The Development and Implementation of the Kennedy Space Center Umbilical Clearance Tool

By David Chesnutt
Structures Analyst
Kennedy Space Center
Mail Code: NE-M6
Presentation Overview

• Engineering Analysis Branch Introduction
• Project Background
• Umbilical Clearance Tool Development
• Underlying Analysis Theory
• Analysis Tool Plots
Analysis Group Background

- KSC NE Engineering Analysis Branch
  - Disciplines
    - Structures
    - Dynamics
    - Fluids
    - Thermal
Background

- NASA developing heavy lift launch vehicle dedicated to increasing the space launch capabilities of the nation
  - Flight vehicle named Space Launch System (SLS)
Background

- Kennedy Space Center developing subsystems, known as umbilicals to provide interfaces and commodities to the Space Launch System (SLS) flight vehicle
- Umbilicals can provide:
  - Mechanical Support
  - Communications
  - Conditioned Air
  - Purges
  - Power
  - Fuel
Background

• The Core Stage Inner Tank Umbilical (CSITU) is a swing arm located at the top of the SLS first stage LH2 tank and provides…
  – Gaseous Hydrogen Vent Line
  – Conditioned Air
  – Gaseous Helium and Nitrogen
  – Electric Power/Control
Background

• During launch umbilical arms, upon separation, required to never re-contact vehicle
• KSC required to verify re-contact requirement
Background

• With supplied DAC3R flight data from Marshall Space Flight Center (MSFC), and in house umbilical specifications, KSC has data to perform analysis

• Result: KSC Umbilical Clearance Tool (UCT) undertaken
• Computationally intensive to compare 3D bodies moving as points of interest constantly change
Tool Development

• A number of umbilical arms pivot about a vertical axis
  – Umbilicals of this nature called swing arms
Tool Development

- Restricted motion of swings arms can be used to simplify clearance analysis to a plane of interest
6 Step Approach to Evaluate a Data Set:
- Extract relevant input from Marshal DAC3R raw data
- Generate the flight vehicle projected points with respect to time for each simulation of each data set
- Determine outlier simulations where flight vehicle is furthest from launch origin with respect to time
• 6 Step Approach to Evaluate a Data Set:
  – From outlier simulations generate points to define flight vehicle skin with respect to time
  – Envelope flight vehicle skin points with respect to time
  – Compare umbilical retract relative to envelope curves
Load Raw Data

• For each simulation file the function extracts:
  – Time of launch simulation
  – Core stage base point [relative to time]
  – Unit vector pointing along the core stage longitudinal axis [relative to time]
  – Solid Rocket Booster base points [relative to time]

• Extracted data is saved to a .mat file
Determine Projected Points

• Calculate the coordinates of the intersection between the plane of interest and the flight vehicle unit vector.
Determine Projected Points

• RH is calculated from:
  – Plane of interest elevation (ELV)
  – Base position (Bpos)
  – Flight vehicle unit vector (Uvect)

• RH used for:
  – Lookup function for vehicle skin diameter at plane of interest
  – Calculating location of $O_x$ & $O_y$

\[
RH = \frac{ELV - b_z}{u_z} \quad O_x = RH \cdot u_x + b_x \quad O_y = RH \cdot u_y + b_y
\]
Envelope Creation

- For each time step combine projected base point results
Envelope Creation

- Segment point data into angular regions
- Find \( n \# \text{ of } r_{\text{max}} \) points in descending order per region
Envelope Creation

• Capture sets that correspond to outer most points
  – These sets represent simulations with the greatest drift
Flight Vehicle Skin Analysis

- Steps to calculate ellipse curve:
  1. Determine space fixed rotations to orient Uvect in up (Z) direction
  2. Within the flight vehicle linear space create the plane of interest equation
  3. Create a cylinder equation oriented up using flight vehicle radius at plane of interest
• Steps to calculate ellipse curve:

4. Intersect the two equations to define vehicle skin linear space

5. Transform vehicle skin linear space to global coordinate system using space fixed rotations calculated in step 1

\[
\begin{bmatrix}
    s_x(r, \theta) \\
    s_y(r, \theta) \\
    s_z(r, \theta)
\end{bmatrix}^{abs} = [R_x][R_y]
\begin{bmatrix}
    s_x(r, \theta) \\
    s_y(r, \theta) \\
    s_z(r, \theta)
\end{bmatrix}^{veh}
\]
Lookup Parameters

- Radius value \( r \) in skin function determined from a look up function
  - Input: RH (Relative Height)
  - Output: \( r \) (vehicle skin radius)
Envelope Creation

- $\theta$ is left to the analyst to decide the number of points and their relative angle between them.
- Dilemma arises if same $\theta$ values used for each data set.
- Resolution:
  - Vary $\theta$ values using normalized simulation id as perturbation.

Vehicle skin points all initialize at some locations leaving empty envelope sectors.
Envelope Creation

• Vehicle skin points generated for each outer drift point selected from initial envelope of projected base points

Time Slice Data
Envelope Creation

- Envelope algorithm used on vehicle skin points
- Vehicle envelope defined by outermost points

Envelope Curve
Generate Swing Arm Skin

• In separate analysis umbilical retract dynamics evaluated
  – Simulation Output: Umbilical orientation with respect to time
• Umbilical envelope used with orientation to determine points of interest
Calculate Clearance

- In clearance simulation compare swing arm point of interest to vehicle skin envelop curves w.r.t time
- Find closest vehicle skin point to swing arm POI
  - Clearance outputs:
    - Coordinates of POI and closest vehicle skin point
    - Clearance associated to two points
    - Simulation IDs corresponding min clearance points
Clearance Plot Half Speed
East View Check

- Light/Fast
- ASEU Retract
- Heavy/Slow
BACKUP SLIDES
Flight Vehicle Skin Analysis

- MATLAB Function Name: scurve.m
- Steps to calculate ellipse curve:
  1. Determine space fixed rotations to orient Uvect in up (Z) direction

$$Uvect = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Find $[R_x]$ such that:

$$\begin{bmatrix} a' \\ 0 \\ c' \end{bmatrix} = [R_x] \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Find $[R_y]$ such that:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = [R_y] \begin{bmatrix} a' \\ 0 \\ c' \end{bmatrix}$$

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & \sin(\theta_x) \\ 0 & -\sin(\theta_x) & \cos(\theta_x) \end{bmatrix}, \quad [R_y] = \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix}$$
Flight Vehicle Skin Analysis

• Steps to calculate ellipse curve:
  2. Within the flight vehicle linear space create the plane of interest equation

Find plane of interest normal within flight vehicle linear space

\[
\begin{bmatrix}
  n_x^{\text{veh}} \\
  n_y \\
  n_z
\end{bmatrix}^{\text{abs}} = \begin{bmatrix}
  [R_x][R_y]
\end{bmatrix}^T \begin{bmatrix}
  0 \\
  0 \\
  1
\end{bmatrix} = \begin{bmatrix}
  c_x & sxsy & cxsy \\
  0 & c_x & -sx \\
  -sy & sxcy & cxcy
\end{bmatrix}^{\text{abs}}
\]

Create plane equation in flight vehicle linear space

\[ n_x x + n_y y + n_z z = 0 \]

or

\[ (cxsy)x + (-sx)y + (cxy)z = 0 \]

Note: \( sx, sy = \sin(\theta_x), \sin(\theta_y) \), and \( cx, cy = \cos(\theta_x), \cos(\theta_y) \)
Flight Vehicle Skin Analysis

• Steps to calculate ellipse curve:

3. Create a cylinder equation oriented up using vehicle radius at plane of interest

Define a subspace for a cylinder representing the rocket

\[
\begin{bmatrix}
    r \cdot \cos(\theta) \\
    r \cdot \sin(\theta) \\
    z
\end{bmatrix}^{veh}
\]

for \( r = \) rocket skin radius at plane of interest (rh_lookup.m)
Flight Vehicle Skin Analysis

- Steps to calculate ellipse curve:
  4. Intersect the two equations to define vehicle skin linear space

\[
\begin{bmatrix}
x(r, \theta)
y(r, \theta)
z(r, \theta)
\end{bmatrix}^{veh} =
\begin{bmatrix}
r \cos(\theta) \\
r \sin(\theta) \\
cx \cdot sy \cdot r \cdot \cos(\theta) - sx \cdot r \cdot \sin(\theta) \\
-cx \cdot cy
\end{bmatrix}^{veh}
\]
Flight Vehicle Skin Analysis

- Steps to calculate ellipse curve:
  5. Transform vehicle skin linear space to global coordinate system using space fixed rotations calculated in step 1

\[
\begin{bmatrix}
    s_x(r, \theta) \\
    s_y(r, \theta) \\
    s_z(r, \theta)
\end{bmatrix}_{\text{abs}}^\text{veh} = [R_x][R_y]\begin{bmatrix}
    s_x(r, \theta) \\
    s_y(r, \theta) \\
    s_z(r, \theta)
\end{bmatrix}_{\text{veh}}^\text{veh}
\]
Marshall Data Synthesis

• The supplied structures data from the August 2015 technical interchange meeting is in the form of a set of points moving relative to their nominal position with respect to time.

• Each point represents a flex body grid point location in the SLS X, Y, and Z coordinate frame.

• Task:
  – Determine vehicle position and orientation with respect to time from point data
Marshall Data Synthesis

- Pick 3 points to represent flight vehicle in the nominal position \((P_1, P_2, \text{ and } P_3)\)
- Average \(P_1, P_2, \text{ and } P_3\) point coordinates to find \(P_{\text{avg}}\)
- Create unit vectors \(u_1\) and \(u_2\) from \(P_{\text{avg}}\) to \(P_1\) and \(P_2\)
Marshall Data Synthesis

- Normalize the cross product of $u_1$ and $u_2$ to determine $v_3$
- Take the cross product of $u_1$ and $v_3$ to determine $v_4$
- Assemble $u_1$, $v_3$, and $v_4$ to form $T_A$
Marshall Data Synthesis

- $A_T$ is the matrix which will transform a coordinate system in the default coordinate frame orientation to $CSY_v$

$$A_T = \begin{bmatrix} u_1 & u_x1 & u_y1 & u_z1 \\ v_4 & v_x4 & v_y4 & v_z4 \\ v_3 & v_x3 & v_y3 & v_z3 \end{bmatrix}$$

\begin{align*}
v_x4 &= \frac{(ux2*uy1^2 - ux1*uy2*uy1 + ux2*uz1^2 - ux1*uz2*uz1)}{(abs(ux1*uy2 - ux2*uy1)^2 + abs(ux1*uz2 - ux2*uz1)^2 + abs(uy1*uz2 - uy2*uz1)^2)^{1/2}} \\
v_y4 &= \frac{(uy2*ux1^2 - ux2*uy1*ux1 + uy2*uz1^2 - uy1*uz2*uz1)}{(abs(ux1*uy2 - ux2*uy1)^2 + abs(ux1*uz2 - ux2*uz1)^2 + abs(uy1*uz2 - uy2*uz1)^2)^{1/2}} \\
v_z4 &= \frac{(uz2*ux1^2 - ux2*uz1*ux1 + uz2*uy1^2 - uy2*uz1*uy1)}{(abs(ux1*uy2 - ux2*uy1)^2 + abs(ux1*uz2 - ux2*uz1)^2 + abs(uy1*uz2 - uy2*uz1)^2)^{1/2}} \\
v_x3 &= \frac{(uy1*uz2 - uy2*uz1)}{(abs(ux1*uy2 - ux2*uy1)^2 + abs(ux1*uz2 - ux2*uz1)^2 + abs(uy1*uz2 - uy2*uz1)^2)^{1/2}} \\
v_y3 &= -\frac{(ux1*uz2 - ux2*uz1)}{(abs(ux1*uy2 - ux2*uy1)^2 + abs(ux1*uz2 - ux2*uz1)^2 + abs(uy1*uz2 - uy2*uz1)^2)^{1/2}} \\
v_z3 &= \frac{(ux1*uy2 - ux2*uy1)}{(abs(ux1*uy2 - ux2*uy1)^2 + abs(ux1*uz2 - ux2*uz1)^2 + abs(uy1*uz2 - uy2*uz1)^2)^{1/2}}
\end{align*}
Repeat process for the 3 corresponding flight vehicle points in the deflected position \( P_{1}' \), \( P_{2}' \), and \( P_{3}' \) to create \( P_{\text{avg}{'}} \), \( u_{1{'}} \), \( u_{2{'}} \), \( v_{3{'}} \), and \( v_{4{'}} \).
$B_T$ is the matrix which will transform a coordinate system in the default coordinate frame orientation to the deflected vehicle frame $CSY_v'$. 

$$B_T = \begin{bmatrix} u_1' \\ v_4' \\ v_3' \\ \end{bmatrix} = \begin{bmatrix} u_{x1}' & u_{y1}' & u_{z1}' \\ v_{x4}' & v_{y4}' & v_{z4}' \\ v_{x3}' & v_{y3}' & v_{z3}' \end{bmatrix}$$

\[
\begin{align*}
v_{x4} &= \frac{u_{y1}u_{z2} - u_{y2}u_{z1}}{(u_{x1}u_{y2} - u_{x2}u_{y1})^2 + (u_{x1}u_{z2} - u_{x2}u_{z1})^2 + (u_{y1}u_{z2} - u_{y2}u_{z1})^2}^{1/2} \\
v_{y4} &= \frac{-u_{x1}u_{z2} + u_{x2}u_{z1}}{(u_{x1}u_{y2} - u_{x2}u_{y1})^2 + (u_{x1}u_{z2} - u_{x2}u_{z1})^2 + (u_{y1}u_{z2} - u_{y2}u_{z1})^2}^{1/2} \\
v_{z4} &= \frac{u_{x1}u_{y2} - u_{x2}u_{y1}}{(u_{x1}u_{y2} - u_{x2}u_{y1})^2 + (u_{x1}u_{z2} - u_{x2}u_{z1})^2 + (u_{y1}u_{z2} - u_{y2}u_{z1})^2}^{1/2}
\end{align*}
\]
Marshall Data Synthesis

• To determine the transformation matrix that will orient the vehicle from the nominal position to the deflected position we use the equation:

\[
BA = BTAT = BTAT^T
\]

\[
BA = \begin{bmatrix}
  r_{x1} & r_{y1} & r_{z1} \\
  r_{x2} & r_{y2} & r_{z2} \\
  r_{x3} & r_{y3} & r_{z3}
\end{bmatrix} = \begin{bmatrix}
  u_{x1}' & u_{y1}' & u_{z1}' \\
  v_{x4}' & v_{y4}' & v_{z4}' \\
  v_{x3}' & v_{y3}' & v_{z3}'
\end{bmatrix} \begin{bmatrix}
  u_{x1} & u_{x4} & u_{x3} \\
  u_{y1} & u_{y4} & u_{y3} \\
  u_{z1} & u_{z4} & u_{z3}
\end{bmatrix}
\]
Marshall Data Synthesis

• Equate T to Rxyz
• Solve for \( \theta_y \), \( \theta_x \), and \( \theta_z \)

\[
R_{xyz} = [R_z][R_y][R_x] = \begin{bmatrix}
cz & sz & 0 \\
-sz & cz & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
cy & 0 & -sy \\
0 & 1 & 0 \\
0 & cx & sx
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & -sx & cx
\end{bmatrix}
\]

\[
R_{xyz} = \begin{bmatrix}
cy * cz & cx * sz + cz * sx * sy & sx * sz - cx * cz * sy \\
-cy * sz & cx * cz - sx * sy * sz & cz * sx + cx * sy * sz \\
sy & -cy * sx & cx * cy
\end{bmatrix}
= \begin{bmatrix}
r_{x1} & r_{y1} & r_{z1} \\
r_{x2} & r_{y2} & r_{z2} \\
r_{x3} & r_{y3} & r_{z3}
\end{bmatrix} = R_AT
\]

\[
\theta_y = \pm \text{atan2} \left( r_{x3}, \sqrt{1 - r_{x3}^2} \right) \quad \theta_x = \text{atan2} \left( \frac{r_{y3}}{-cy'}, \frac{r_{z3}}{cy'} \right) \quad \theta_z = \text{atan2} \left( \frac{r_{x2}}{-cy'}, \frac{r_{x1}}{cy'} \right)
\]

\text{Note: } sx, sy, sz = \sin(\theta_x), \sin(\theta_y), \sin(\theta_z) \text{ and } cx, cy, cz = \cos(\theta_x), \cos(\theta_y), \cos(\theta_z)