3-Axis Attitude Determination and Control of the AeroCube-4 CubeSats

Darren Rowen
Rick Dolphus
The Aerospace Corporation

Vehicle Systems Division
10 August 2013

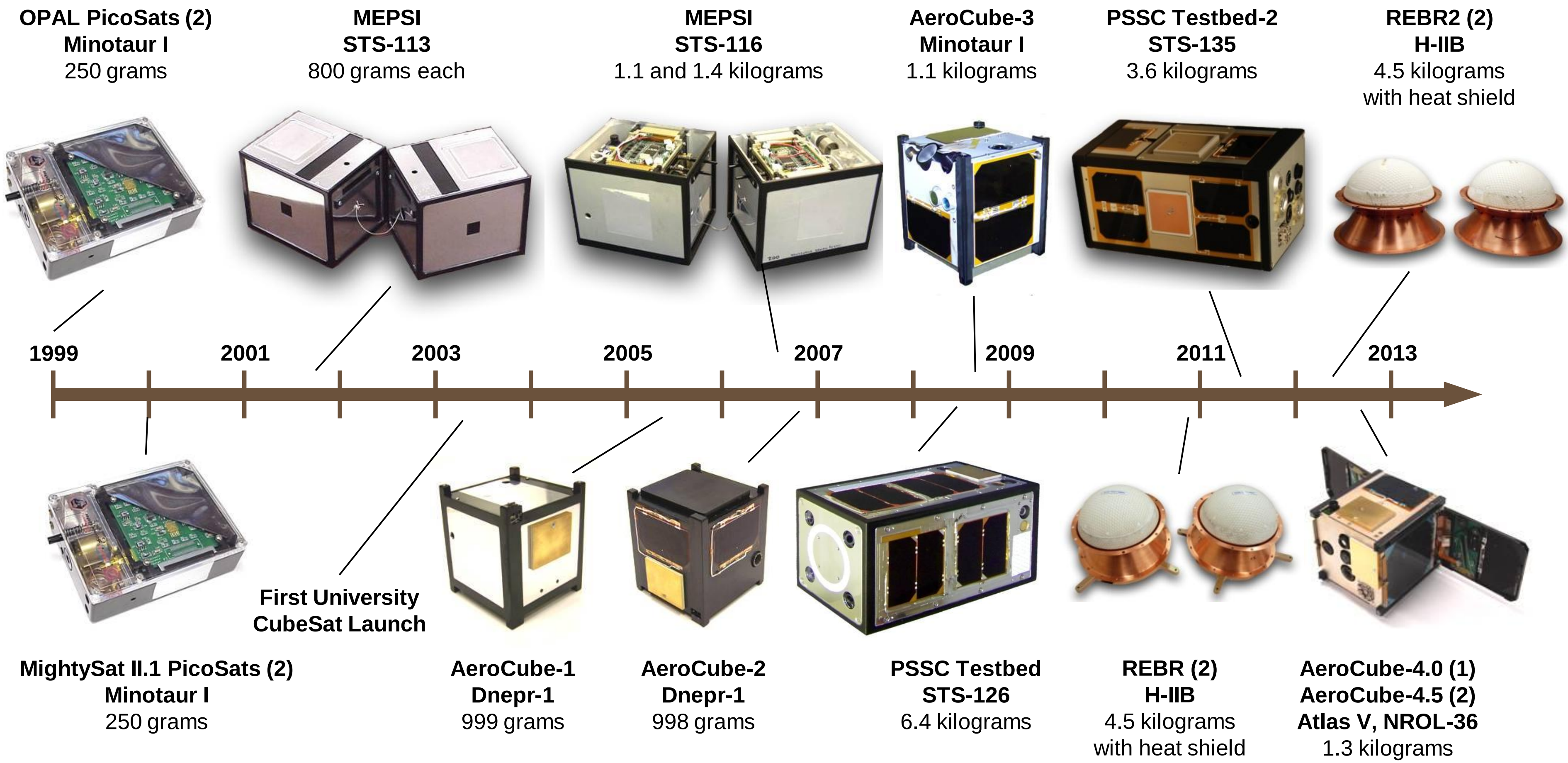
Topics

- AeroCube History and Overview
- Hardware
- Attitude Determination and Control
- Flight Software
- Photos

AeroCube History and Overview

Darren.Rowen@aero.org
Control Analysis Department

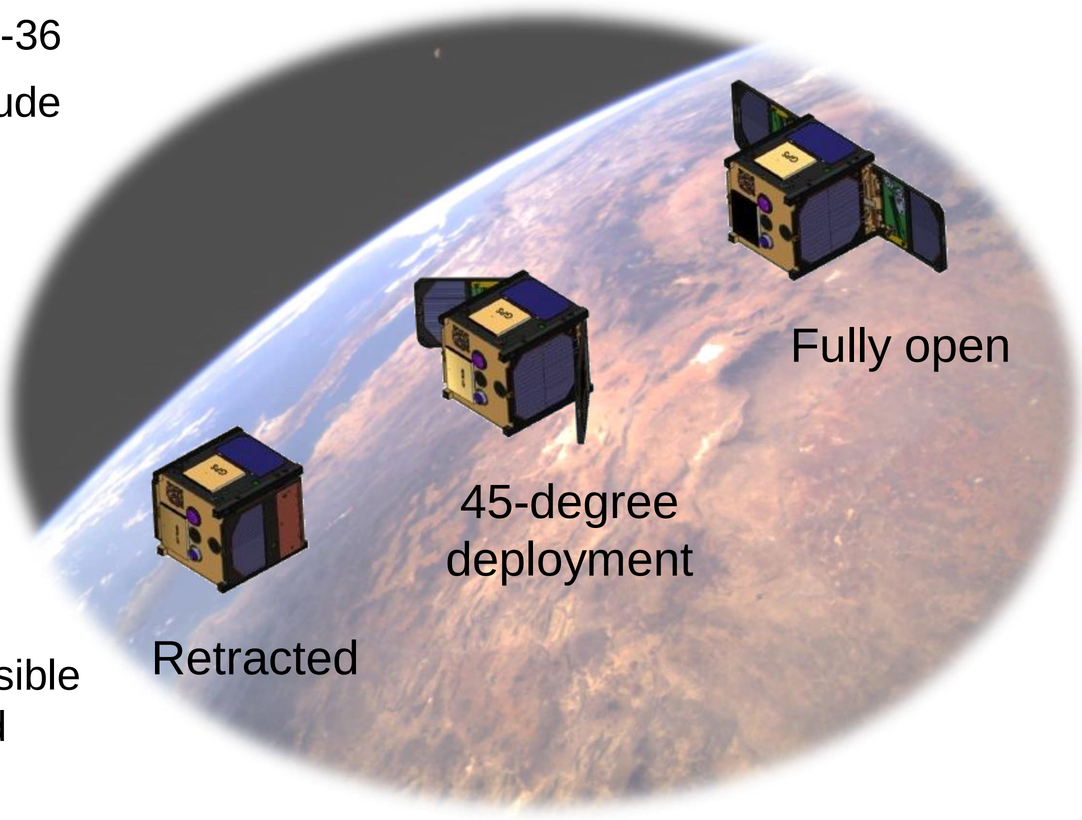
Aerospace’s “PicoSat” History



Darren.Rowen@aero.org
Control Analysis Department

Aerocube 4 Satellites

- Launched September 2012 on NROL-36
- 63° inclination and 470 x 780 km altitude
- 10 x 10 x 10 cm size
- 1.3 kg total mass
- Full attitude control
- Adjustable wings for variable drag
- 2 ft. diameter x 1.5 ft. tall conical deorbit chute
- Avionics include GPS, redundant radios, and reprogrammable on orbit
- Sensor includes three 2 megapixel visible cameras with 1 km and 10 km ground resolution and one “fisheye” lens



Variable-angle wings for orbit control

- Lower altitude or accelerate deorbit
- Avoid collision with another space object

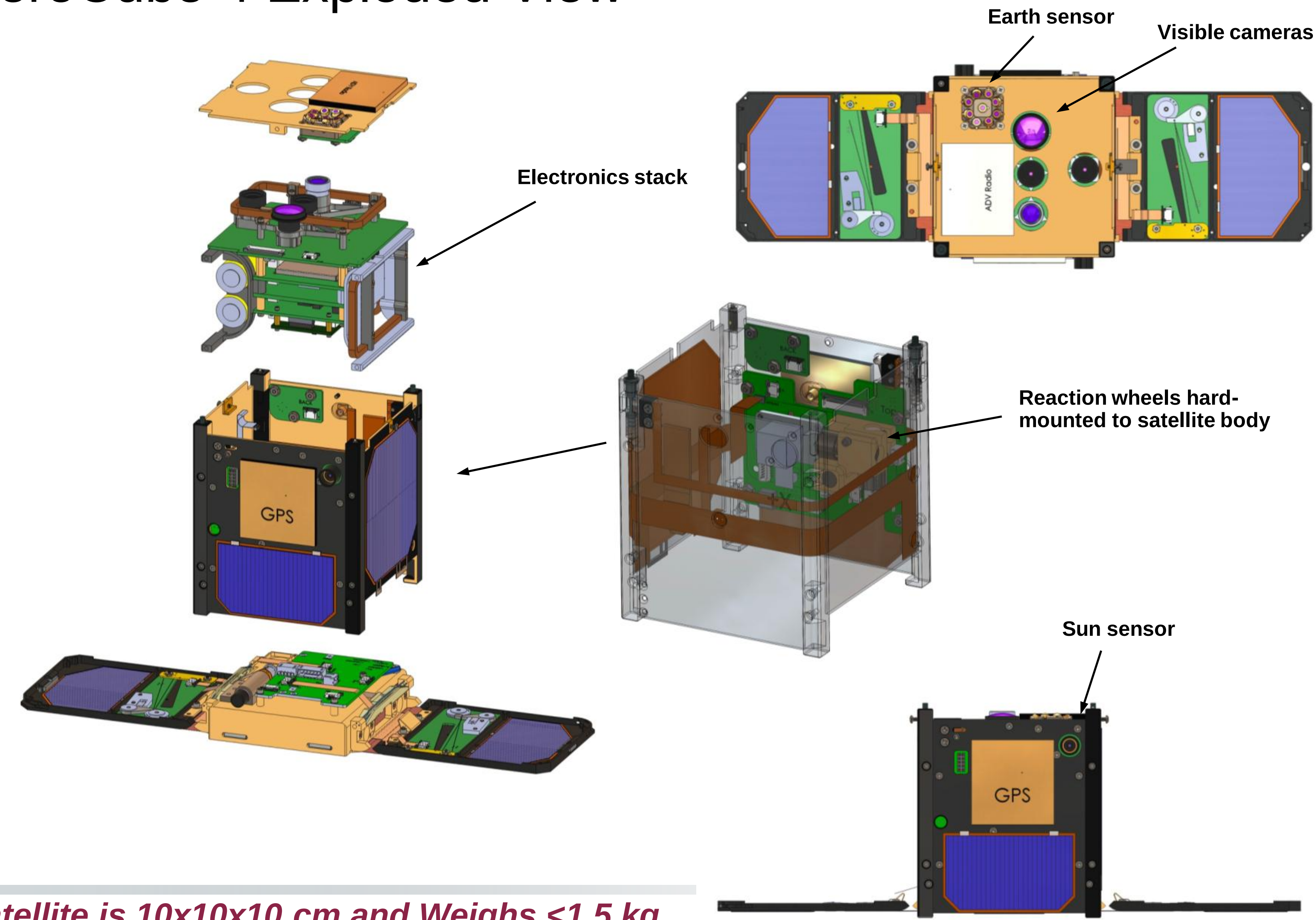
Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Hardware

Darren.Rowen@aero.org
Control Analysis Department

AeroCube 4 Exploded View

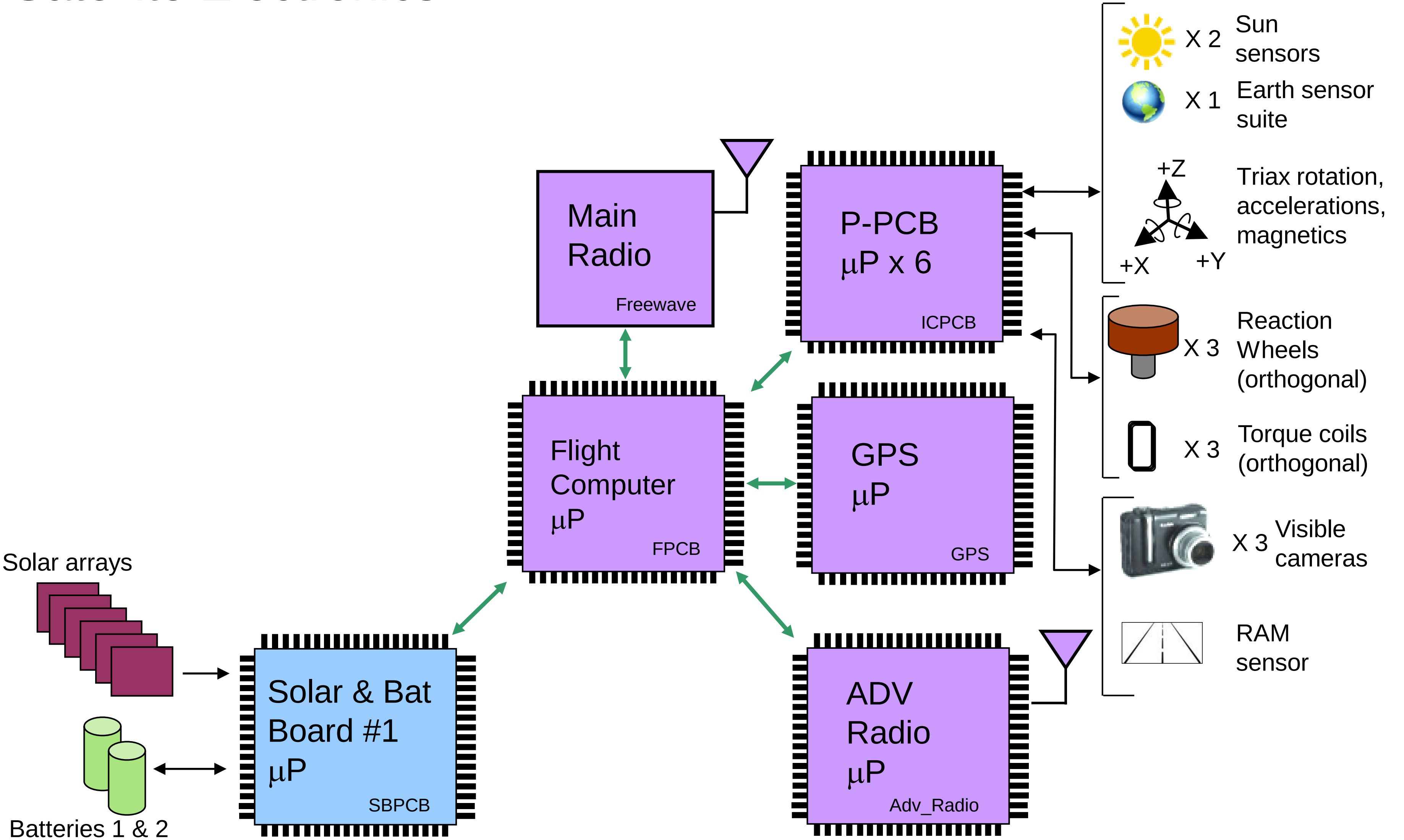


Satellite is 10x10x10 cm and Weighs <1.5 kg

Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

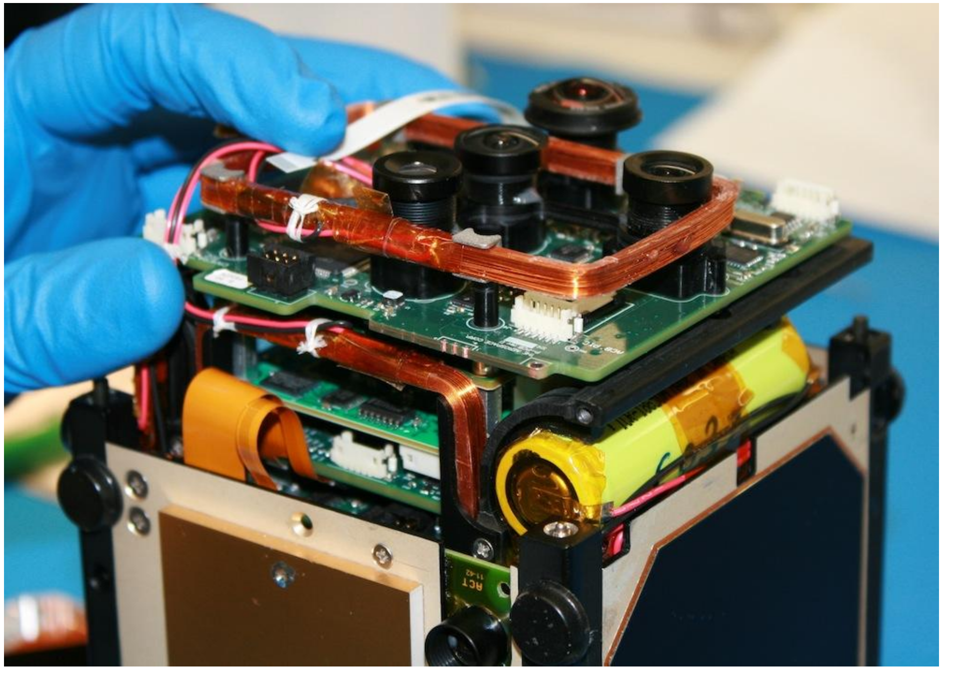
Satellite Electronics



Darren.Rowen@aero.org
Control Analysis Department

Satellite Electronics

- Attitude Control on a PIC Processor
  - *8-bit architecture*
  - *Unfolded 32-bit floating point math*
  - *No operating system*
  - *Timing through on-chip timers*



Darren.Rowen@aero.org
Control Analysis Department

Attitude Determination and Control

Darren.Rowen@aero.org
Control Analysis Department

Attitude Determination (1 of 2)

- Attitude sensors
  - Sun sensor
  - Earth sensor
  - Magnetometer
- Require 2 of 3 attitude sensors to get full 3-axis attitude (gyroless design)
  - Sun sensor alignment and calibration is based on ground measurements/testing
  - Earth sensor alignment and calibration is fine-tuned on-orbit
  - Magnetometer biases recalculated on-orbit at beginning of each mission (using attitude from Sun sensor and Earth sensor while nadir pointing)
- Nadir pointing
  - Use Sun & Earth
- Off-nadir (including inertial pointing)
  - Use Sun & Magnetometer (Earth sensor less accurate or unavailable off-nadir)

Darren.Rowen@aero.org
Control Analysis Department

Attitude Determination (2 of 2)

- Fixed covariance filter
  - Based on Kalman filter equations, but with fixed state covariance matrix
  - Fewer calculations than full Kalman filter (satisfies processing time constraints)
  - Propagate state (but do not propagate state covariance, i.e. use steady state solution)
  - State = [attitude ; rate]
  - State covariance, P
    - Solve for steady state covariance P_{ss}
    - Keep dominant terms (block diagonals)
  - Kalman filter gain matrix (info only): $K = P*H^T * (H*P*H^T + R)^{-1}$
  - Gain matrix (assume $H*P*H^T << R = I*\sigma^2$): $K = (P_{ss} / \sigma^2) * H^T$
  - Example: Earth sensor measurement (boresight along Z-axis)

Measurement:

$$y = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Measurement Matrix:

$$H = \begin{bmatrix} 0 & -u_3 & u_2 & 0 & 0 & 0 \\ u_3 & 0 & -u_1 & 0 & 0 & 0 \end{bmatrix}$$

State Update:

$$\Delta x = \frac{1}{\sigma^2} \begin{bmatrix} P_{11} & 0 & 0 \\ 0 & P_{22} & 0 \\ 0 & 0 & P_{33} \\ \hline P_{41} & 0 & 0 \\ 0 & P_{52} & 0 \\ 0 & 0 & P_{63} \end{bmatrix} \begin{bmatrix} 0 & u_3 \\ -u_3 & 0 \\ u_2 & -u_1 \end{bmatrix} \left(y_{meas} - y_{pred} \right)$$

Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Attitude Control

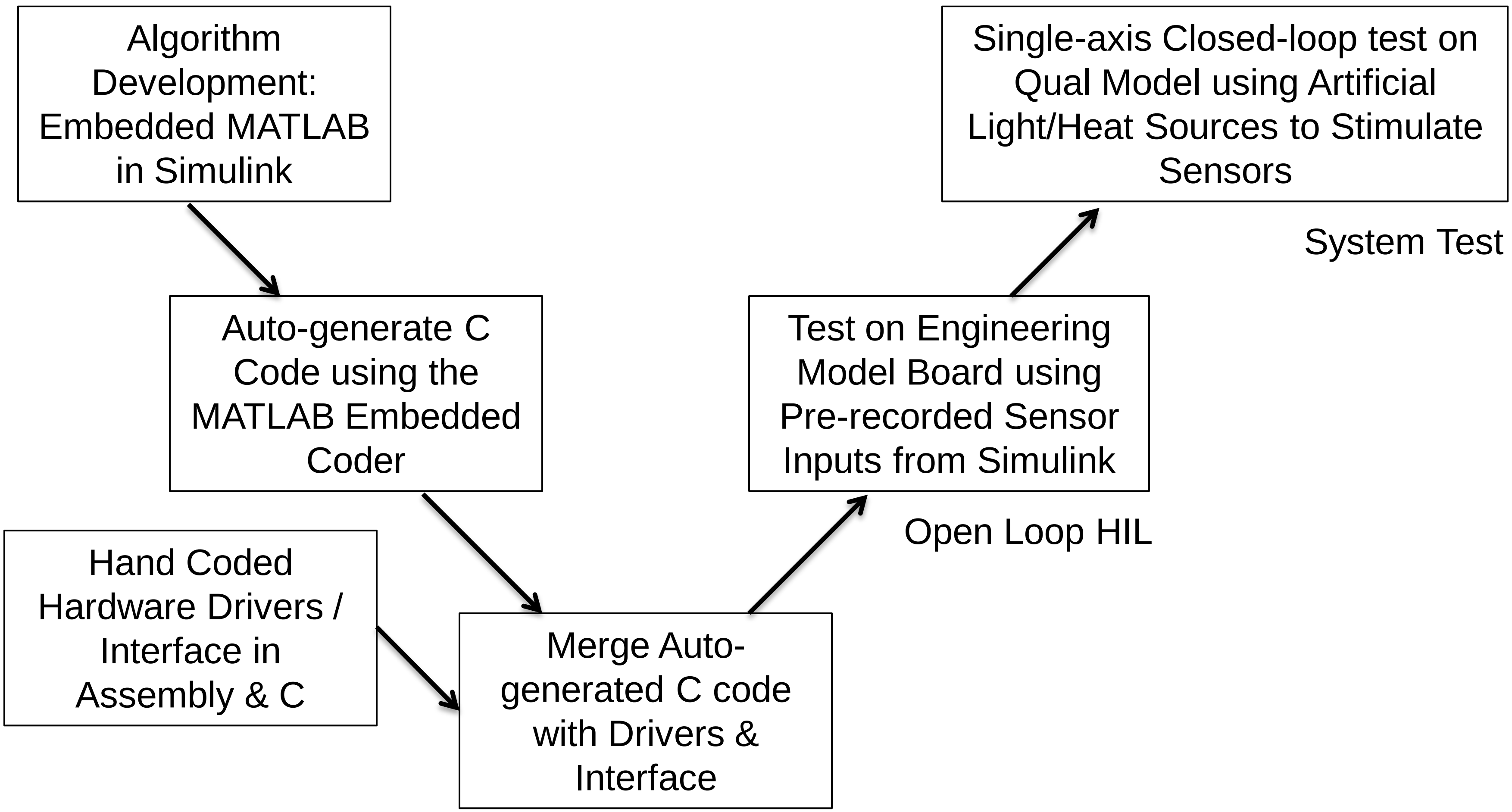
- 3-axis attitude control
  - PI control law
  - Actuators: Reaction wheels (speeds up to 100 KRPM)
- Momentum control
  - Momentum builds due to spacecraft dipole and atmospheric drag
  - Dump wheel momentum by applying coil torque = M x B
  - Earth magnetic field B
    - B in ECI-frame is calculated using polynomial fit generated on ground and uploaded to spacecraft
    - Map B to Body-frame using estimated attitude
  - Actuators: Magnetic torque coils (magnetic moment M)

Darren.Rowen@aero.org
Control Analysis Department

Flight Software

Darren.Rowen@aero.org
Control Analysis Department

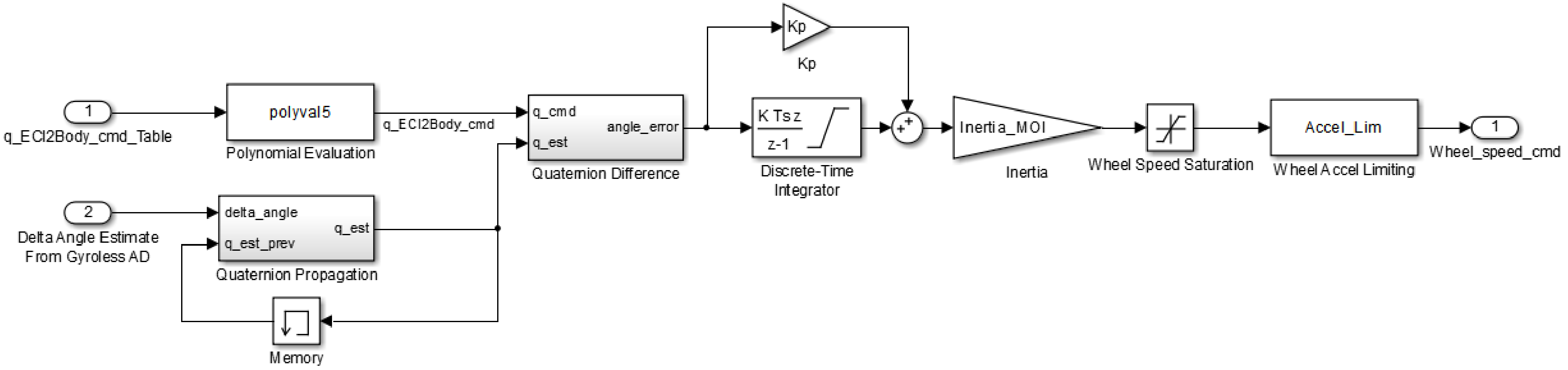
Software Development Process Flow



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Control System Block Diagram



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Flight Software Auto-Code Generation

- Streamlined Flight Software Development
- ACS Algorithms were developed in MATLAB
- The MATLAB Coder was used to auto-generate C code that was later merged with hand coded low level drivers and supporting command and data handling functions

/Users/dwr27059/Documents/AeroCube/Attitude_Control.m

```matlab
persistent Integrated_Gyro_Angle_prev Quat_ECI2Body_est Integral_State

if isempty(Integrated_Gyro_Angle_prev)
    Integrated_Gyro_Angle_prev=zeros(3,1);
    Quat_ECI2Body_est=[0;0;0;1];
    Integral_State=zeros(3,1);
end

% Quaternion Interpolation
q_ECI2Body_cmd_10Hz=QuatProp_rate(q_ECI2Body_cmd_1Hz,Ts,w_Body_cmd_1Hz);

% Delta Angle w/ rollover correction
deltatheta=Integrated_Gyro_Angle-Integrated_Gyro_Angle_prev;
for n=1:3
    if deltatheta(n)>deltatheta_Max
        deltatheta(n)=deltatheta(n)-Integrated_Gyro_Angle_RolloverNeg;
    elseif deltatheta(n)<deltatheta_Min
        deltatheta(n)=deltatheta(n)+Integrated_Gyro_Angle_RolloverPos;
    end
end
Integrated_Gyro_Angle_prev=Integrated_Gyro_Angle;

% Quaternion Propagation
Quat_ECI2Body_est=QuatProp_deltatheta(Quat_ECI2Body_est,deltatheta);

% Quaternion Difference
QuatError=QuatDiff(Quat_ECI2Body_est,q_ECI2Body_cmd_10Hz);
QuatError=MakeQuatCanonical(QuatError);

Att_Control_Error=2*QuatError(1:3);

% Rate Error
w_Body_est=deltatheta/Ts;
Att_Rate_Error=w_Body_cmd_1Hz-w_Body_est;

% PID
% Backward Euler Discrete-Time Integrator
Integral_State = Integral_State + Ki*Ts*Att_Control_Error;
for n=1:3
    if Integral_State(n) > Integral_State_Max
        Integral_State(n)=Integral_State_Max;
    elseif Integral_State(n) < Integral_State_Min
```

/Users/dwr27059/Documents/AeroCube/Attitude Control/Attitude Control.c

```c
  }

  c2_QuatError[3] = (1.0 - c2_mag2 / 8.0) + c2_mag2mag2 / 384.0;
  c2_QuatAdd(c2_QuatError, *c2_q_ECI2Body_cmd_1Hz, c2_q_ECI2Body_cmd_10Hz);
  for (c2_k = 0; c2_k < 3; c2_k++) {
    c2_mag2 = (*c2_Integrated_Gyro_Angle)[c2_k] -
      c2_Integrated_Gyro_Angle_prev[c2_k];
    if (c2_mag2 > c2_b_deltatheta_Max) {
      c2_mag2 -= c2_b_Integrated_Gyro_Angle_RolloverNeg;
    } else {
      if (c2_mag2 < c2_b_deltatheta_Min) {
        c2_mag2 += c2_b_Integrated_Gyro_Angle_RolloverPos;
      }
    }

    c2_Integrated_Gyro_Angle_prev[c2_k] =
      (*c2_Integrated_Gyro_Angle)[c2_k];
    c2_y[c2_k] = c2_mag2*c2_mag2;
    c2_theta[c2_k] = c2_mag2;
  }

  c2_mag2 = c2_y[0];
  for (c2_k = 0; c2_k < 2; c2_k++) {
    c2_mag2 += c2_y[c2_k + 1];
  }

  c2_mag2mag2 = c2_mag2 * c2_mag2;
  c2_sin_term = (0.5 - c2_mag2 / 48.0) + c2_mag2mag2 / 3840.0;
  for (c2_k = 0; c2_k < 3; c2_k++) {
    c2_QuatError[c2_k] = c2_theta[c2_k] * c2_sin_term;
  }

  c2_QuatError[3] = (1.0 - c2_mag2 / 8.0) + c2_mag2mag2 / 384.0;
  for (c2_k = 0; c2_k < 4; c2_k++) {
    c2_dv2[c2_k] = c2_Quat_ECI2Body_est[c2_k];
  }

  c2_QuatAdd(c2_QuatError, c2_dv2, c2_Quat_ECI2Body_est);
  for (c2_k = 0; c2_k < 4; c2_k++) {
    c2_QuatError[c2_k] = 0.0;
  }
```

MATLAB Code | Auto-generated C Code

Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Photos

1. Qatar (10/30/2012)
   - Use photos to align Earth sensor

2. Hurricane Sandy (10/26/2012)
   - Sweep LOS across ground target

3. Dubai (05/30/2013)
   - Point LOS at ground target
   - Verify pointing performance

4. Stars (02/06/2013)
   - Point LOS at inertial target off-nadir
   - Use stars to determine actual attitude

Darren.Rowen@aero.org
Control Analysis Department

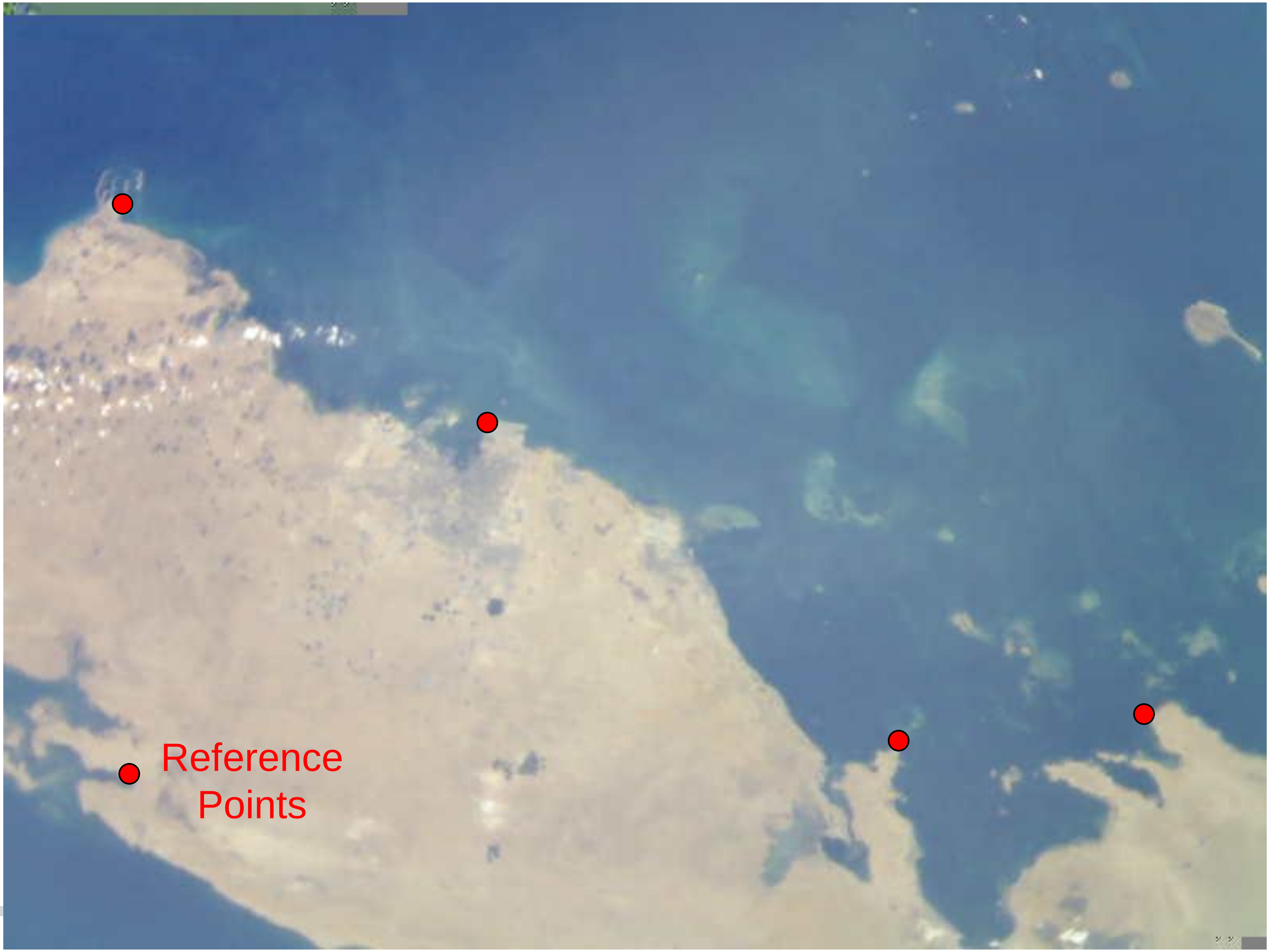
Photo 1:

Qatar

(Using Photos to Align Earth Sensor)

Darren.Rowen@aero.org
Control Analysis Department

Photo Qatar (AeroCube AC4 Narrow FOV)



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Earth Sensor Alignment

- Reference points
  - Recognizable geographic features (usually coastline)
  - Using pixel locations and camera alignment, calculate unit vectors in body frame: U_B
  - Using latitude/longitude and spacecraft ephemeris at time of photo, calculate unit vectors in ECI: U_I
- Solve for attitude TBI (DC matrix from ECI-frame to Body-frame)
  - Wahba problem: Solve for TBI that minimizes cost norm(U_B – TBI * U_I)
  - Solution (Markley)

```
X = [U1_B, U2_B, U3_B, U4_B, U5_B]*[U1_I, U2_I, U3_I, U4_I, U5_I]'  ;
[U,S,V] = svd(X) ; % U*S*V'  = X
d = det(U)*det(V) ;
TBI = U*diag([1 1 d])*V'  ;
```

- Use calculated TBI and Earth sensor measurements to align Earth sensor

Darren.Rowen@aero.org
Control Analysis Department

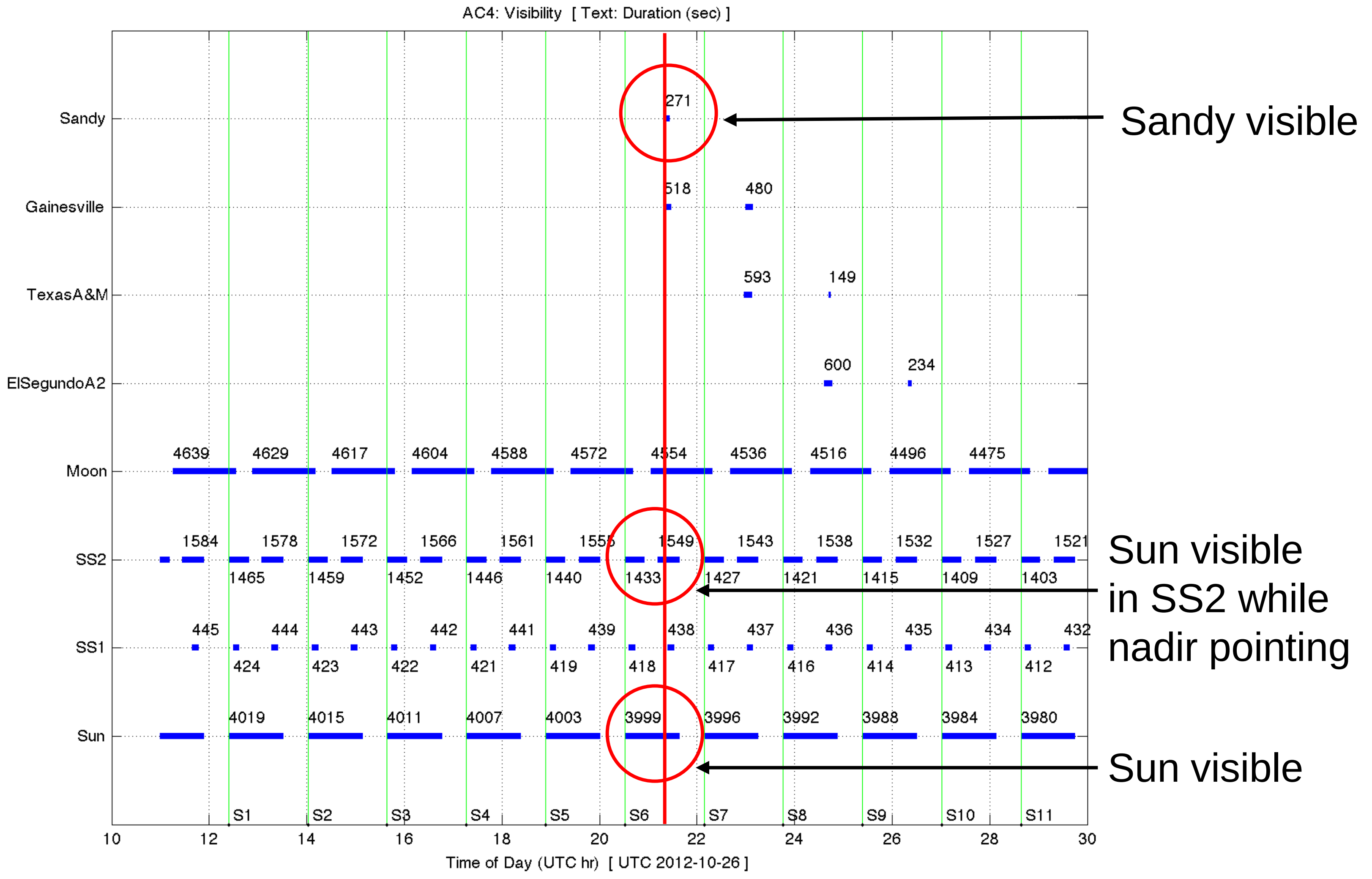
Photo 2:

Hurricane Sandy

(Sweep LOS Across GroundTarget)

Darren.Rowen@aero.org
Control Analysis Department

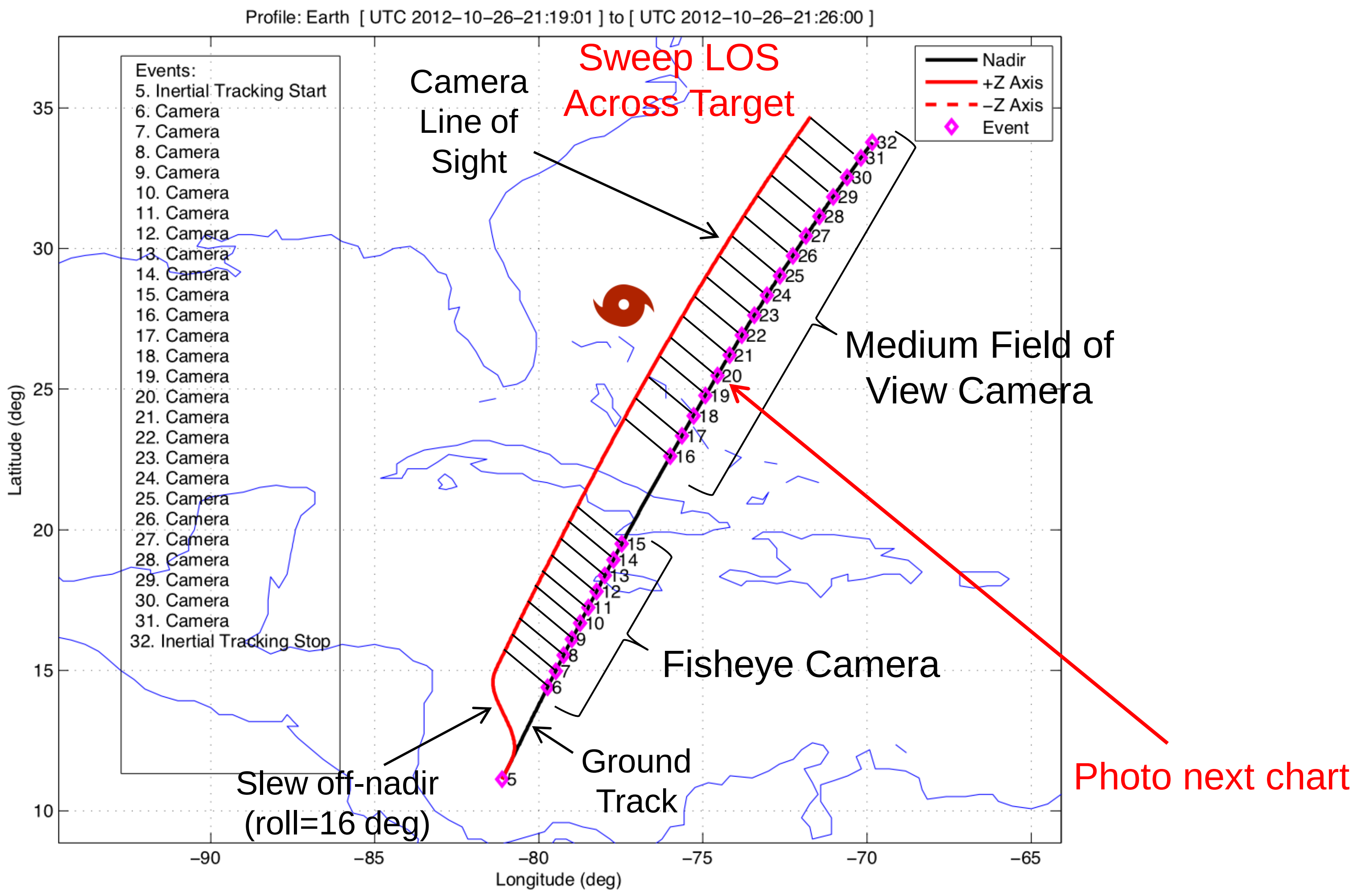
Ground Tool Used to Plan Mission



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

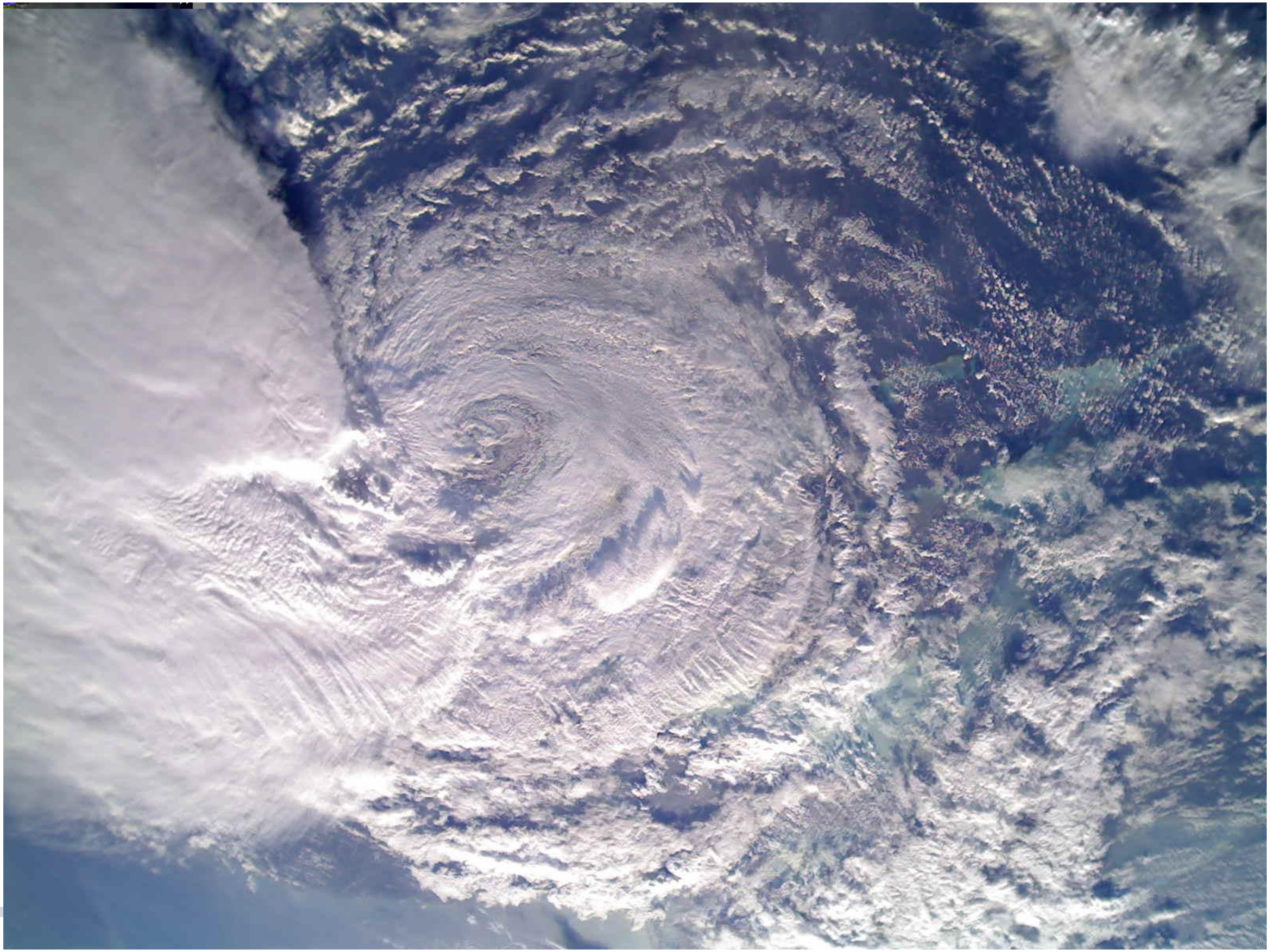
Line of Sight Profile



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Photo Hurricane Sandy (AeroCube AC4 Medium FOV)



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

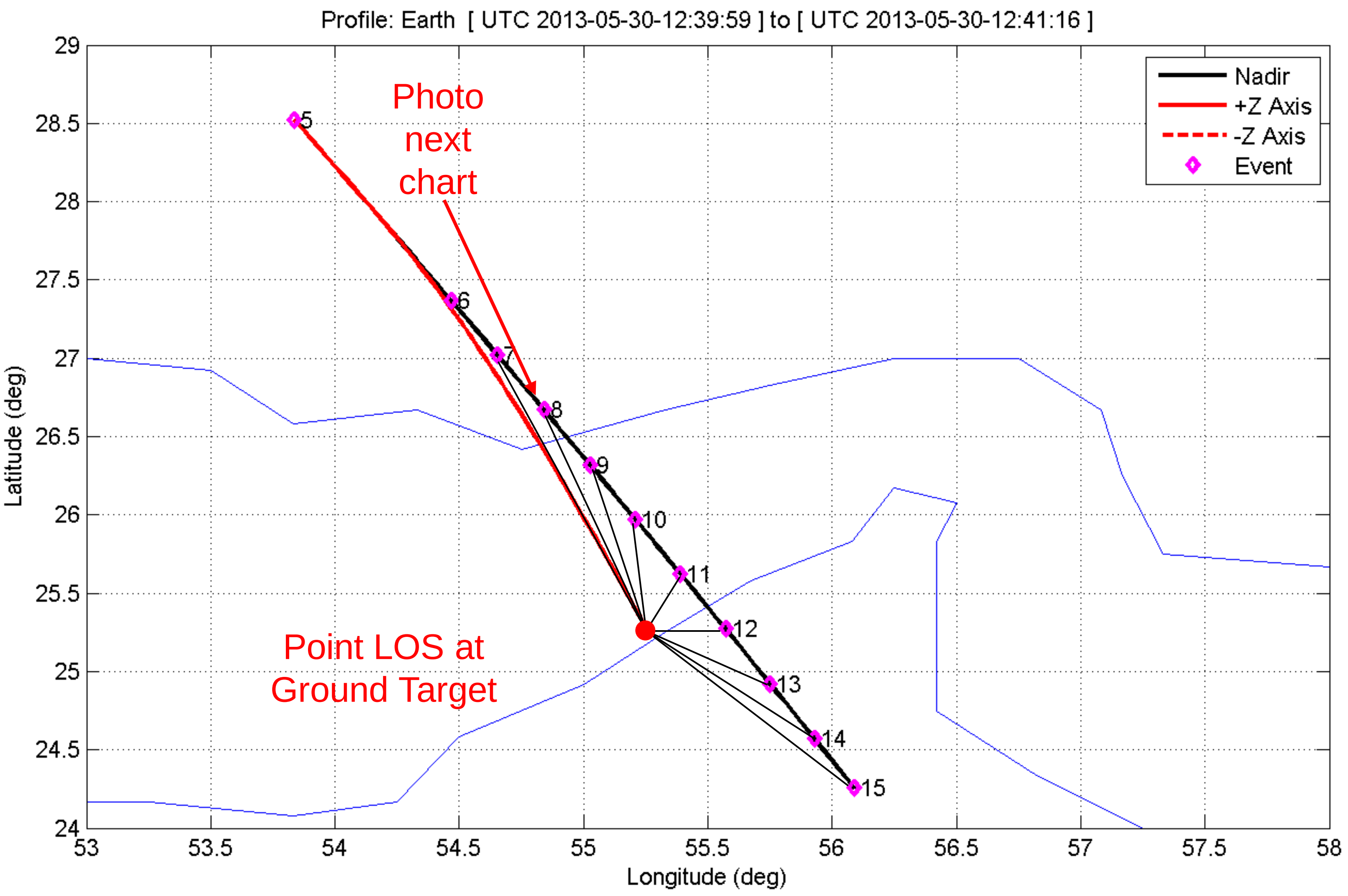
Photo 3:

Dubai

(Point LOS at Ground Target,
Verify Pointing Performance)

Darren.Rowen@aero.org
Control Analysis Department

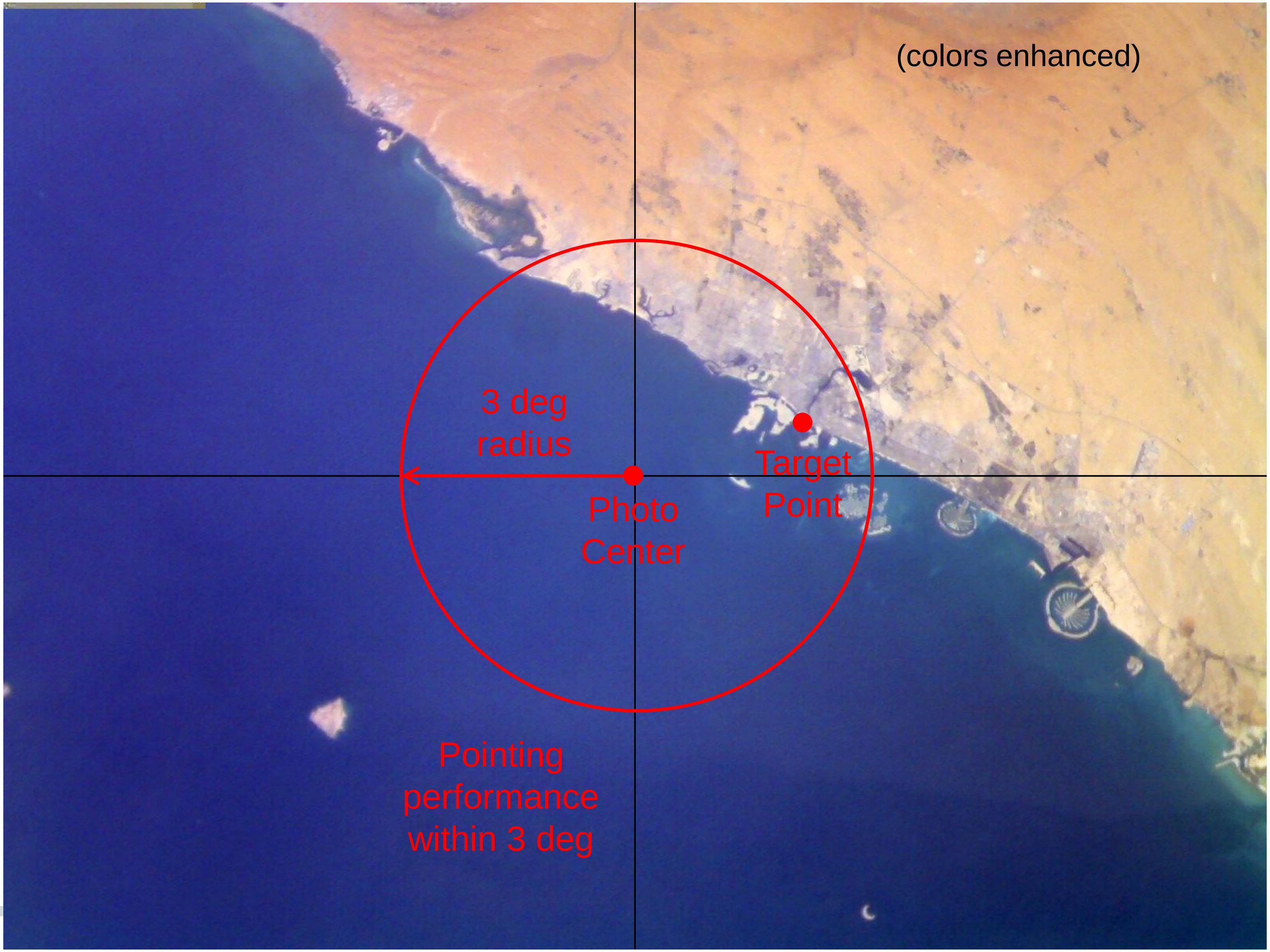
Line of Sight Profile



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Photo Dubai (AeroCube AC4 Narrow FOV)



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Photo 4:

Stars

(Point LOS at Inertial Target,
Use Stars to Determine Actual Attitude)

Darren.Rowen@aero.org
Control Analysis Department

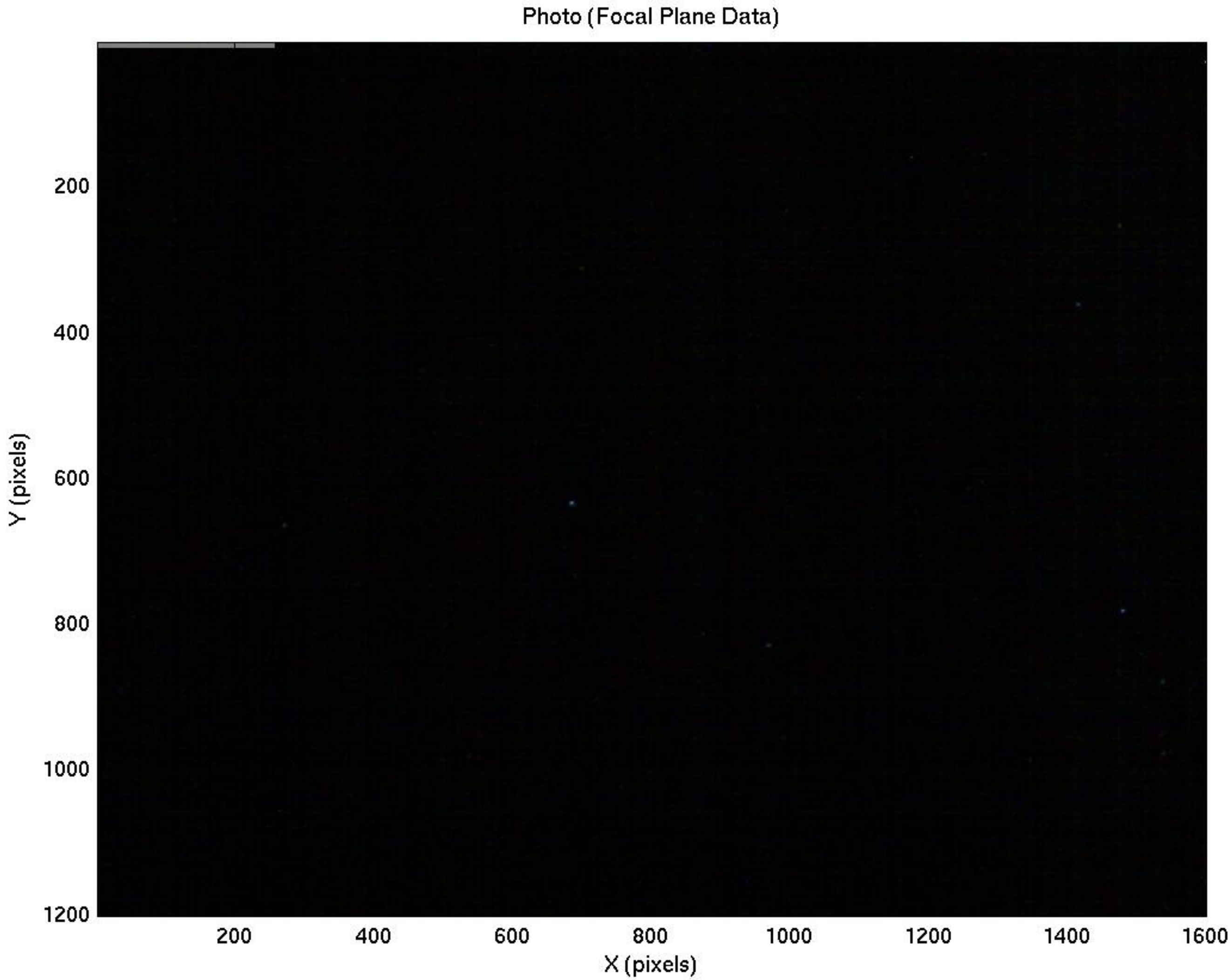
Use Camera as Star Tracker

- The next charts show results using photo of stars (02/06/2013)
  - *NFOV, 1600 x 1200 pixels, 16 x 12 deg (20 deg diagonal)*
  - *Exposure time unknown (around 1-2 sec auto-exposure)*
- Post-processing (on ground) of photo data
  - *Download compressed jpeg to ground*
  - *Apply median filter (row and column)*
  - *Find pixels above threshold*
  - *Find centroids of pixel clusters*
  - *Use lost-in-space algorithm to identify stars and determine attitude*
- Results
  - *Number of pixel clusters = 409 (based on selected threshold)*
  - *Use 6 brightest clusters in lost-in-space algorithm*
  - *FOV contains 31 catalog stars as dim as magnitude 6.00*
  - *26 catalog stars line up with a pixel cluster*
    - Dimmest magnitude match 5.98
    - No match for some catalog stars at magnitude 5.73+
    - Match for 22 brightest catalog stars down to magnitude **5.64**

Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Photo Stars (AeroCube AC4 Narrow FOV)

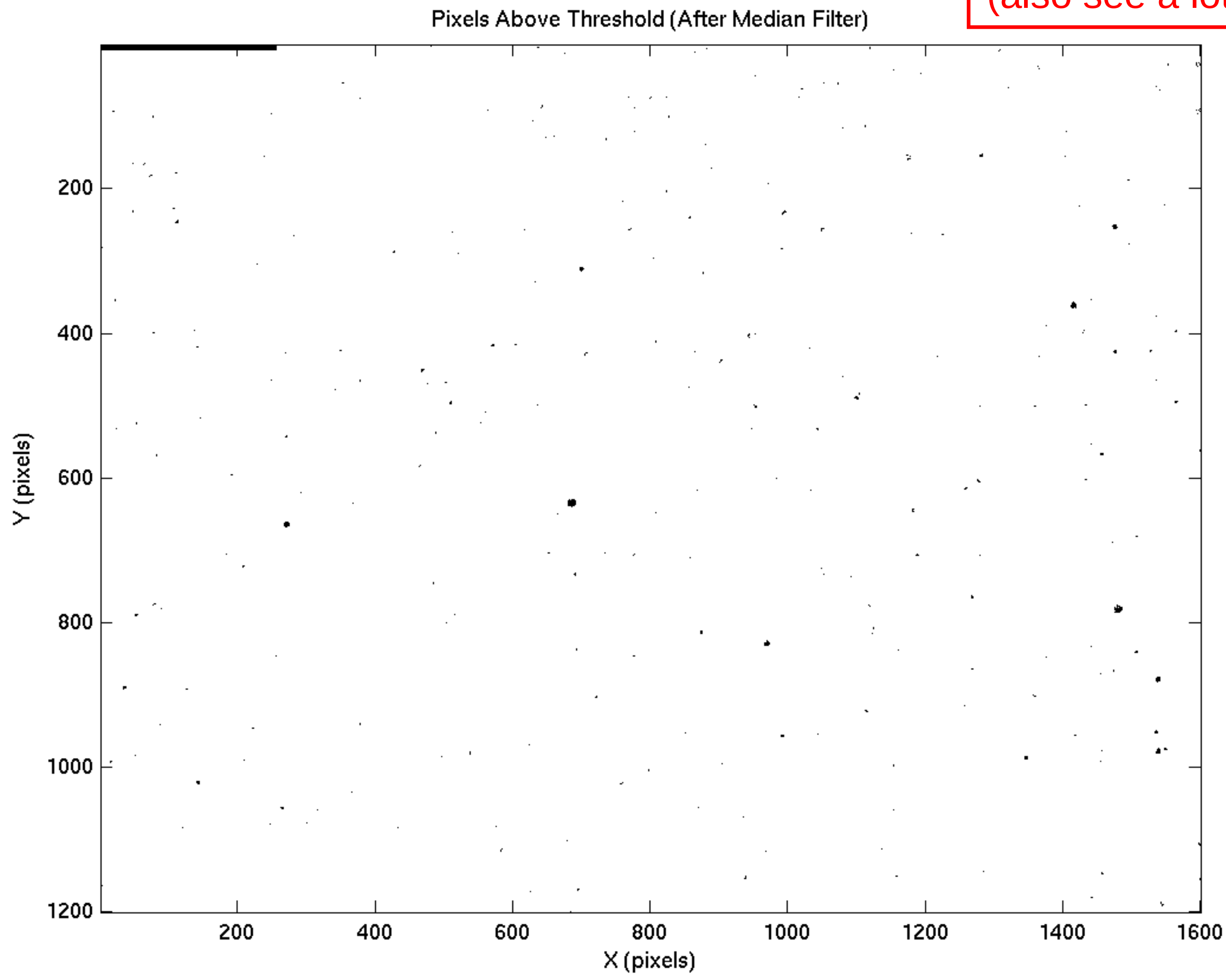


Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Apply Threshold

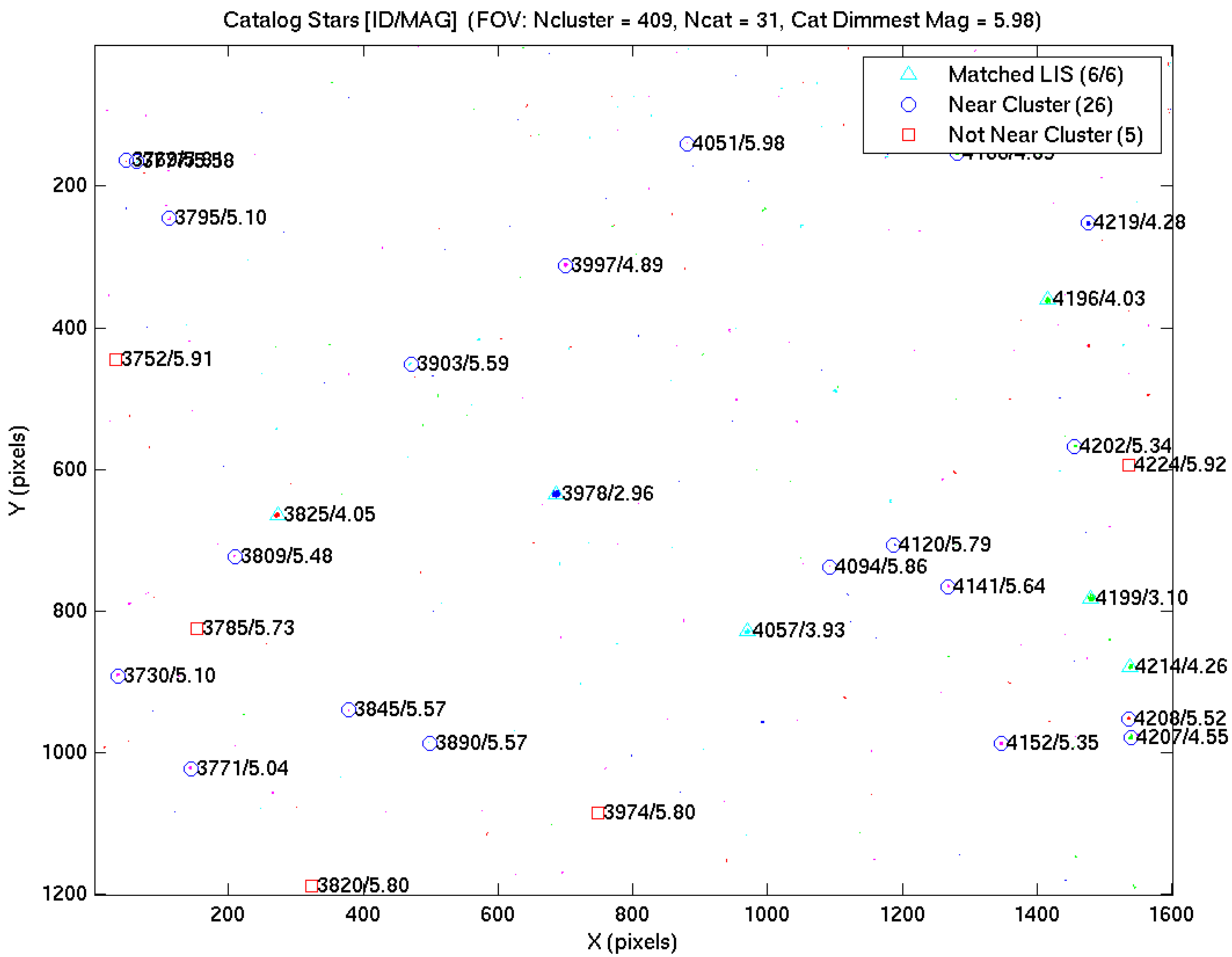
Here apply low threshold to see dimmest stars (also see a lot of noise)



Darren.Rowen@aero.org
Control Analysis Department

Catalog Stars and Pixel Clusters

Stars identified using lost-in-space algorithm



Darren.Rowen@aero.org
Control Analysis Department

AEROSPACE

Conclusions

- Successful 3-axis attitude determination and control
  - *Nadir pointing*
  - *Sweep across off-nadir ground target*
  - *Track ground target*
  - *Inertial pointing*

- COTS Camera Photos Very Valuable
  - *Sensor alignment / calibration*
  - *Pointing performance verification*
  - *Attitude determination (post-processing on ground)*

Darren.Rowen@aero.org
Control Analysis Department

Thank you