Modeling Coherent Radar Ground Echoes in Planetary Landing Simulations

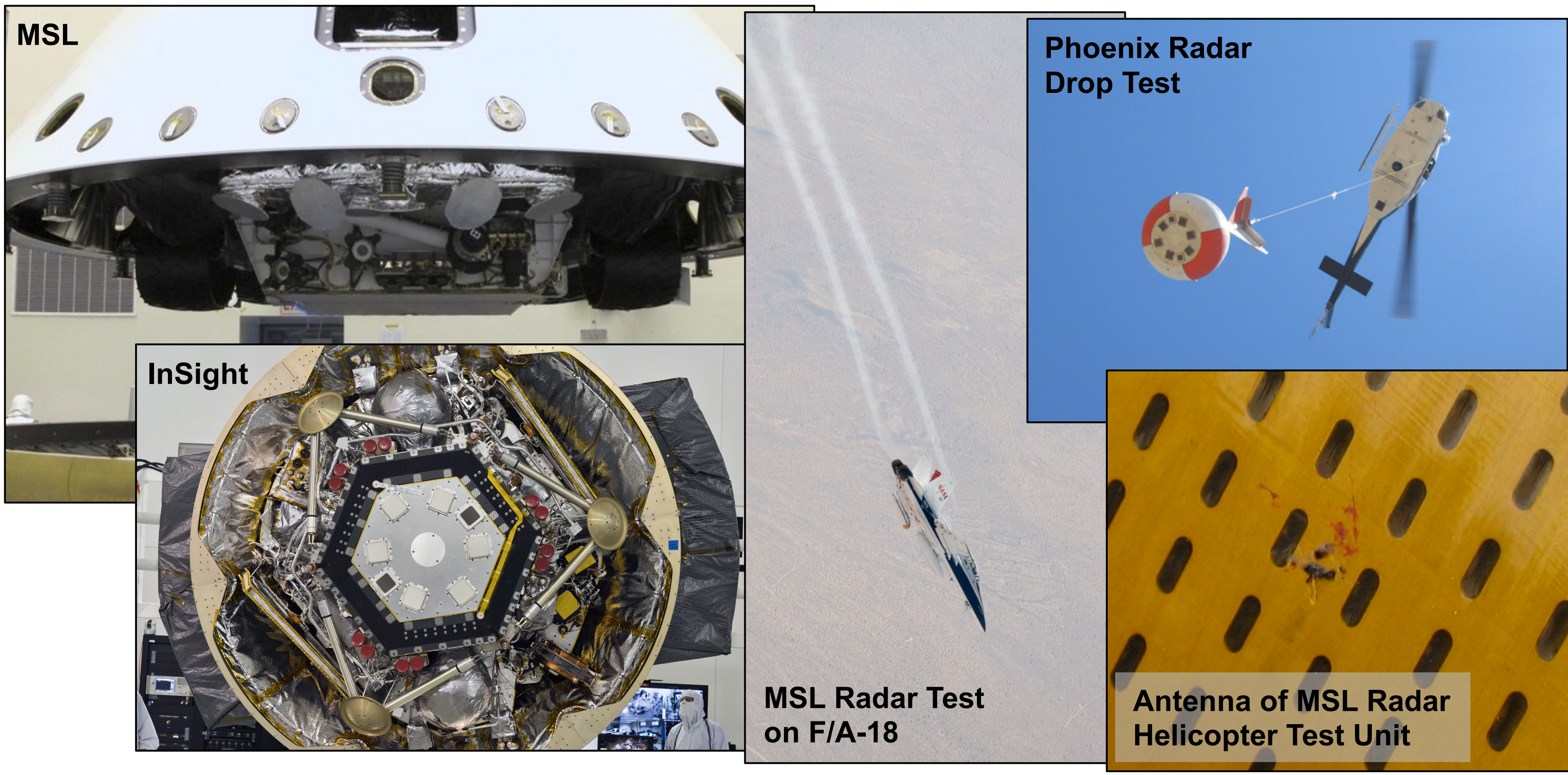
Curtis W. Chen

Radar Science and Engineering Section (334)

2019-02-28

Motivation

- Radar measurements of altitude and velocity are critical for planetary landings
- Simulation/modeling of landing radars is necessary because EDL system cannot be tested in completely flight-like manner



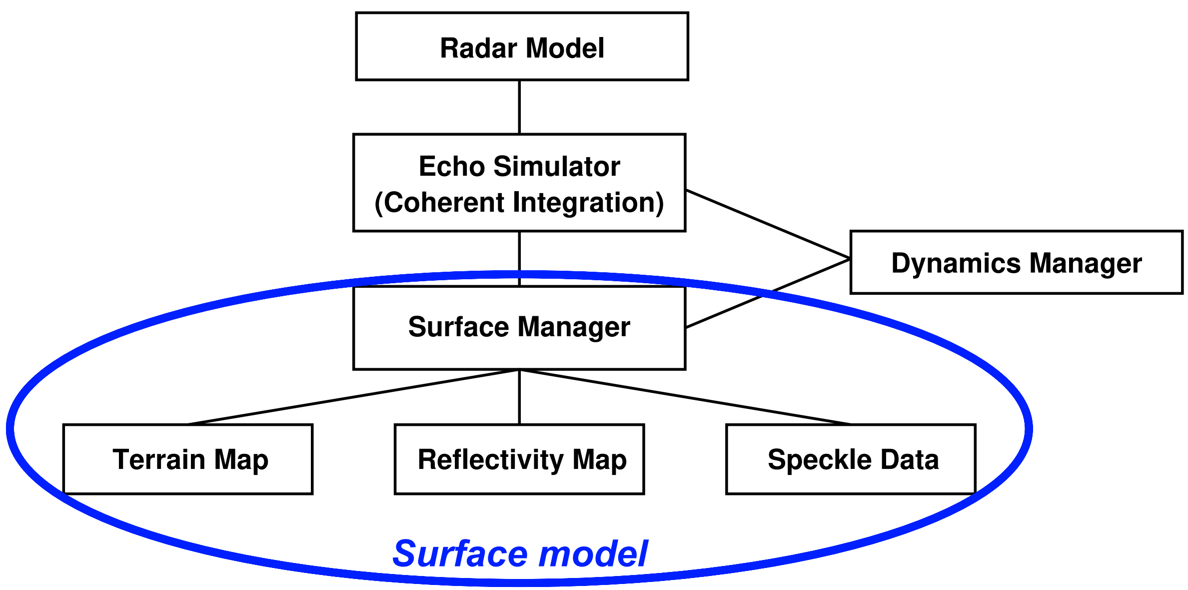
Unique Aspects of Landing Radars

- Landing radars are distinct from remote-sensing radars in many respects:
  - Must operate over ~3 orders of magnitude in range (meters to kilometers)
  - Must operate over highly dynamic trajectory and attitude profiles (e.g., swinging on parachute)
  - Dynamic range of signal power varies ~60 dB
  - Field of view of antenna changes dramatically
- Landing radars typically vary their operating parameters (transmit pulse width, pulse rate, etc.) autonomously based on their own measurements of range
- Radar echoes from surface must be modeled regardless of radar design (e.g., for both wide-beam Phoenix/InSight-style radar and narrow-beam MSL/M2020-style radar)

Radar Modeling Approaches

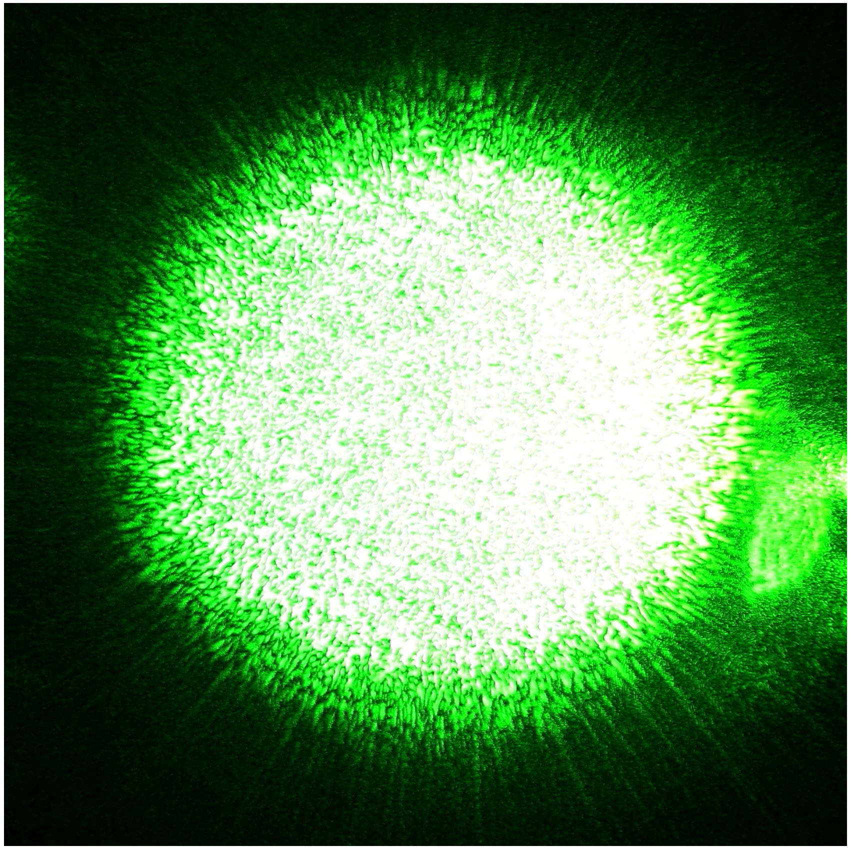
- Different radar modeling approaches are suited to different purposes:
  - Behavioral radar model can be used to assess sensitivities of GN&C/EDL system to known radar characteristics
  - High-fidelity simulations of radar measurement from first principles can provide predictive capability for validating radar design
- High-fidelity simulations must include model for echoes from ground surface
  - Same surface model can be used in different simulations for different radar hardware designs

- *Aldabra (phxlrsim) for Phoenix and InSight*
- *Sulcata for MSL and M2020,*



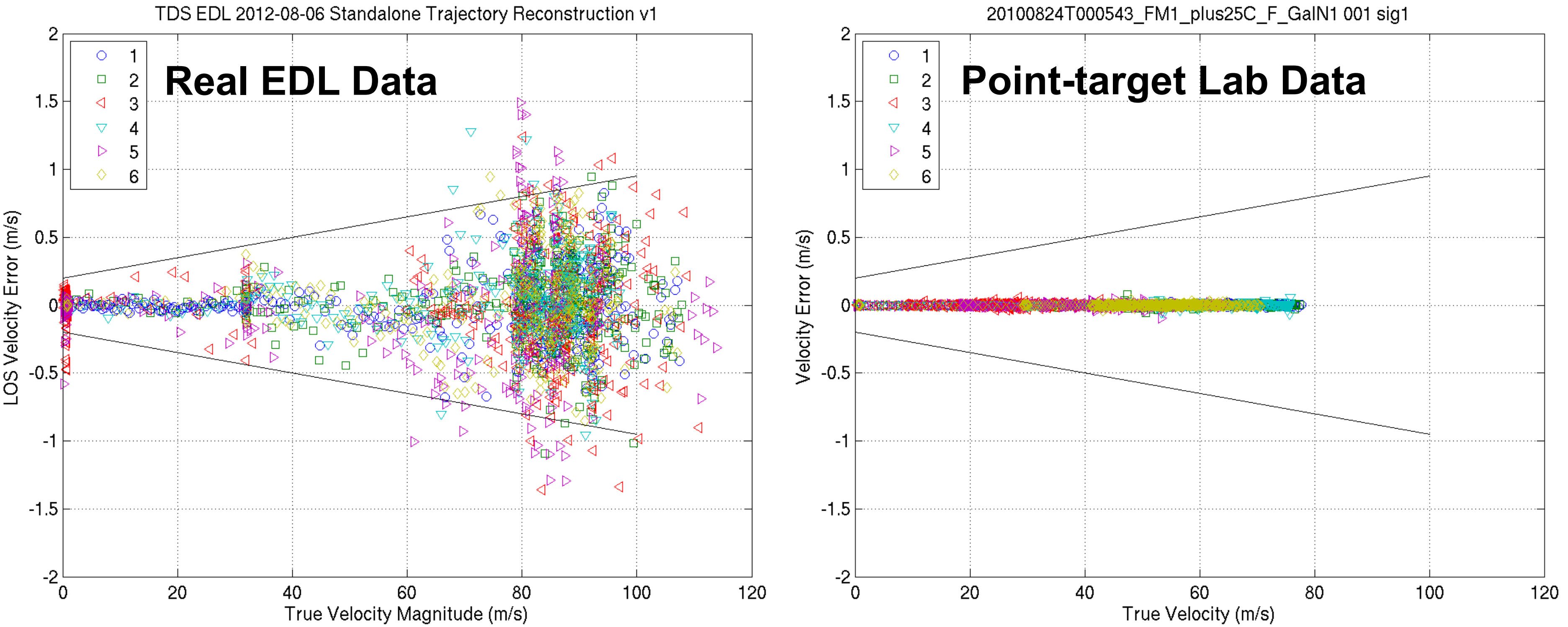
Surface Speckle

- Radars typically make coherent measurements (both magnitude and phase of signal echoes)
- Coherent echoes from ground are subject to speckle or fading
  - Ground surface can be modeled as collection of randomly distributed elemental scattering elements
  - Reflections from individual scattering elements sometimes add constructively and sometimes add destructively
- Speckle is due to randomness of surface (not noise of radar), so errors due to speckle cannot be overcome with increased SNR



Speckle pattern from green laser pointer

Distributed vs. Point Targets

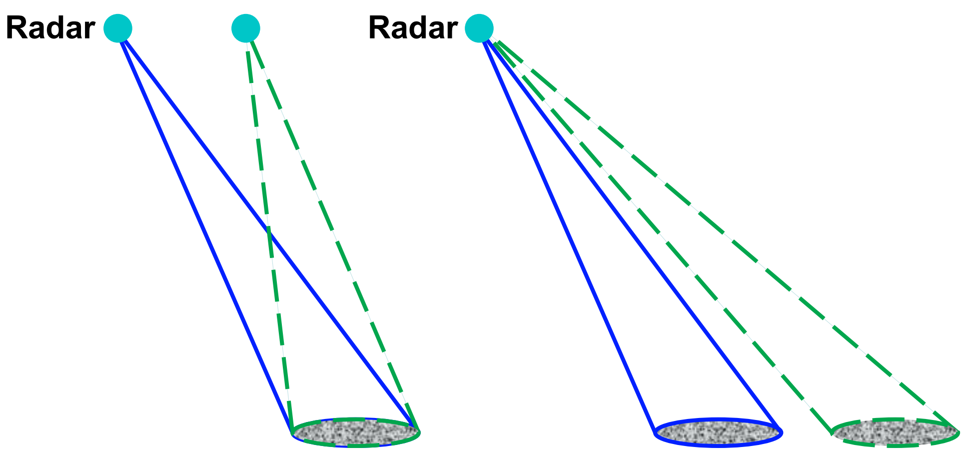


Black lines represent TDS 99% requirements

Point-target behavior of electronically generated lab targets grossly underestimate error characteristics

Speckle Variations

- Speckle is determined by viewing geometry; variations will depend on both ground location and radar location

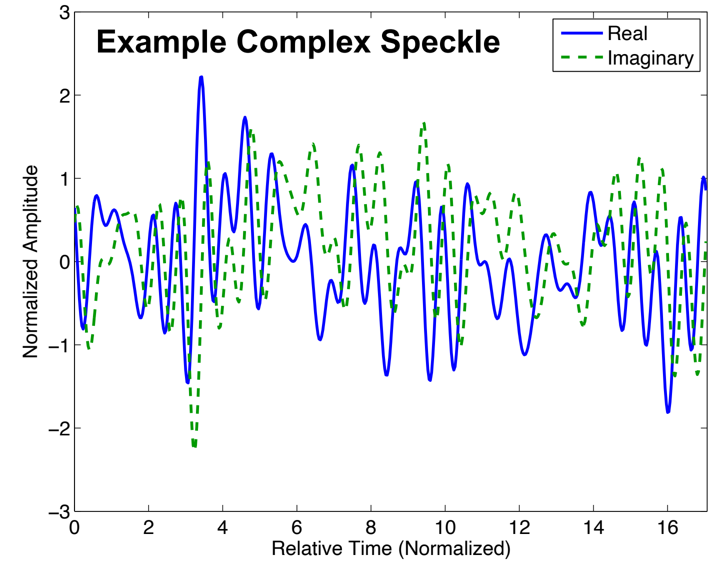


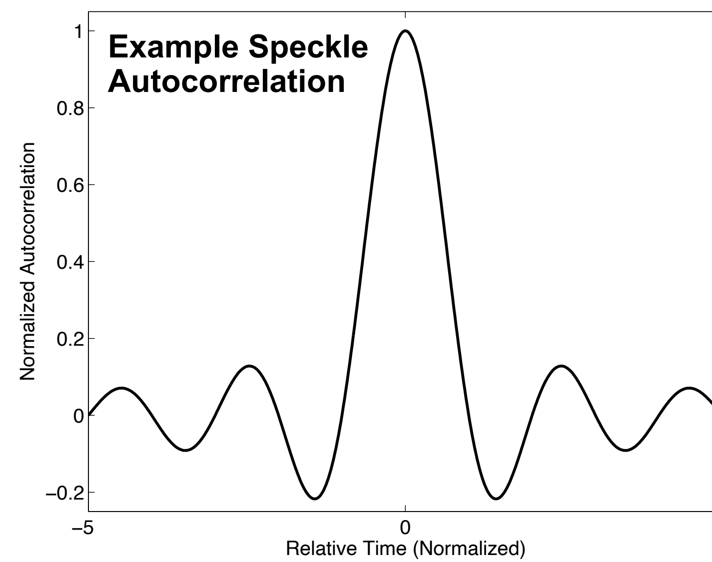
Beam footprint on ground

Echoes from given ground location will exhibit random amplitude and phase if viewed from different radar positions

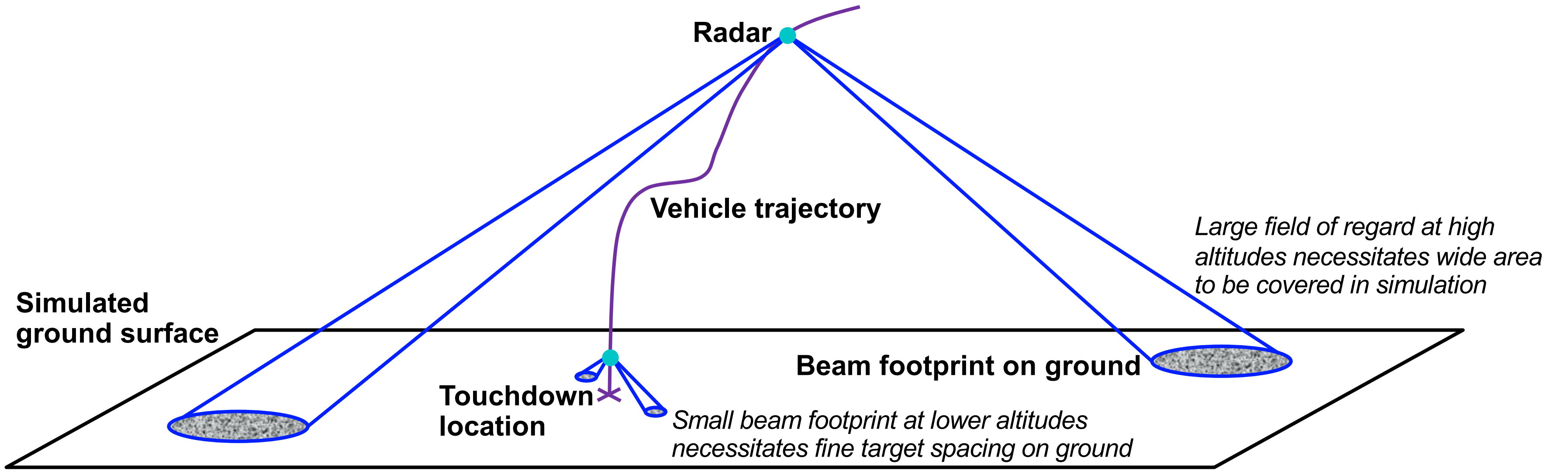
Beam footprints on ground

Echoes from different ground locations will exhibit random amplitude and phase if viewed from fixed radar position





Brute-Force Speckle Model



- Common, brute-force speckle modeling approach is to simulate large number of individual point scatterers on ground
  - Excellent representation of fundamental mechanisms and speckle behavior
  - Simple
  - Computationally demanding:
    - **Fine target spacing needed for low altitudes over wide area needed for high altitudes makes brute-force approach impractical for landing scenarios**
    - Using different scenes for different parts of descent would introduce discontinuities and potential artifacts in radar performance

Total Radar Echo

- Total echo is sum of complex contributions from all cells, weighted by terms of radar equation:

$$a(\rho) = C_a \sum_k b_k \left[\frac{g(\hat{\imath}_{Ak})}{\rho_k} \right]^2 \exp\left(-j \frac{4\pi}{\lambda} \rho_k \right) \chi(\rho - \rho_k)$$

Total echo as function of range

Amplitude scaling term that captures constant or slowly varying terms (transmit power, etc.)

Summation over all cells indexed by *k*

Backscatter contribution of cell

Complex antenna pattern and inverse range term

Phase from range to cell center

Range point-target response

$$b_k = s_k \sqrt{A_k \sigma_k^0}$$

Normalized speckle term from lookup table

$$\left\langle |s_k|^2 \right\rangle = 1$$

Cell area

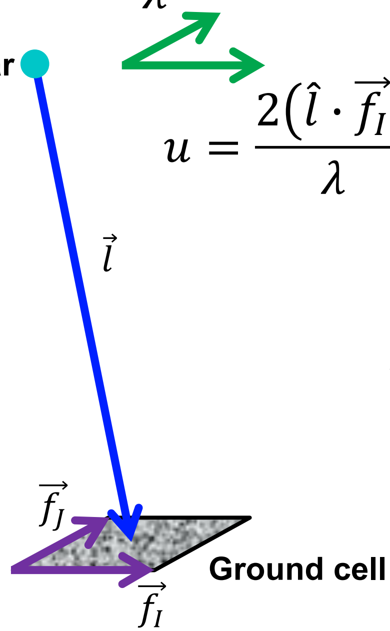
Cell backscatter coefficient

Coherent Facet Model

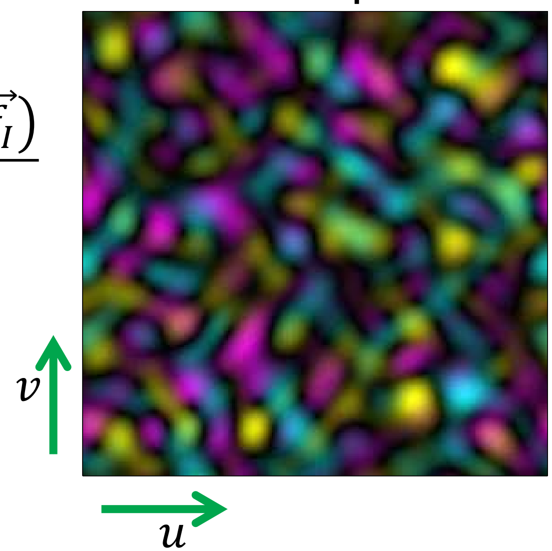
- Model surface as collection of coherent rectangular facets or cells
- Van Cittert-Zernike theorem implies that autocorrelation of speckle as function of viewing angle is Fourier transform of cell shape (same idea behind InSAR geometric or baseline decorrelation)

$$v = \frac{2(\hat{l} \cdot \overrightarrow{f_J})}{\lambda}$$

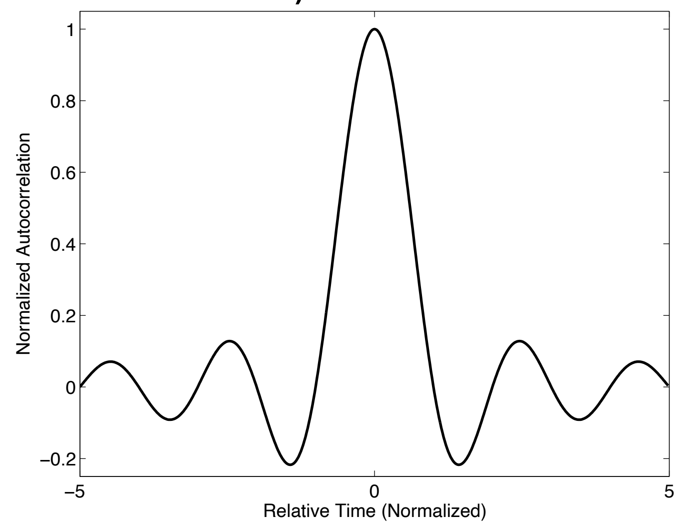
$$u = \frac{2(\hat{l} \cdot \overrightarrow{f_I})}{\lambda}$$



Complex speckle field from ground cell as function of radar horizontal position



Speckle autocorrelation function (for each dimension, normalized)



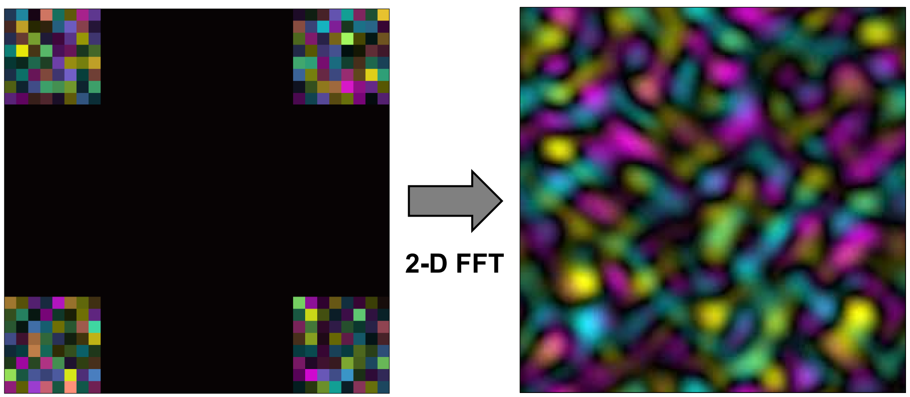
$$R_S(u, v) = \mathrm{sinc}(u)\,\mathrm{sinc}(v)$$

2-D sinc function is Fourier transform of rectangular shape of ground cell (assuming uniform illumination)

- Surface echo is linear combination of contributions from all cells, weighted by backscatter, antenna pattern, range response, etc.

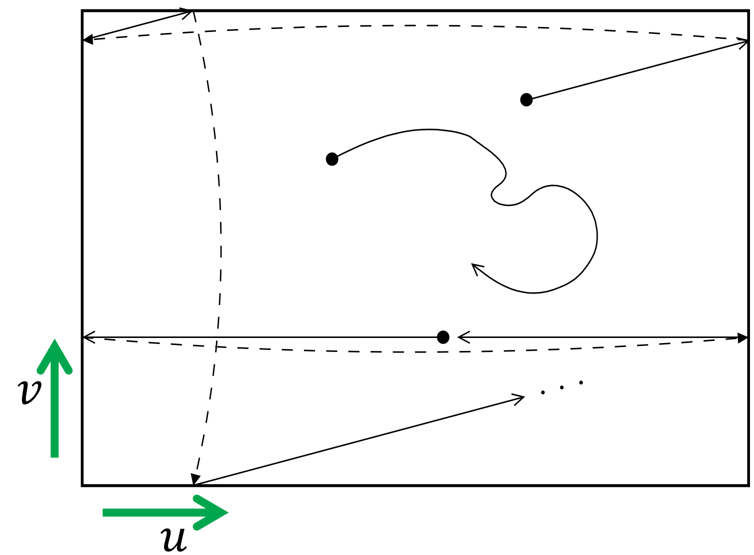
Speckle Lookup Model

- Model speckle with lookup table of oversampled Gaussian random values that wrap around at edges
- Each cell has its own random seed into same lookup table
- Wrapped indices in table for each cell are computed from look vector to cell



Zero padded array of independent complex Gaussian random values

Oversampled complex values with proper autocorrelation

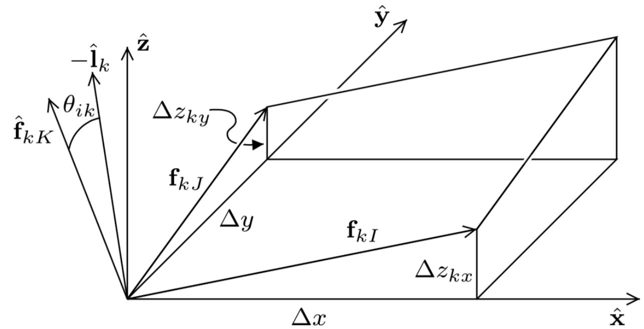


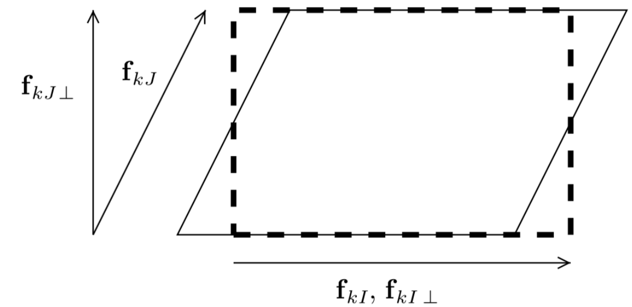
$$u = \frac{2(\hat{l} \cdot \vec{f_I})}{\lambda} \qquad v = \frac{2(\hat{l} \cdot \vec{f_J})}{\lambda}$$

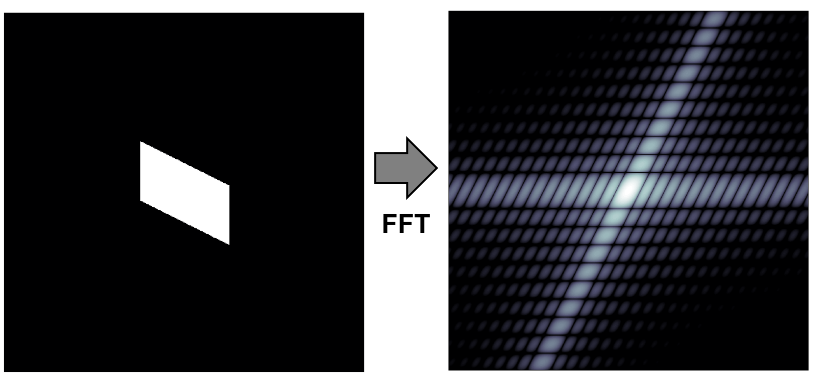
Repetition of cell speckle realizations from table wrapping does not cause problems in practice because different cells wrap at different rates, so total echo still behaves randomly

Facet Geometry and Implementation

- For each facet, store
  - Center location
  - Vectors of facet edges; can compute area, incidence angle, and speckle indices
  - Speckle array parameters
  - Backscatter power parameters
- If surface is not flat, facets may not be perfectly rectangular
  - Approximate with equivalent rectangles when computing speckle indices
    - Difference mainly affects sidelobes of autocorrelation function, not width of main lobe, so speckle still behaves as desired
    - Equivalent rectangular facets may not give continuous surface, but objective is to model representative rather than exact surface
  - True surface will not really be faceted anyway

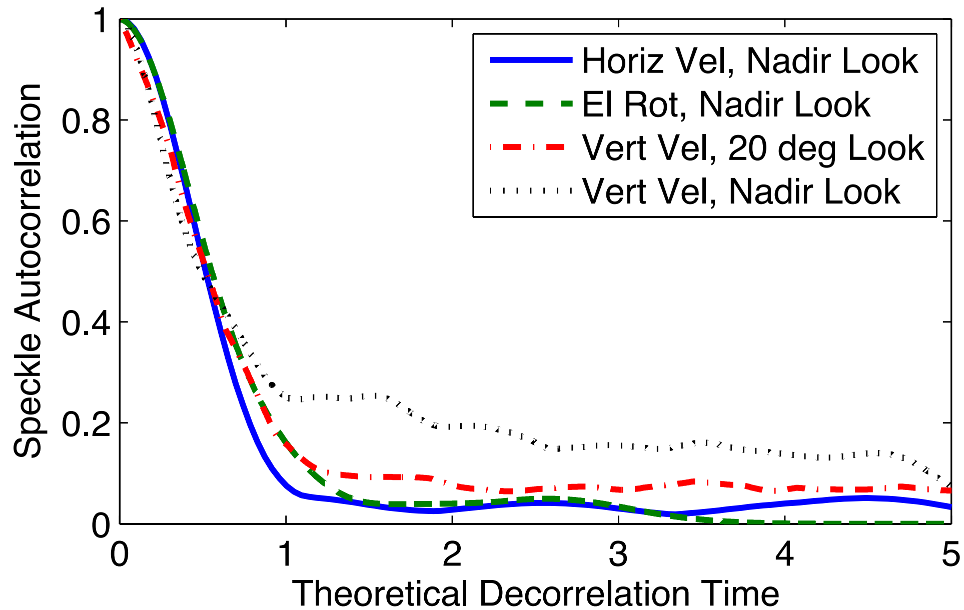


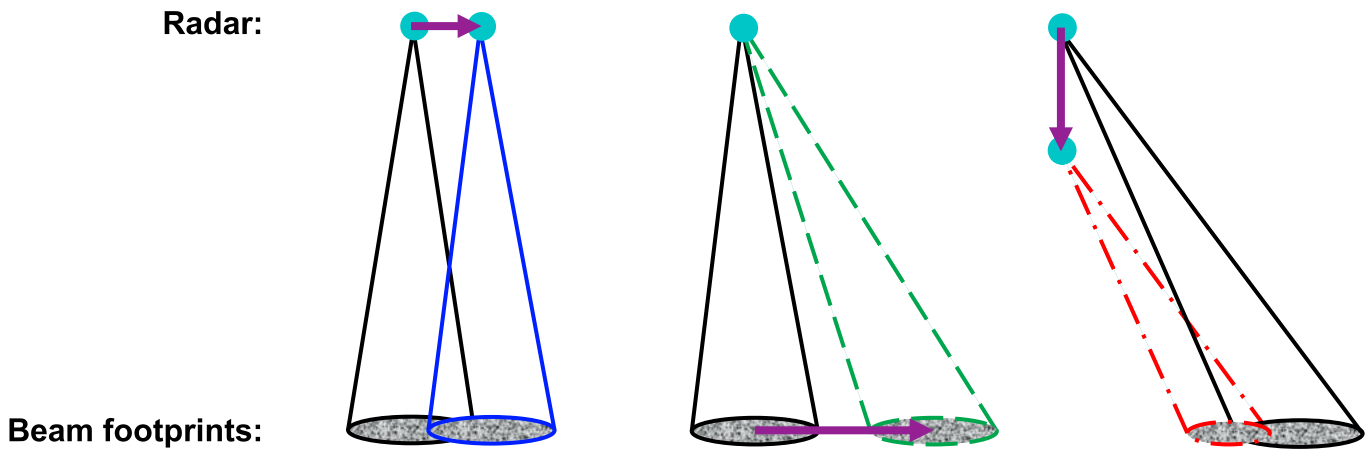




Statistical Validation of Model

- Numerical autocorrelation results from simulation show expected behavior for canonical cases:
  - Horizontal velocity, nadir look direction
  - Elevation rotation, nadir look direction
  - Vertical velocity, 20˚ look direction
  - Vertical velocity, nadir look direction





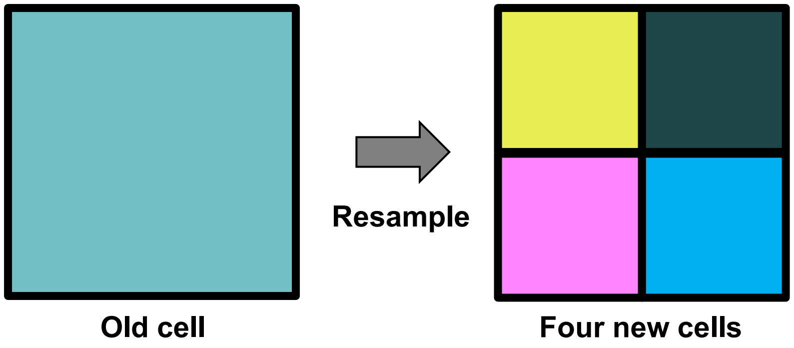
Speckle decorrelates after radar moves half antenna width

Speckle decorrelates after radar rotates by beamwidth

Speckle decorrelation distance equivalent to InSAR critical baseline

Speckle is bit more complicated

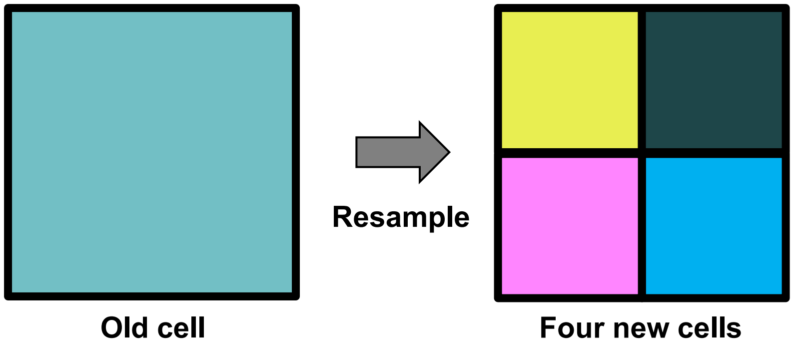
Scene Resampling

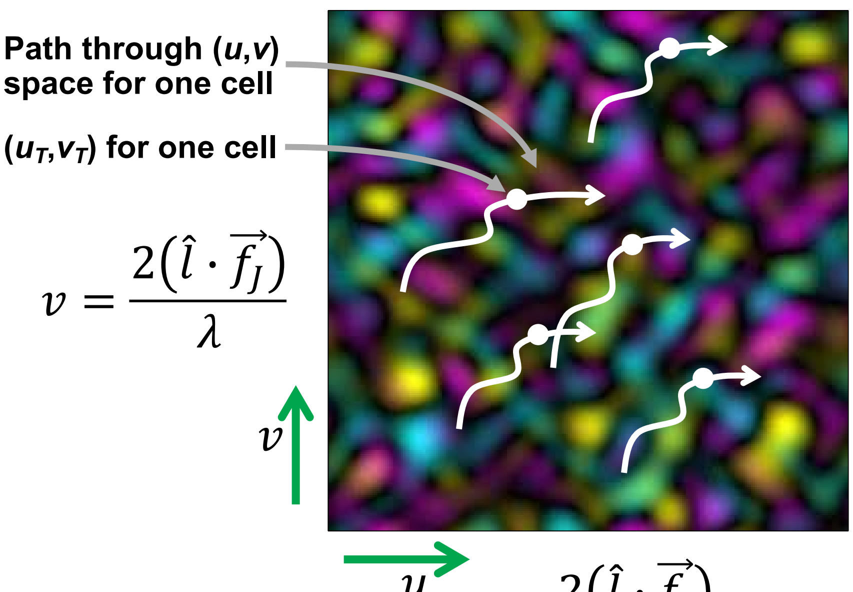


Speckle term provides independent multiplicative factor for each cell, so override speckle values from lookup table with special transition values that are selected to make sum of echoes from new cells match echo from old cell when viewed from location of radar at time of scene resampling

- Cell model is not necessarily more computationally efficient than brute-force approach if reasonable number of targets is maintained for each
- Advantage of cell model is that it allows for seamless resampling of scene
  - Resample to increase spatial density of targets as vehicle descends to maintain proper number of targets per footprint
  - Avoid discontinuities in speckle when resampling occurs

Speckle Transition Approach





$$v = \frac{2(\hat{l} \cdot \vec{f_J})}{\lambda}$$

$$u = \frac{2(\hat{l} \cdot \vec{f_I})}{\lambda}$$

- New cells after resampling will have independent random seeds into speckle lookup table
- Scene resampling at time *t* occurs at particular transition index (u_T, v_T) that is different for each cell
- Compute speckle override value s_T for each new cell that gives smooth speckle through resampling
- Compute speckle for new cells as:

```
if ((u,v) is near (uT,vT))   // within correlation width
   s = sT                    // use transition speckle value
else
   s = s(u,v)                // from lookup table
```

- Speckle override value may be used again as path through (*u*,*v*) space wraps, but it behaves like any other random speckle value

Speckle Transition Formulation

$$a(\rho) = C_a \sum_k b_k \left[\frac{g(\hat{l}_{Ak})}{\rho_k}\right]^2 \exp\left(-j\frac{4\pi}{\lambda}\rho_k\right)\chi(\rho - \rho_k)$$

Total echo

$$b_k = s_k \sqrt{A_k \sigma_k^0}$$

Cell reflection term

Speckle term

$$a(\rho) = C_a \sum_k h_k s_k \qquad h_k = \sqrt{A_k \sigma_k^0}\left[\frac{g(\hat{l}_{Ak})}{\rho_k}\right]^2 \exp\left(-j\frac{4\pi}{\lambda}\rho_k\right)\chi(\rho - \rho_k)$$

At transition

Echo from old scene (before resampling)

$$\sum_k h_k s_k = \sum_{k'} h_{k'} s_{k'}$$

Echo from new scene (after resampling)

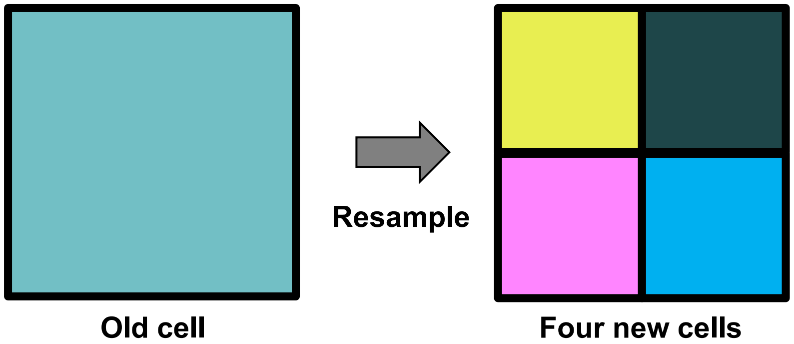
Choose transition speckle values to enforce equality at transition time

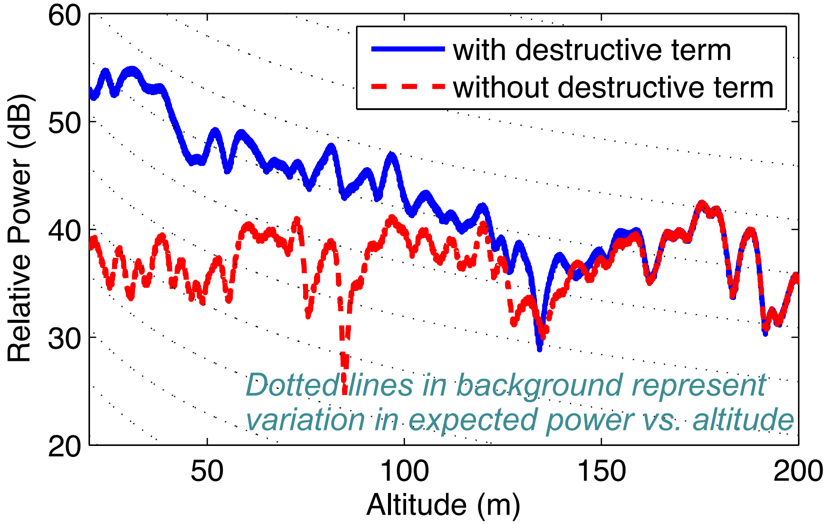
$$s_{Tk'} = s_{Tck'} + s_{Tdk'}$$

Constructive term gives speckle continuity at transition time

Destructive term gives zero contribution at transition point but gives randomness away from transition time

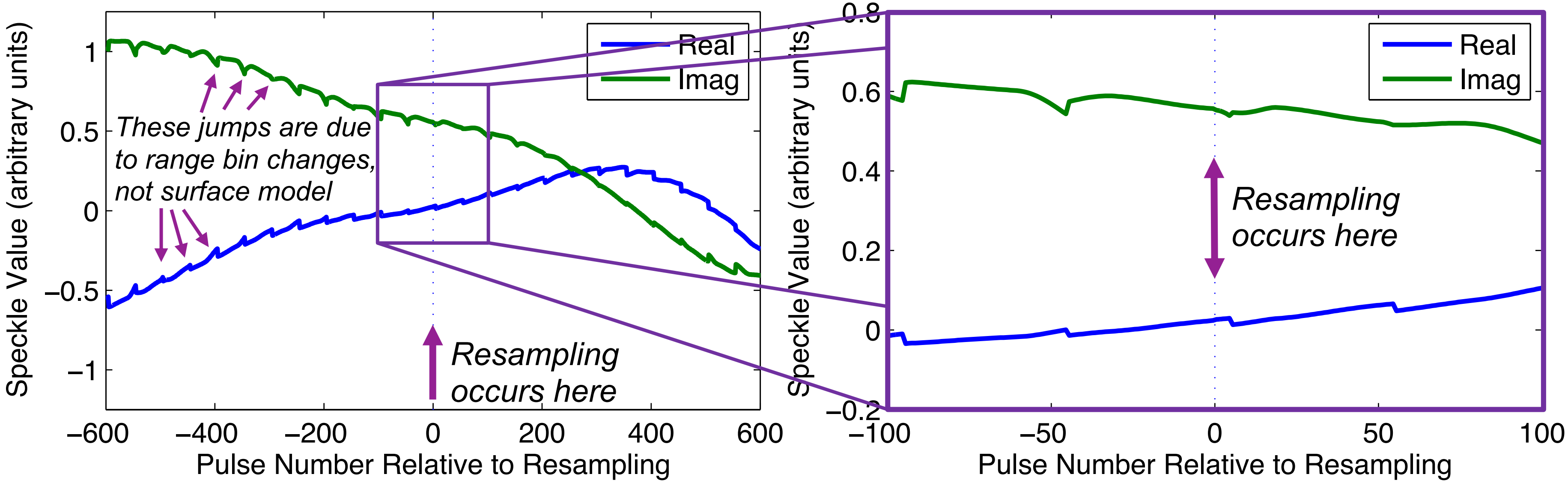
Speckle Transition Values





- Simplest approach to ensure speckle continuity: compute s_T for each new cell so that new cell contribution is 1/*n* of old cell contribution
  - Total echo is continuous at transition, but speckle between new cells becomes correlated such that speckle statistics no longer behave properly after transition time
  - Call this "constructive" speckle transition term
- Must also include "destructive" speckle term
  - Gives no net contribution when weighted by radar-equation terms (antenna gain, range response, etc.) at transition time, so constructive term gives speckle continuity
  - Gives extra randomness when different weighting of radar-equation terms is applied after transition time

Scene Resampling Example



- Example complex echo from Sulcata simulation with velocity and beam pointing straight down
  - Scene is resampled to 1.5x finer cell spacing in each dimension at 0 on horizontal axis
  - Phase discontinuity due to scene transition is not discernible from other phase variations (much less than 1˚)
  - Jumps every 50 pulses are due to discretization of echo into (oversampled) range bins (echo moves between bins as range closes)

Model Limitations

Jet Propulsion Laboratory
California Institute of Technology

- Limitations of model as currently implemented:
  - Model is not polarimetric
  - Model assumes homogeneous rough surface; does not handle penetration, multiple scattering, Bragg scatter, etc.
  - Does not handle shadowing
- Generalizations of model are possible
- Limitations have not impacted applicability of model in practice to date for Phoenix, MSL, InSight, or M2020

Model Usage

- Surface model has been used in different simulations:
  - Aldabra (phxlrsim) for Phoenix and InSight
    - Model originally developed for Phoenix
    - Software re-used for InSight with no significant changes
  - Sulcata for MSL and Mars 2020
    - Theoretical model for surface is common between Sulcata and Aldabra
    - Software implementation of surface model reorganized for MSL to better manage scenes with TDS beam geometry
    - Software re-used for Mars 2020
- Typical Sulcata run time to reproduce real MSL landing:
  - *O*(2 hours) on ~2016 MacBook Pro
  - *O*(12 hours) on single node of ~2012 JPL galaxy or kelvin clusters

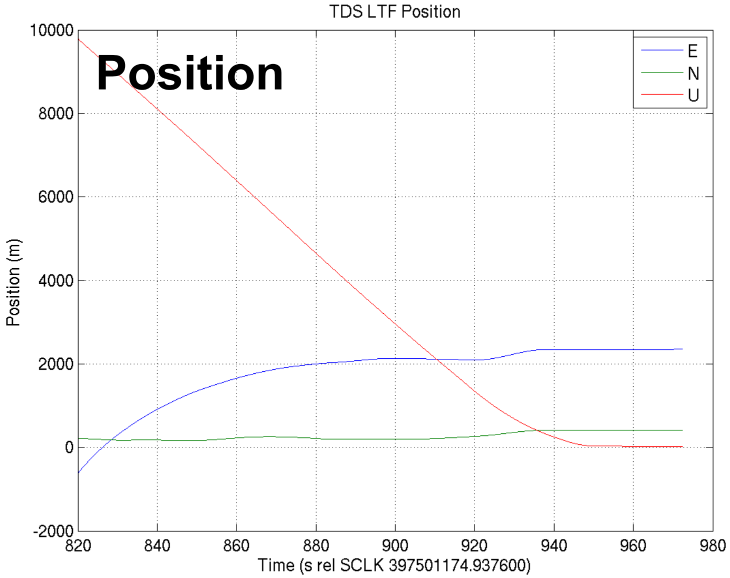
Comparisons to MSL Data

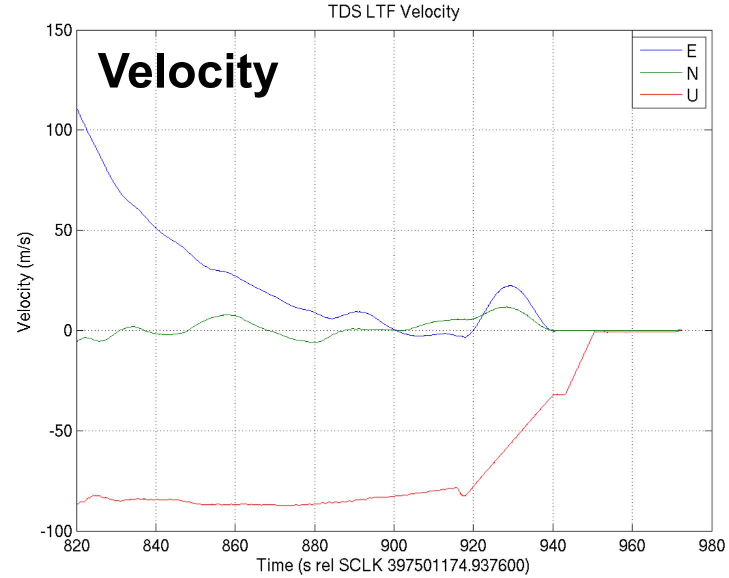
Jet Propulsion Laboratory
California Institute of Technology

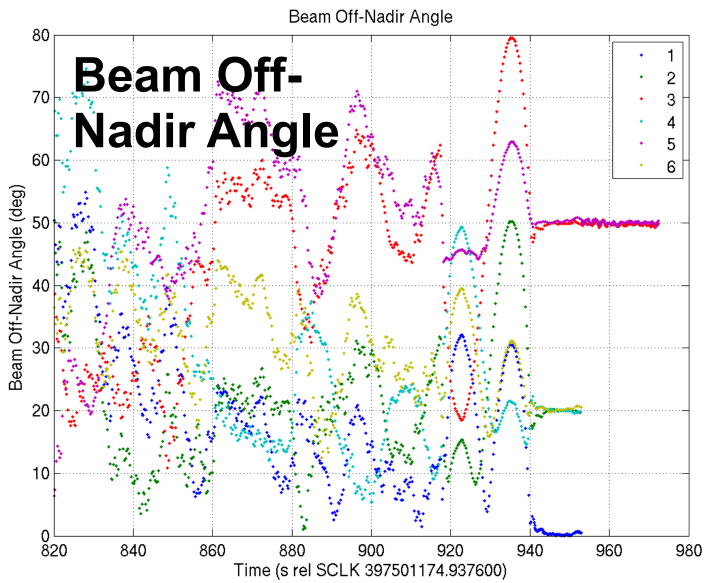
- Following slides compare Sulcata simulations to actual radar data from MSL landing on Mars and MSL helicopter testing of EM radar unit
  - Real radar measurements compared to “truth” from trajectory reconstruction to compute errors for real radar data
  - Truth trajectory used as input to Sulcata, then simulated radar measurements compared to simulation truth to compute errors for Sulcata
- Simulations were not tuned to match real data except to use surface reflectivity inferred from real radar data

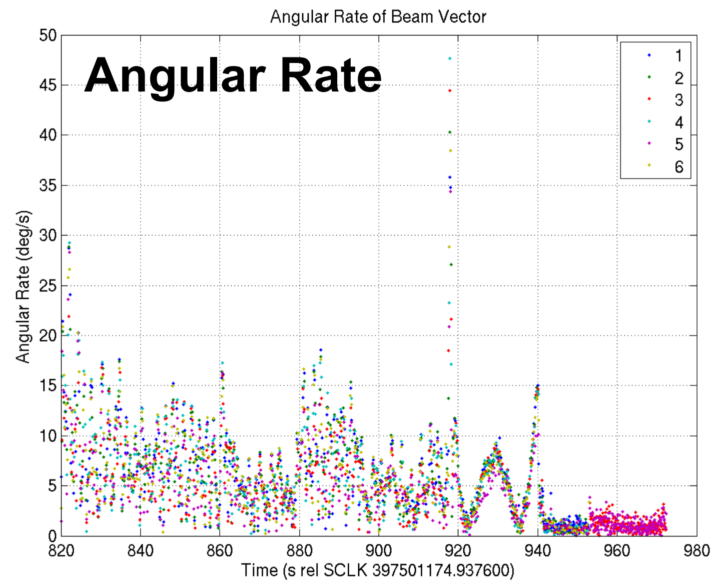
MSL EDL Trajectory

Jet Propulsion Laboratory
California Institute of Technology

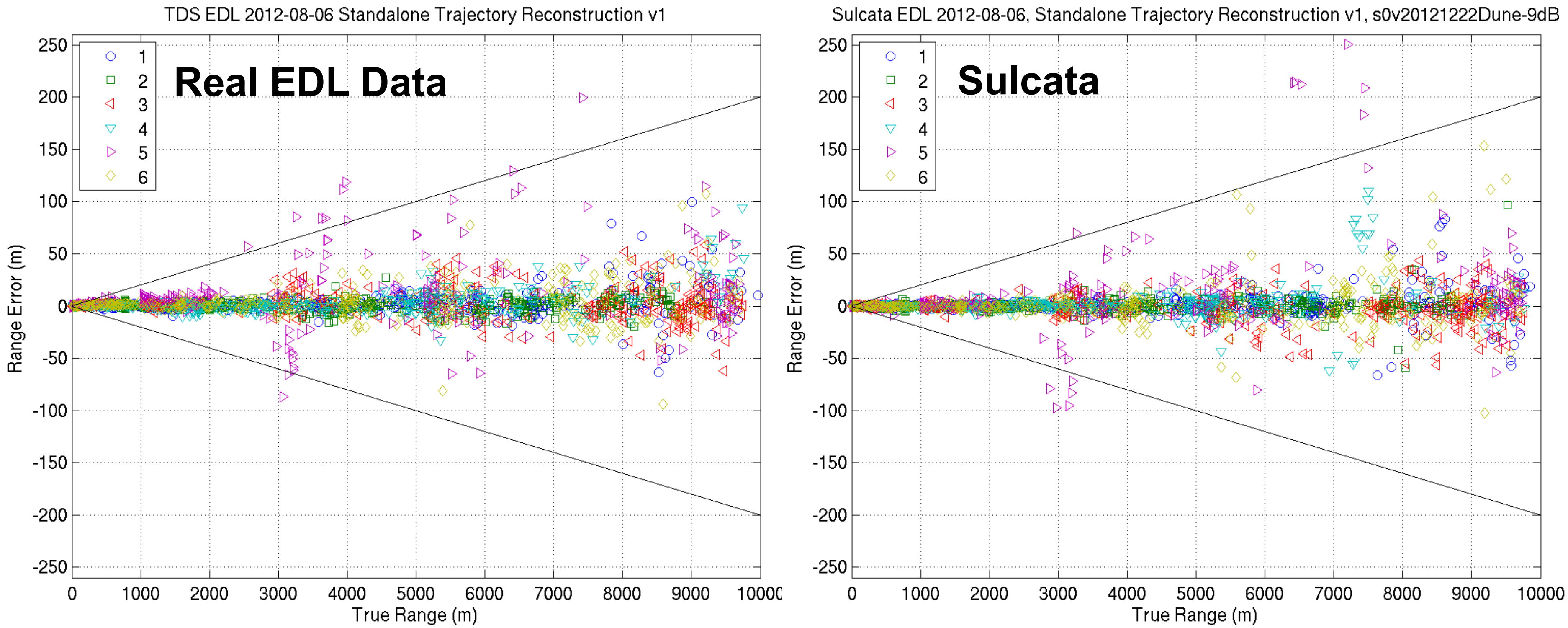








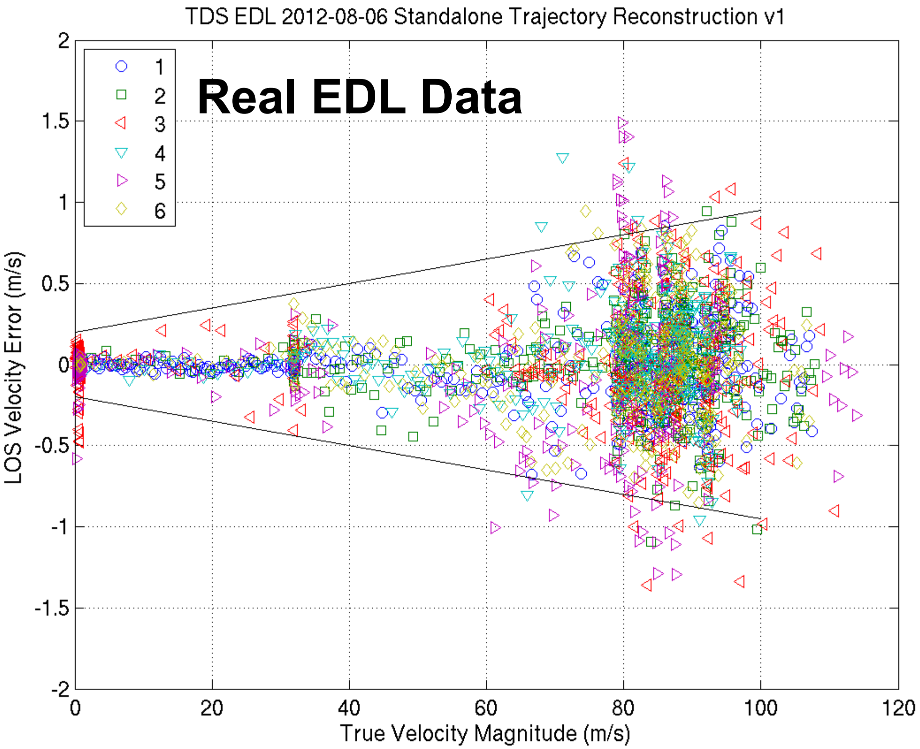
MSL EDL Range Error Comparison

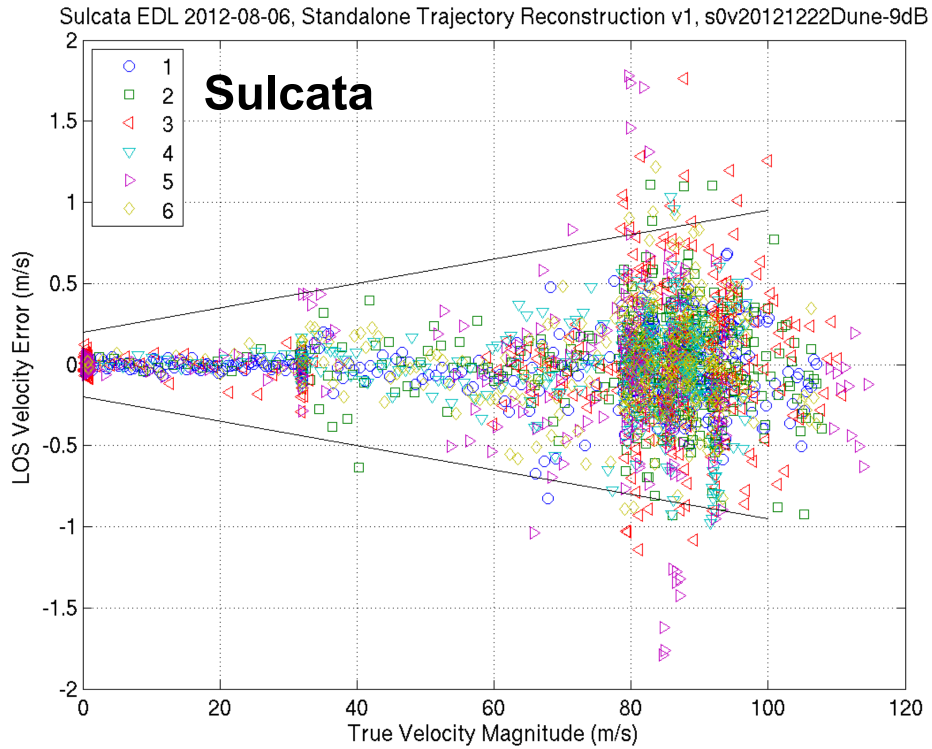


Black lines represent TDS 99% requirements

Good general agreement between actual radar data and simulation; outlier behavior slightly different (not unexpected)

MSL EDL Velocity Error Comparison





Black lines represent TDS 99% requirements

Good general agreement between actual radar data and simulation; outlier behavior slightly different (not unexpected)

MSL EDL Range Error Histograms

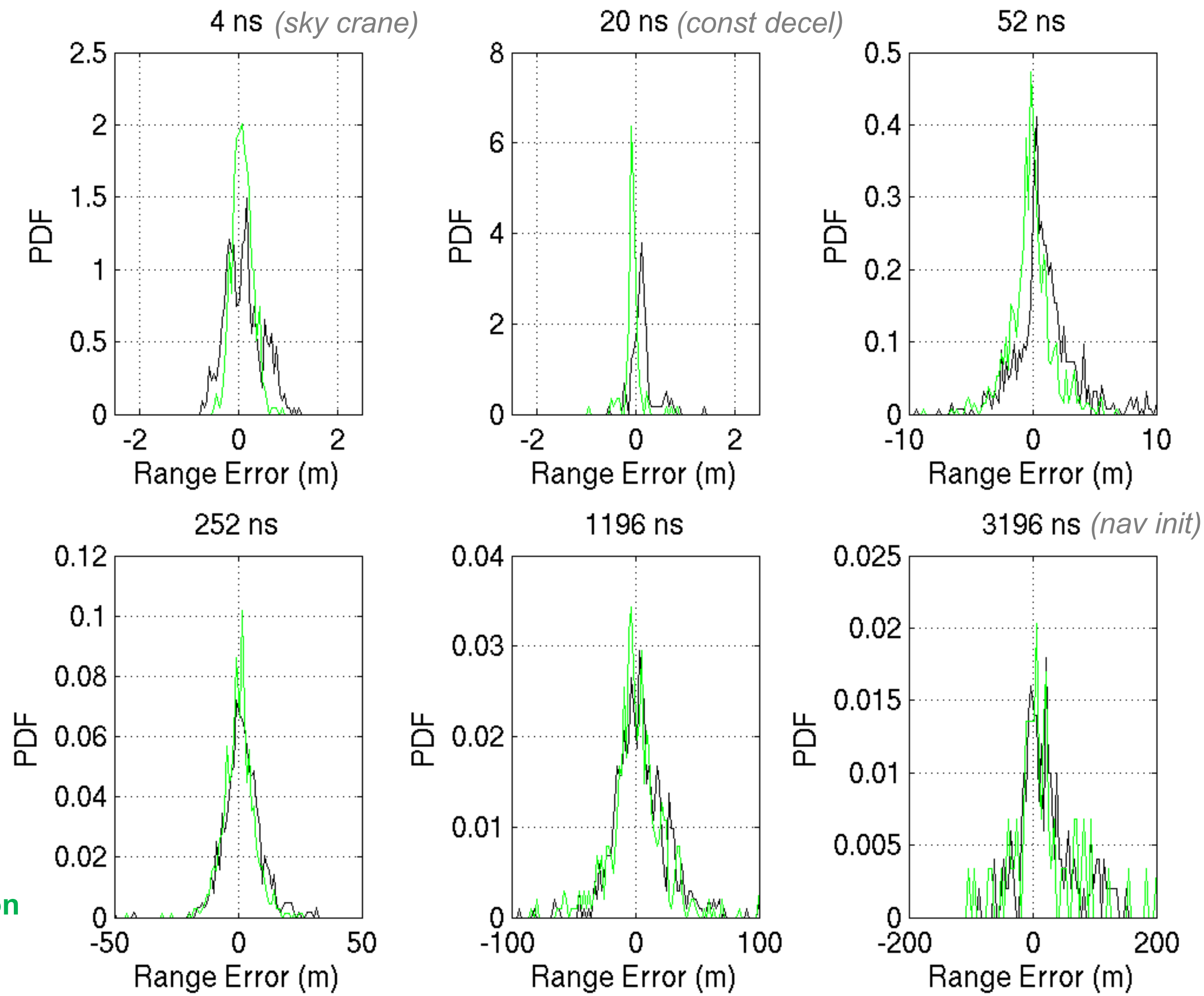
Histograms show all valid data from actual MSL landing

Data are sorted by radar transmit pulsewidth (as proxy for different regimes of operational envelope)

Data from real FM unit and Sulcata agree reasonably well

DEM errors are counted against TDS

Black = Real Data
Green = Simulation

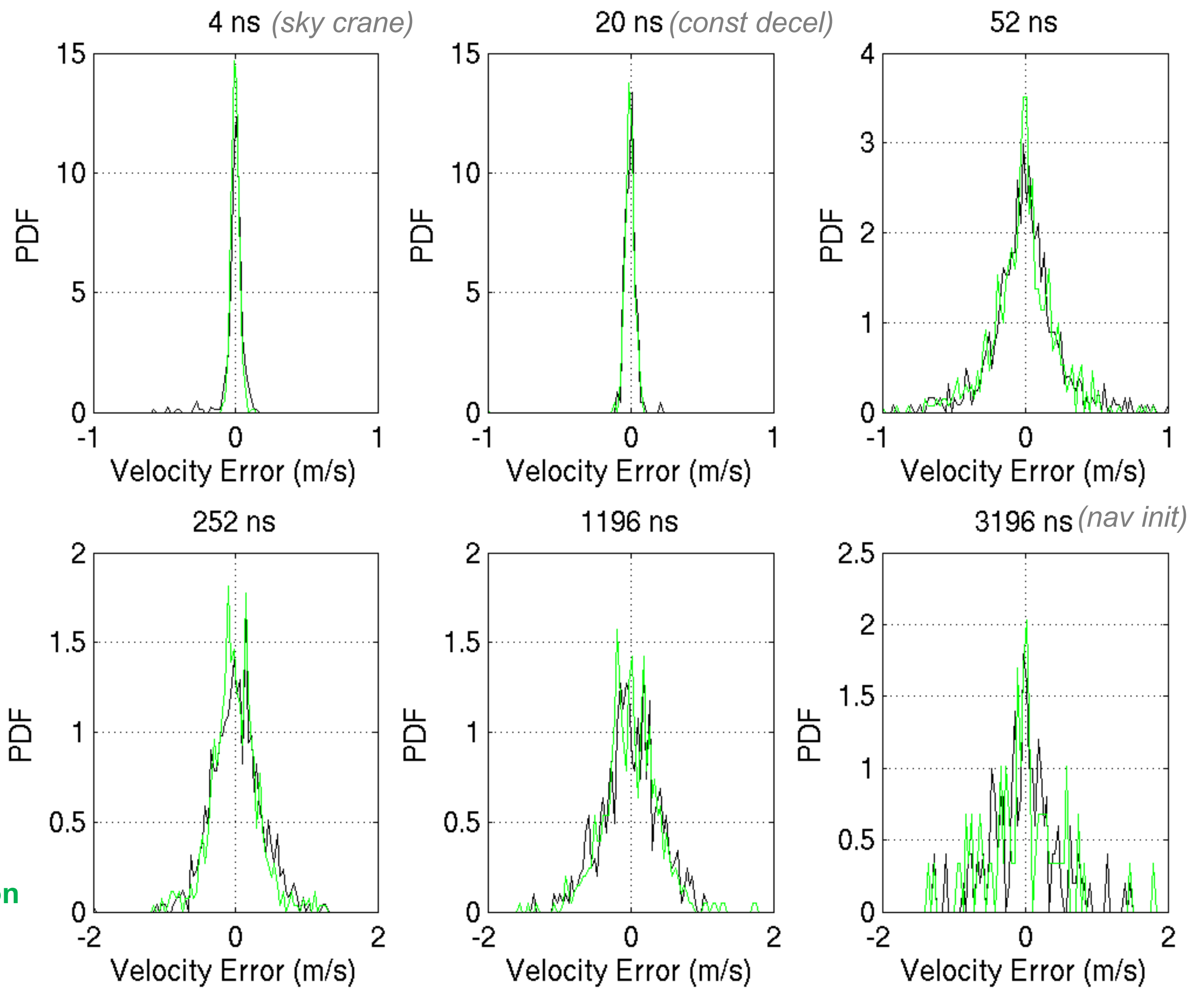


MSL EDL Velocity Error Histograms

Histograms show all valid data from actual MSL landing

Data are sorted by radar transmit pulsewidth (as proxy for different regimes of operational envelope)

Data from real FM unit and Sulcata agree very closely



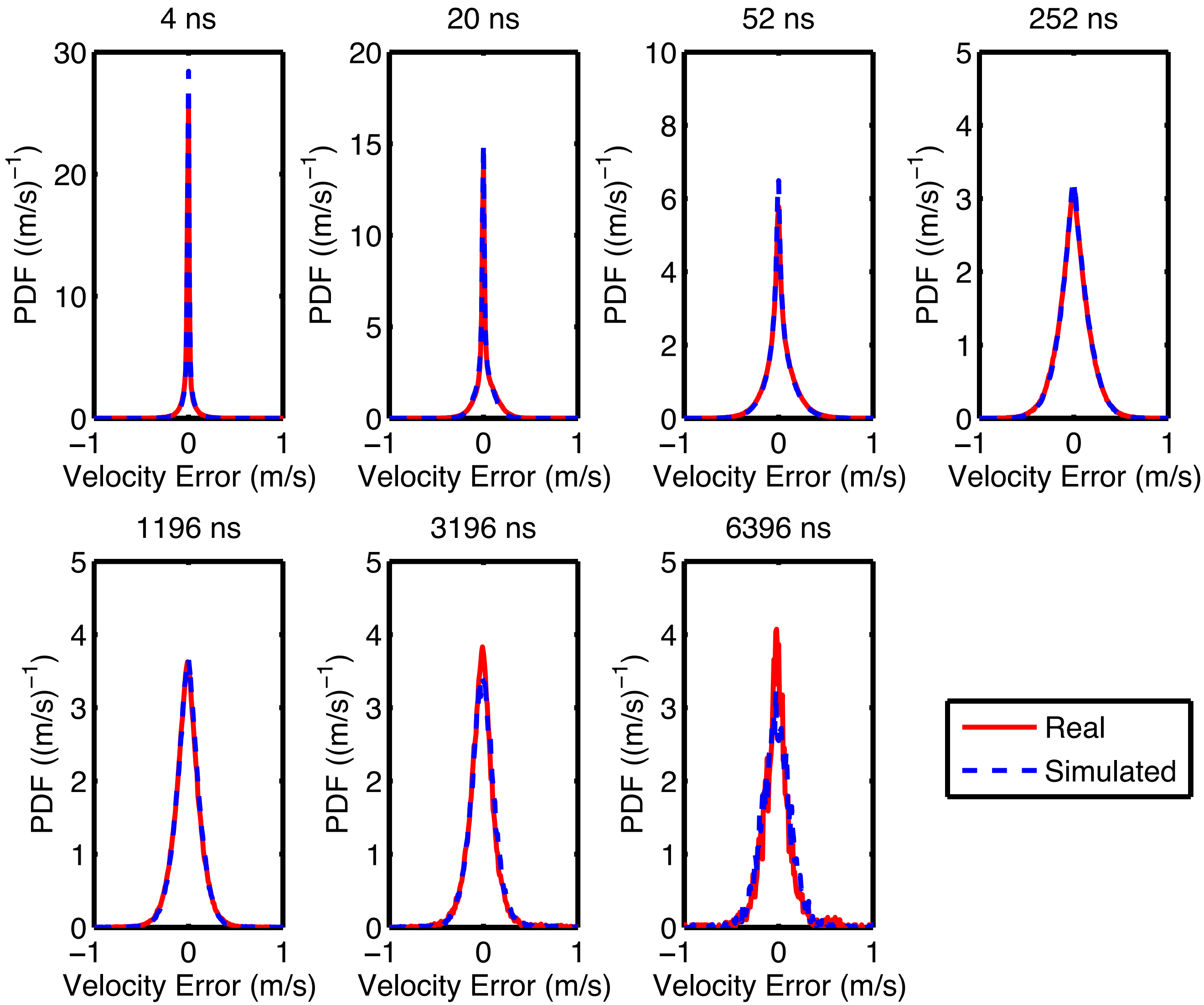
Black = Real Data
Green = Simulation

Field Test Velocity Error Histograms

Histograms show valid data within operational envelope from all MSL helicopter and F/A-18 field tests (~34 hrs)

Data are sorted by radar transmit pulsewidth (as proxy for different regimes of operational envelope)

Data from real TDS EM unit and Sulcata agree very closely



What's Next?

- M2020 landing
- Surface model is reasonably general and is applicable to different types of radars, but we need to see what next landing radar looks like...

Acknowledgements

- Thanks to Phoenix and MSL folks who worked on radar, field test, GN&C, and EDL; specifically,
  - Erik S. Bailey
  - J. Eric Belz
  - Eugene P. Bonfiglio
  - D. J. Byrne
  - Alexandra H. Chau
  - Benjamin D. Cichy
  - Yonggyu Gim
  - Michael P. Johnson
  - Cathleen E. Jones
  - Brian D. Pollard
  - Ernesto Rodriguez
  - A. Miguel San Martin
  - Scott J. Shaffer
  - E. David Skulsky
  - David W. Way

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA.

Selected References

1. Chen, C. W., "Approach for Modeling Coherent Radar Ground Echoes in Planetary Landing Simulations," *Journal of Spacecraft and Rockets*, Vol. 55, No. 1, 2018, pp. 85–94. DOI 10.2514/1.A33853
2. Chen, C. W. and Pollard, B. D., "Radar Terminal Descent Sensor Performance During Mars Science Laboratory Landing," *Journal of Spacecraft and Rockets*, Vol. 51, No. 4, 2014, pp. 1208–1216. DOI:10.2514/1.A32641
3. Pollard, B. D., and Chen, C. W., "A Radar Terminal Descent Sensor for the Mars Science Laboratory Mission," *Proceedings of the 2009 IEEE Aerospace Conference*, IEEE Publ., Piscataway, NJ, 2009, pp. 1–10. DOI:10.1109/AERO.2009.4839463
4. Montgomery, J. F., Bodie, J. H., Brown, J. D., Chen, A., Chen, C. W., Essmiller, J. C., Fisher, C. D., Goldberg, H. R., Lee, S. W., and Shaffer, S. J., "Implementing the Mars Science Laboratory Terminal Descent Sensor Field Test Campaign," *Proceedings of the 35th Annual AAS Rocky Mountain Guidance and Control Conference*, AAS Paper 2012- 091, 2012, pp. 647–664.
5. Goodman, J. W., *Statistical Optics*, Wiley-Interscience, New York, 1985, pp. 207–228, 347–356.