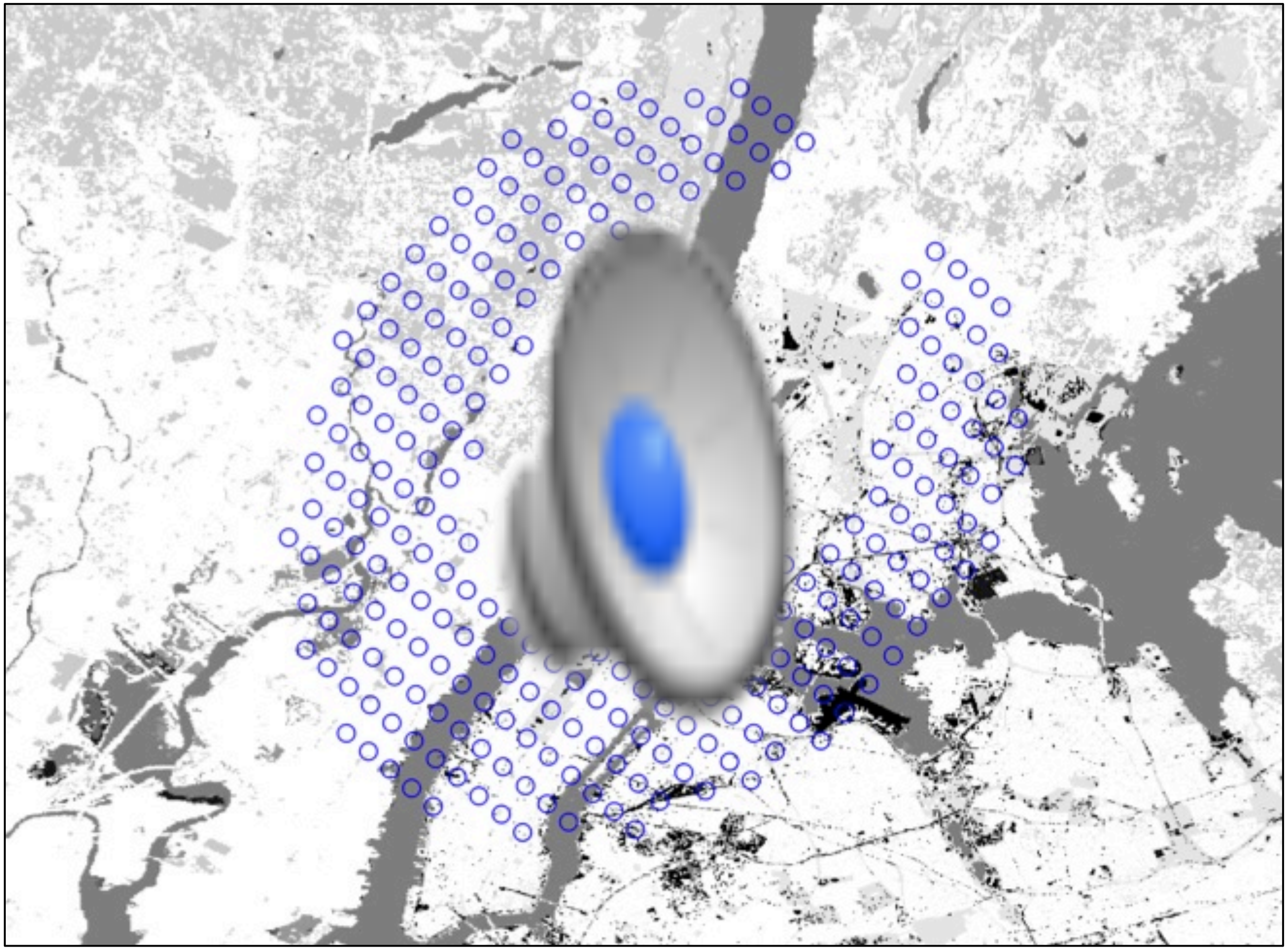
Optimum glide velocity

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_{MAX} and α_{MAX}



Optimal Landing Location

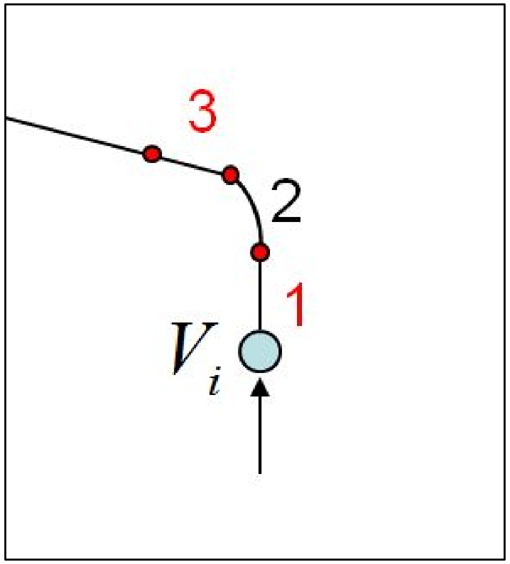


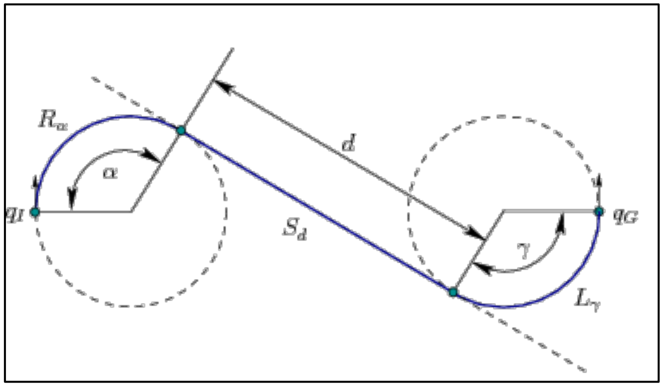
o	Reachable Node
●	Searched Node
■	Landing Footprint

- 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

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

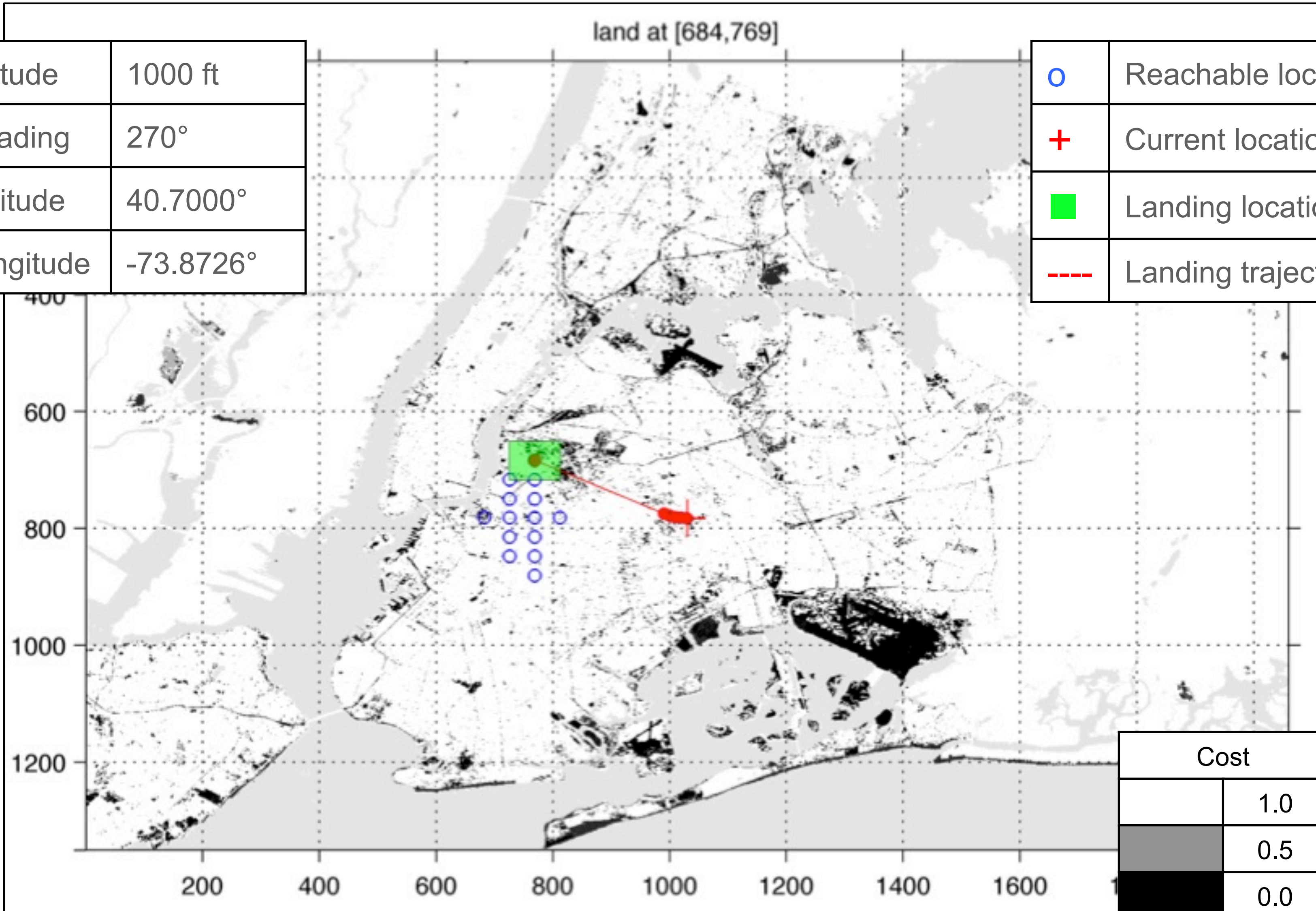
Test Cases

Test #	Altitude (ft)	Latitude	Longitude	Initial Heading
1	1000	40.70°	-73.8726°	270°
2	1000	40.70°	-73.8726°	15°
3	1000	40.85°	-73.70°	270°
4	4000	40.70°	-73.8726°	270°
5	4000	40.70°	-73.8726°	15°
6	4000	40.85°	-73.70°	270°
7	4000	40.85°	-73.70°	15°
8	3026	40.865	-73.88°	220

- 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

Altitude	1000 ft
Heading	270°
Latitude	40.7000°
Longitude	-73.8726°

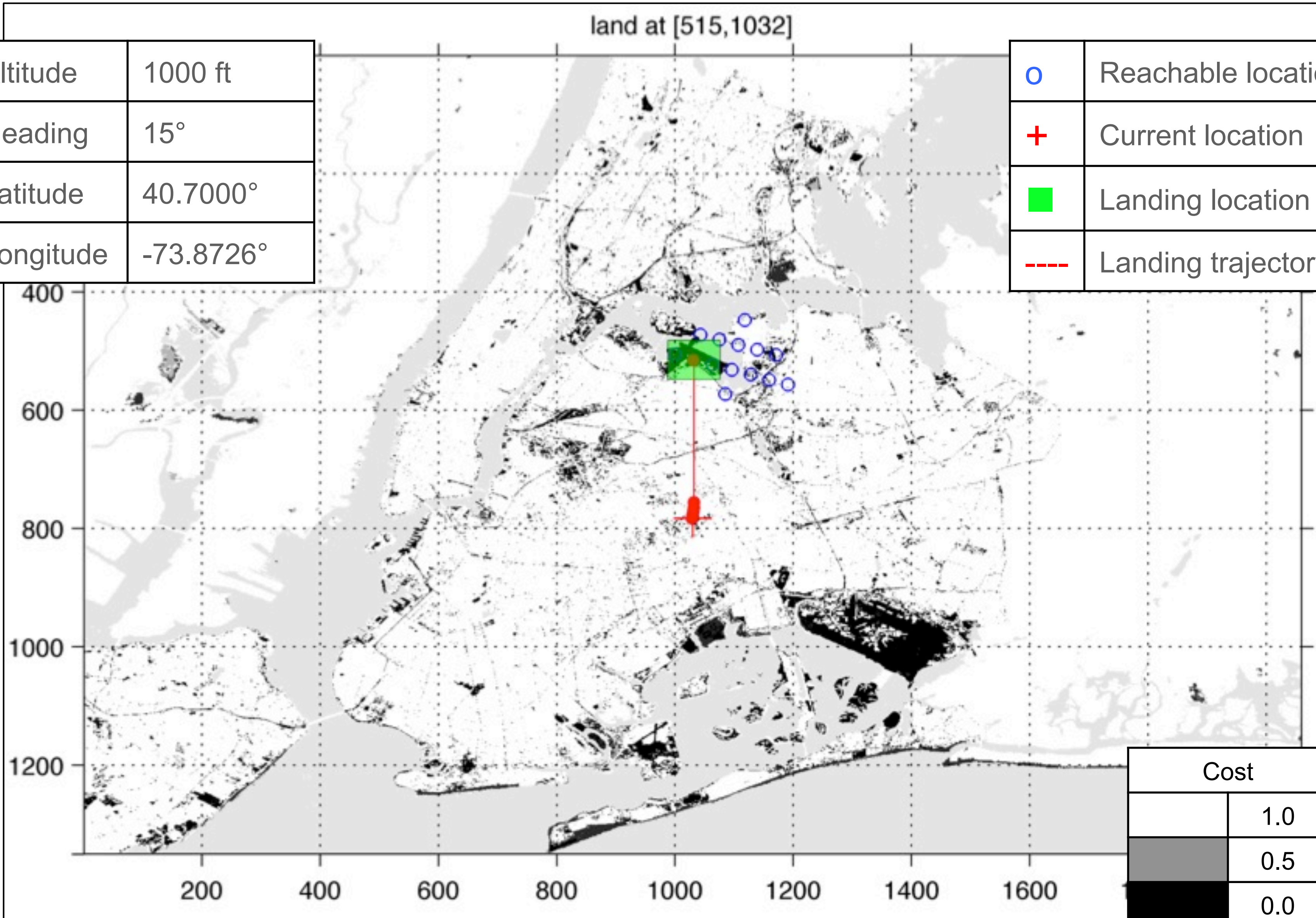


o	Reachable location
+	Current location
■	Landing location
----	Landing trajectory

Cost	
(white)	1.0
(gray)	0.5
(black)	0.0

Results: Test Case 2

Altitude	1000 ft
Heading	15°
Latitude	40.7000°
Longitude	-73.8726°

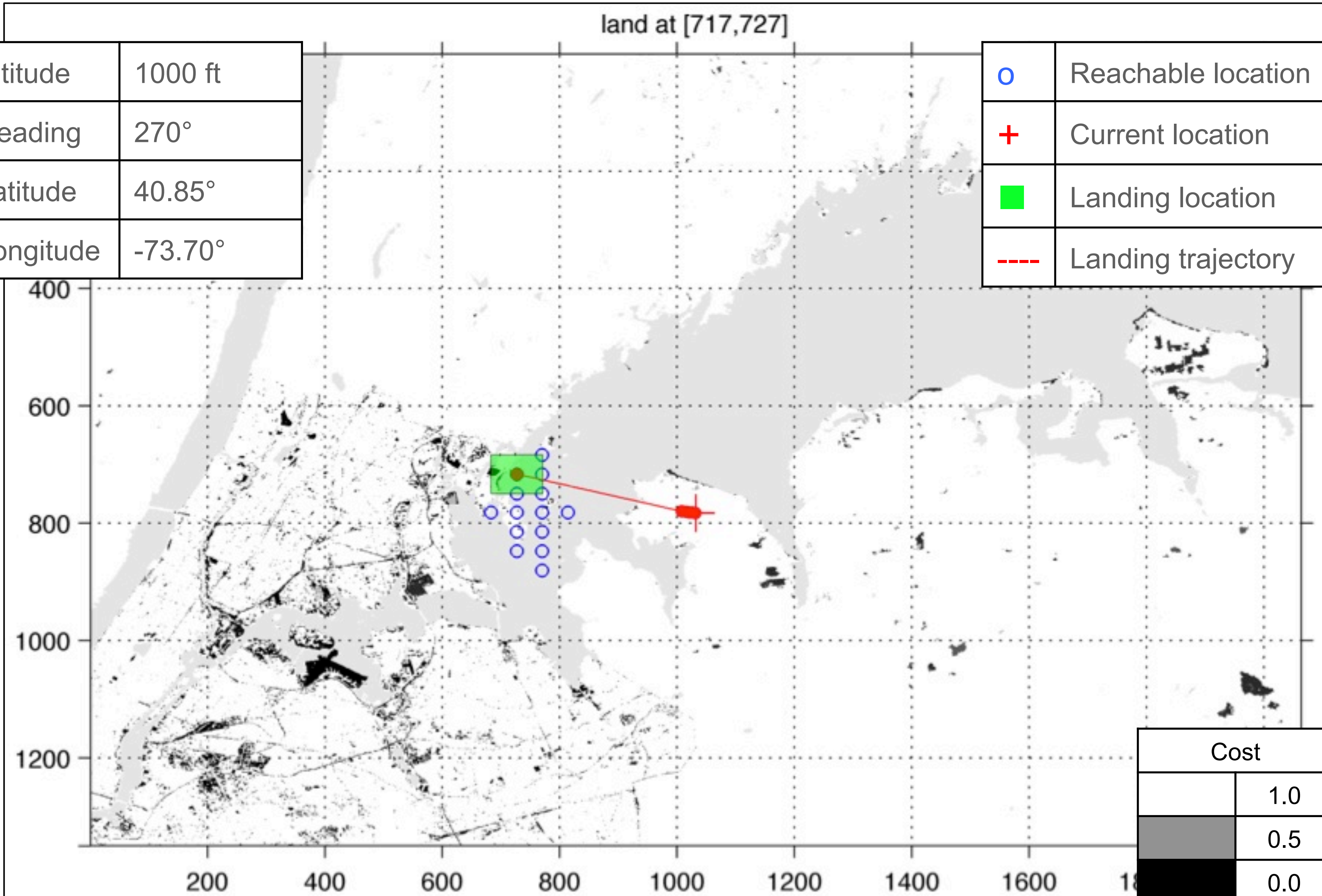


o	Reachable location
+	Current location
■	Landing location
----	Landing trajectory

Cost	
	1.0
	0.5
	0.0

Results: Test Case 3

Altitude	1000 ft
Heading	270°
Latitude	40.85°
Longitude	-73.70°

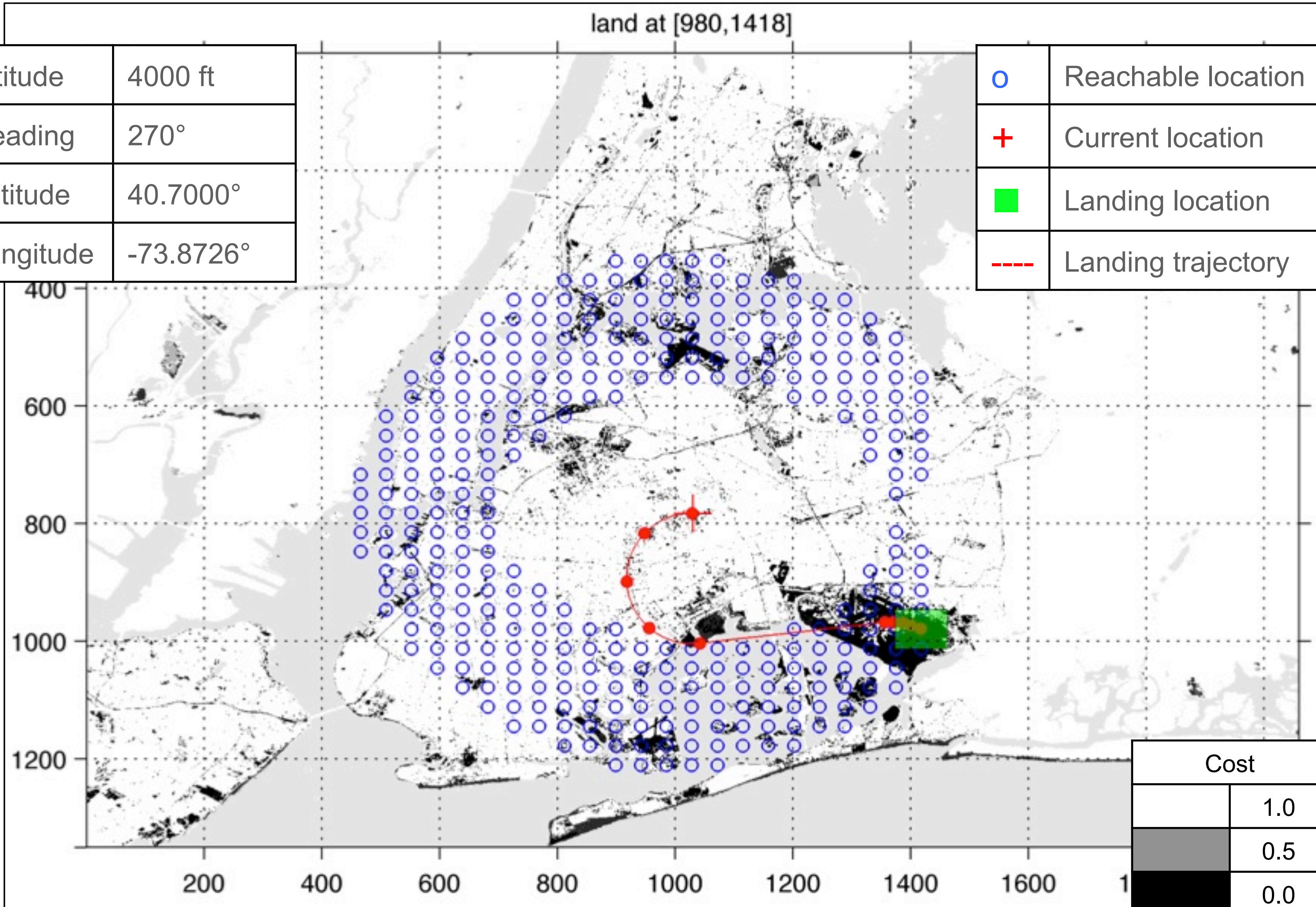


o	Reachable location
+	Current location
■	Landing location
----	Landing trajectory

Cost	
(white)	1.0
(gray)	0.5
(black)	0.0

Results: Test Case 4

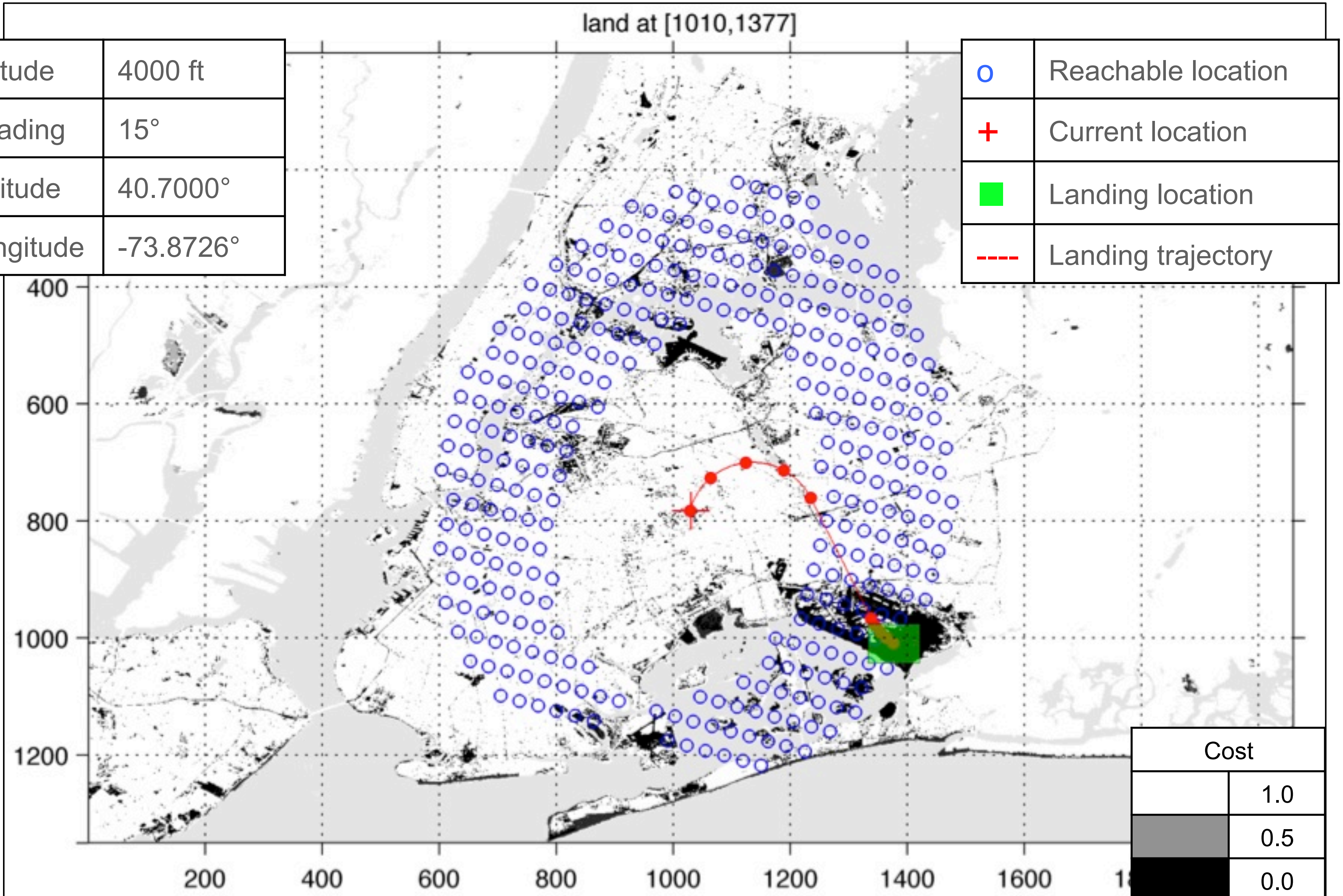
Altitude	4000 ft
Heading	270°
Latitude	40.7000°
Longitude	-73.8726°



o	Reachable location
+	Current location
■	Landing location
----	Landing trajectory

Results: Test Case 5

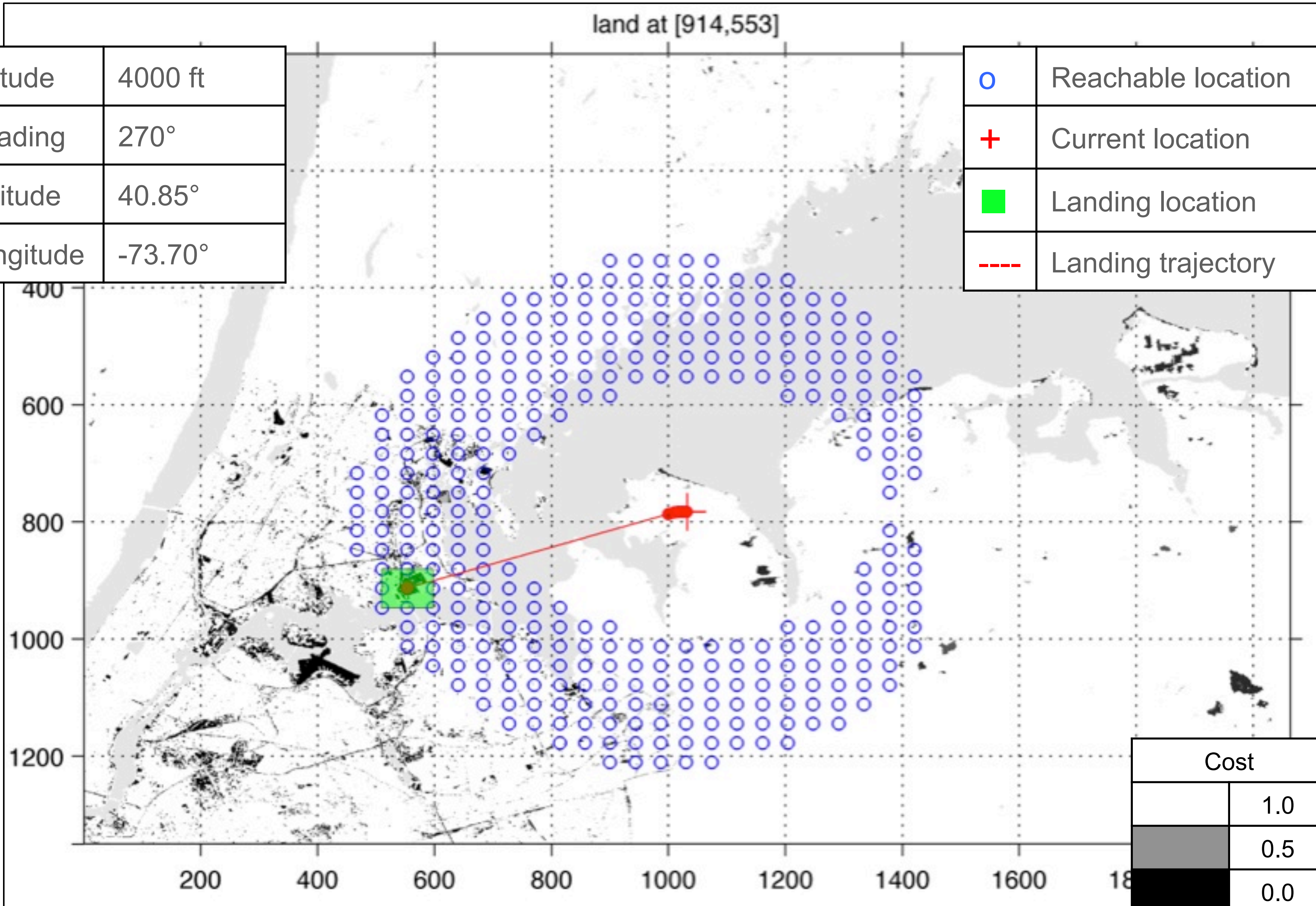
Altitude	4000 ft
Heading	15°
Latitude	40.7000°
Longitude	-73.8726°



Results: Test Case 6

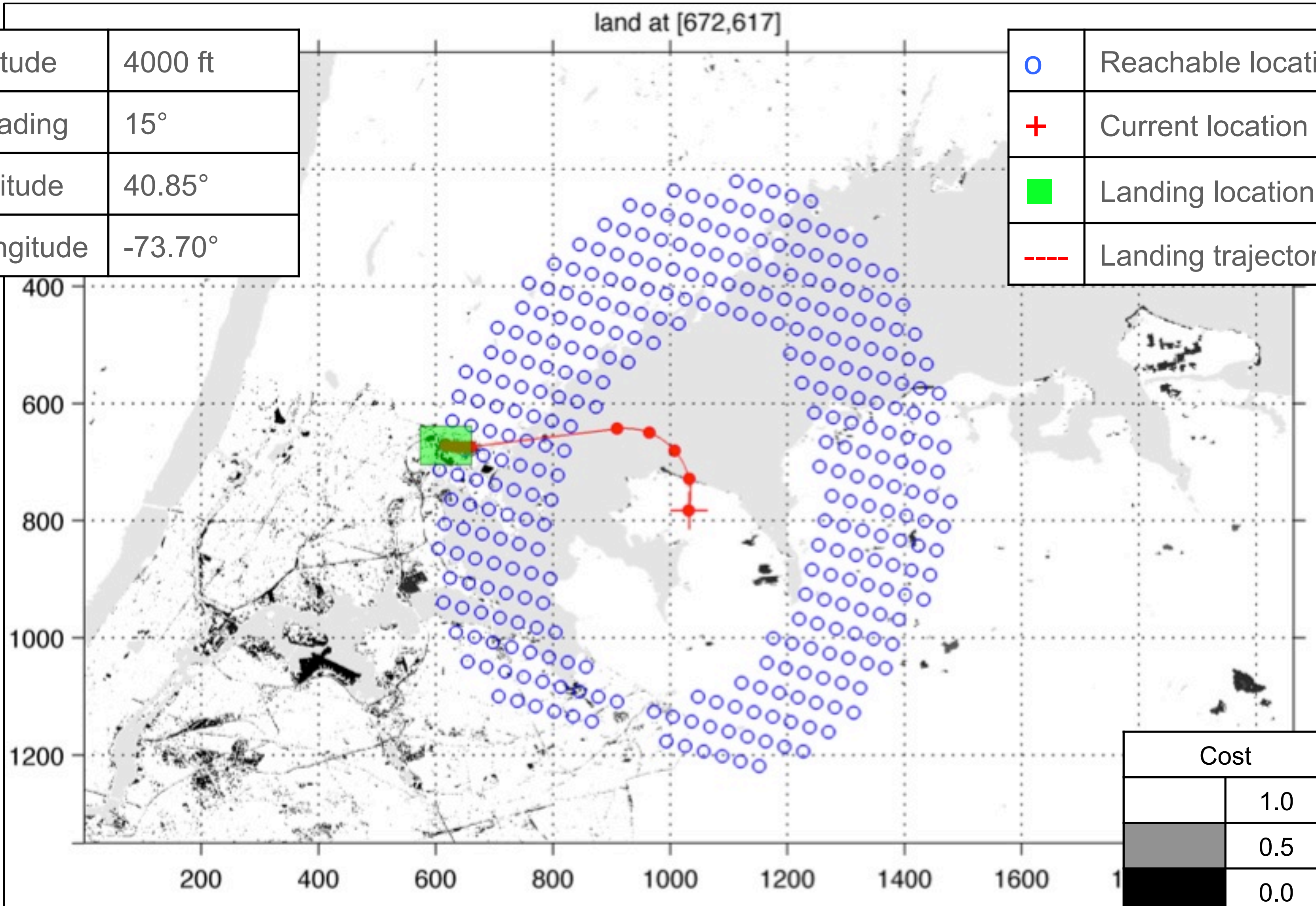
Altitude	4000 ft
Heading	270°
Latitude	40.85°
Longitude	-73.70°

o	Reachable location
+	Current location
(green square)	Landing location
----	Landing trajectory



Results: Test Case 7

Altitude	4000 ft
Heading	15°
Latitude	40.85°
Longitude	-73.70°

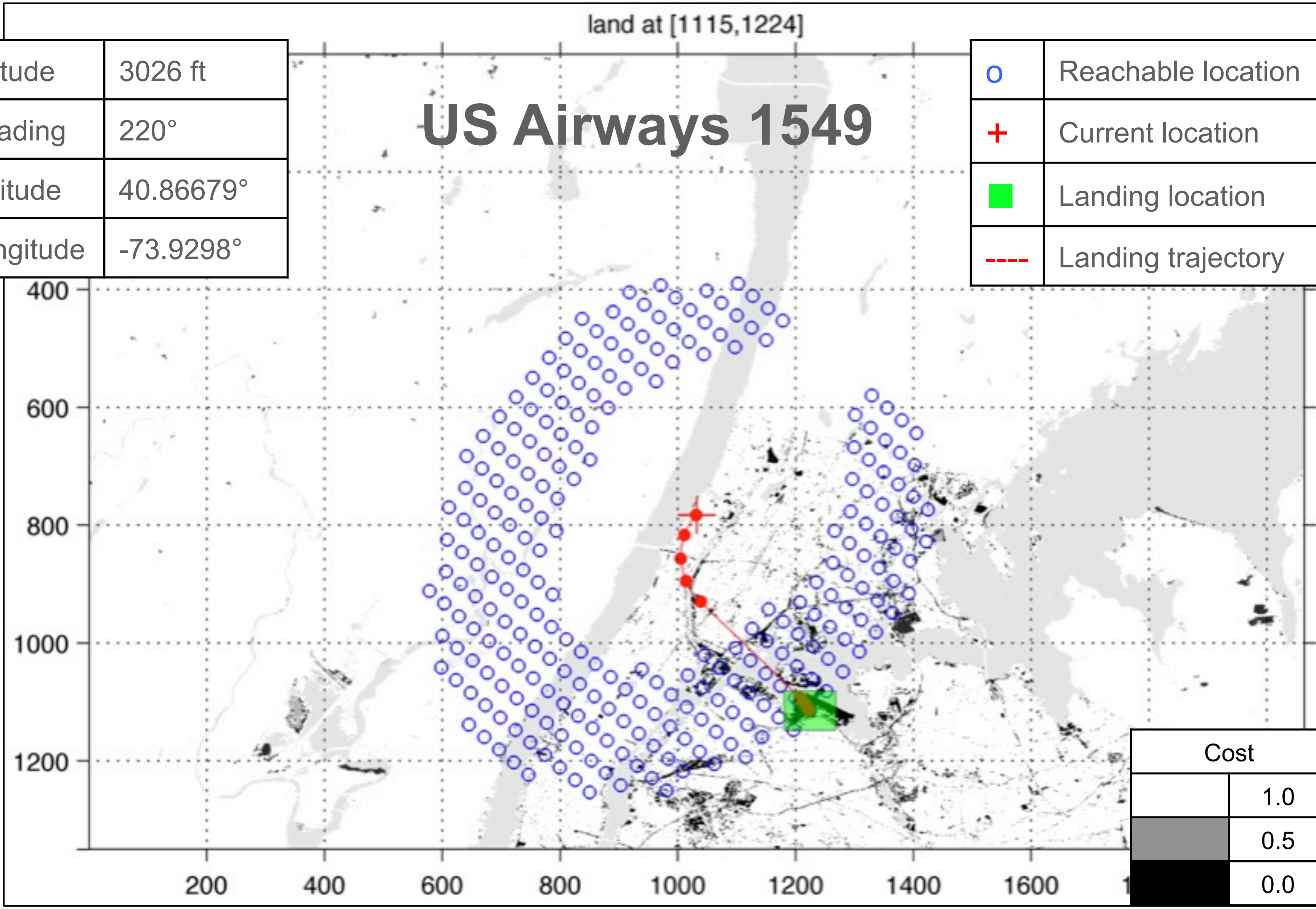


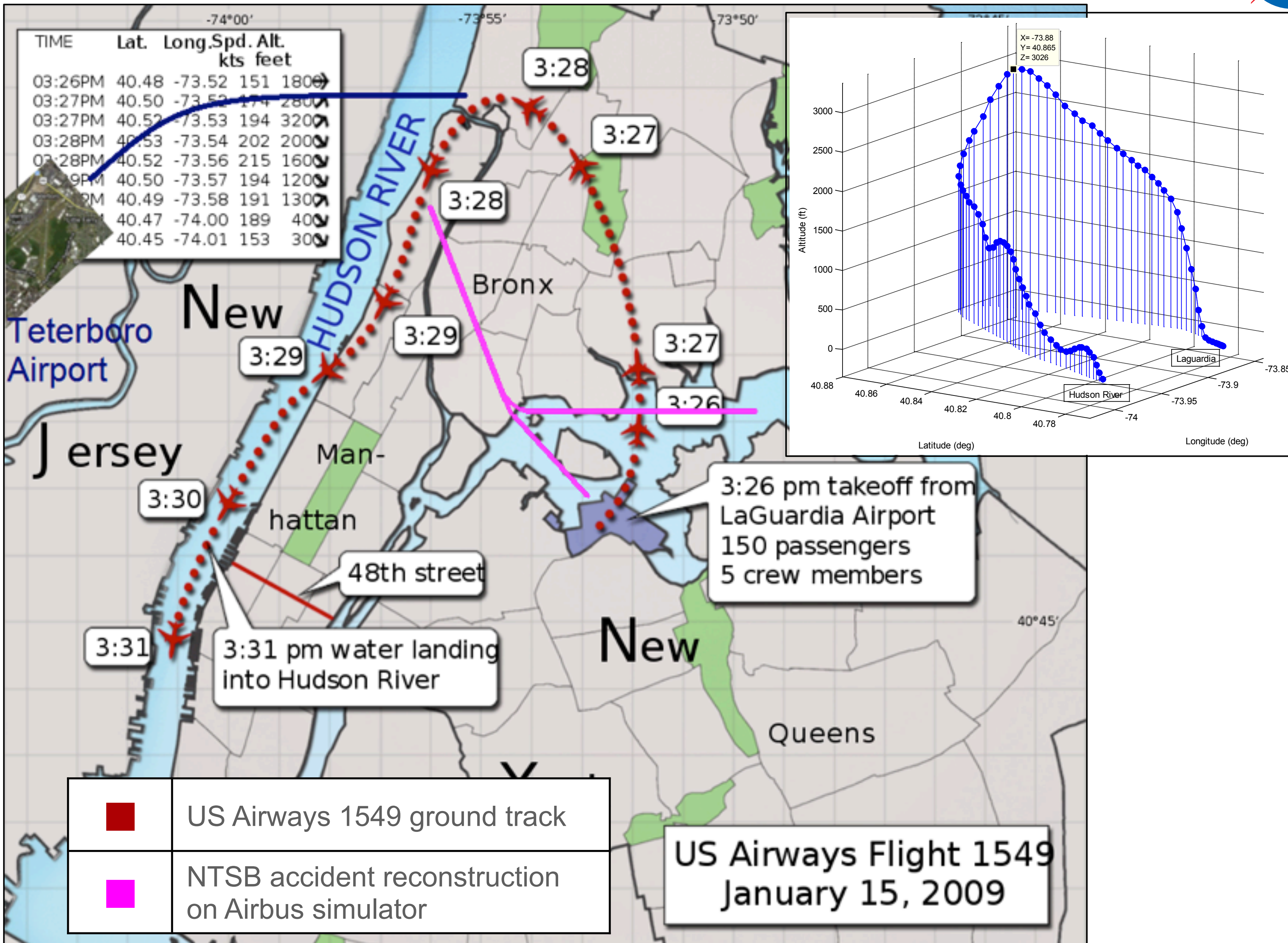
o	Reachable location
+	Current location
(green square)	Landing location
----	Landing trajectory

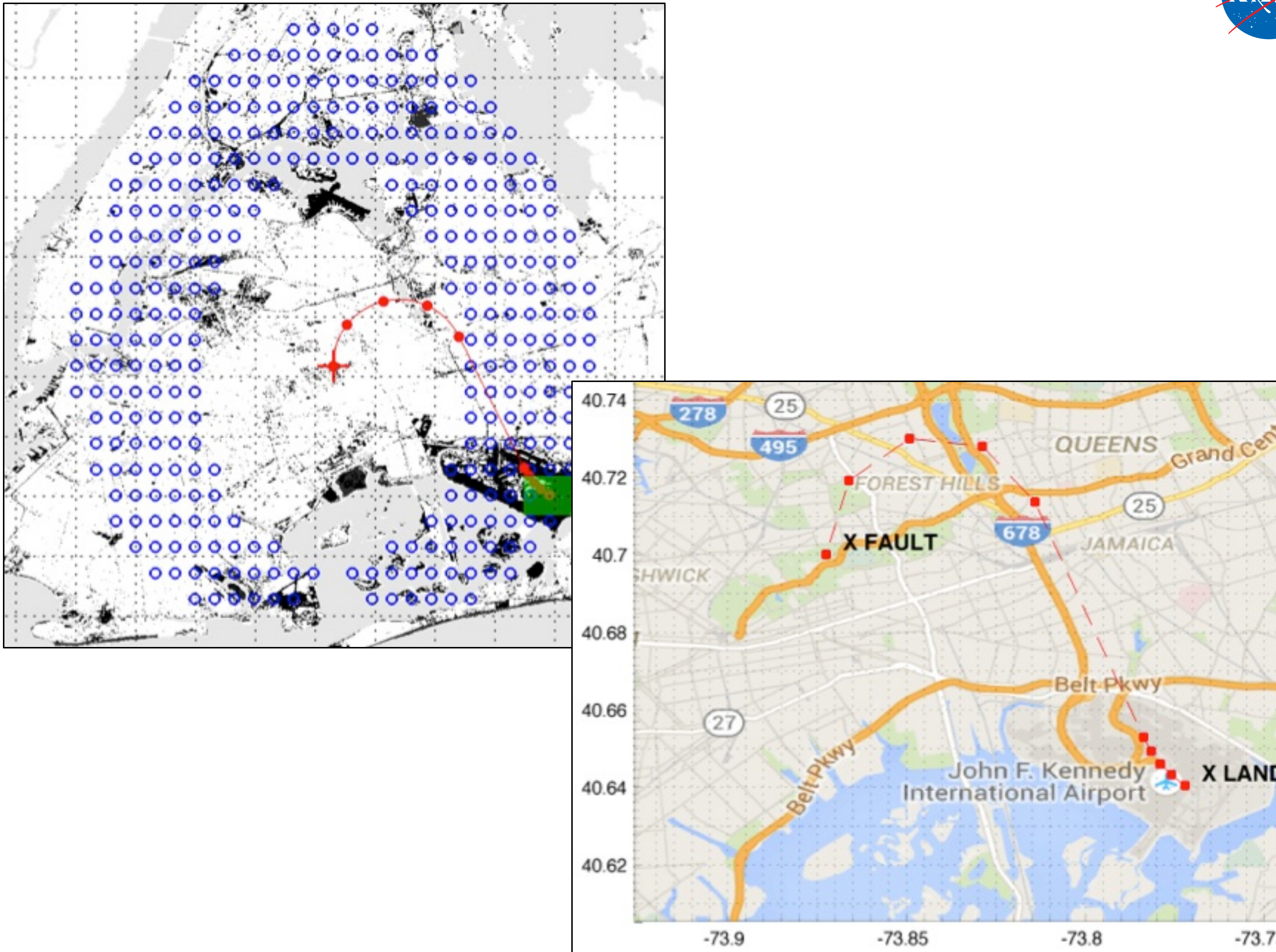
Cost	
	1.0
	0.5
	0.0

Results: Test Case 8

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




US Airways Flight 1549
January 15, 2009
TIME Lat. Long. Spd. kts Alt. feet
03:26PM 40.48 -73.52 151 180
03:27PM 40.50
03:27PM 40.52 -73.53 194 320
03:28PM 40.53 -73.54 202 200
40.52 -73.56 215 160
40.50 -73.57 194 120
40.49 -73.58 191 130
40.47 -74.00 189 40
40.45 -74.01 153 30
-74°00′
-73°55′
-73°50′
40°45′
Teterboro Airport
New Jersey
HUDSON RIVER
Bronx
Manhattan
New York
Queens
3:26
3:27
3:27
3:28
3:28
3:29
3:29
3:30
3:31
3:26 pm takeoff from LaGuardia Airport
150 passengers
5 crew members
48th street
3:31 pm water landing into Hudson River
US Airways 1549 ground track
NTSB accident reconstruction on Airbus simulator
X= -73.88
Y= 40.865
Z= 3026
Altitude (ft)
3000
2500
2000
1500
1000
500
0
Laguardia
Hudson River
Latitude (deg)
40.88
40.86
40.84
40.82
40.8
40.78
Longitude (deg)
-73.85
-73.9
-73.95
-74



40.74
40.72
40.7
40.68
40.66
40.64
40.62
-73.9
-73.85
-73.8
-73.75
278
25
495
QUEENS
FOREST HILLS
25
678
X FAULT
JAMAICA
27
Belt Pkwy
Belt Pkwy
John F. Kennedy International Airport
X LAND

WTL2 HIL Simulation

HIL Simulation Data Overlay	
WTL State	On/Off
Target V (kts)	###
Target Heading	###
Waypoint #	#/#

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

WTL2 (current)

- Loss of Thrust
- TCM/B-757
- Linux
- Local Static Cost Map

WTL3+ (future)

- Common A/C Faults
- Any aerial vehicle
- Linux, Mobile, PixHawk
- Global Dynamic Cost Map

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

General Aviation	Can improve odds of survival
Commercial	Gives pilots more options
Unmanned Vehicles	Can enable expanded UAS in the NAS

Distribution

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

References

1. Mitchell, I., Bayen, A., and Tomlin, C.J., “Computing Reachable Sets for Continuous Dynamics Games Using Level Set Methods”
2. Tomlin, C., Lygeros, J., and Sastry, S., “A Game Theoretic Approach to Controller Design For Hybrid Systems”
3. Ding, J., Gillua, H., Huang, H., “Hybrid Systems in Robotics”
4. Adler, A., Bar-Gill and A., and Shimkin, N., “Optimal Flight Paths for Engine Out Emergency Landing”
5. Rogers, D., “The Possible ‘Impossible’ Turn”
6. Atkins, E., “Emergency Landing Automation Aids: An Evaluation Inspired by US Airways Flight 1549”
7. Bayen, A., Mitchell, I., Oishi, M., and Tomlin, C.J., “Aircraft Autolander Safety Analysis through Optimal Control Based Reach Set Computation”
8. Shkel, A., and Lumelsky, V., “Classification of the Dubins Set”
9. Hueschen, R., “Development of the Transport Class Model Aircraft Simulation from a Sub-Scale Generic Transport Model Simulation”