42

A General-Purpose Multi-body, Multi-spacecraft Simulation

Eric Stoneking

2014

What kind of a name is 42?

- 42 is a tongue-in-cheek reference to the “answer to Life, the Universe, and Everything”, as proclaimed by Douglas Adams in his Hitchhiker's Guide to the Galaxy stories
  - The first book is spent seeking this answer, which had taken a special computer and millions of years to compute. But by the time they had it, nobody knew what the question was anymore.
  - The next two books are spent seeking the question, which turns out to be: “What do you get when you multiply six by nine?”
- The reader may draw a number of lessons from this story...

So what is it really?

- A simulation of spacecraft attitude and orbital dynamics and control
- Intended for use from concept studies through ops
  - Rapid prototyping makes it useful for MDL studies
  - Environment models support actuator sizing, performance studies
  - High-fidelity dynamics handle multi-body, flexible-body spacecraft
  - Portability (Mac, linux, Windows) minimizes infrastructure requirements
  - Clean interface aids progression from flight software “model” to dropping in actual flight software
  - Visualization aids situational awareness from concept to operations
- Designed to be powerful, but easy to get started

Features

- Multiple spacecraft, anywhere in the solar system
  - Two-body, three-body orbit dynamics (with seamless transition between)
  - One sun, nine planets, 45 major moons
  - Minor bodies (comets and asteroids) added as needed
    - RQ36, Eros, Itokawa, Wirtanen, etc
- Multi-body spacecraft
  - Tree topology
  - Kane's dynamics formulation
  - Each body may be rigid or flexible
  - Flexible parameters taken (by m-file script) from Nastran output (.f06 file)
  - Joints may have any combination of rotational and translational degrees of freedom

More Features

- Supports precision formation flying
  - Several S/C may be tied to a common reference orbit
    - Used by FACET, Stellar Imager studies
  - Encke's method or Euler-Hill equations used to propagate relative orbit states
    - Precision maintained by judicious partitioning of dynamics
      - Add big things to big things, small things to small things
- Clean FSW interface facilitates FSW validation
  - Used by GLAST project for independent validation of vendor's (autocoded) GNC flight software
- Open Source, available on sourceforge and github
  - Sourceforge.net/projects/fortytwospacecraftsimulation
  - Github/ericstoneking/42

42 Software Overview

Architecture, Interfaces with Matlab

42 is Written in C for Speed, Portability

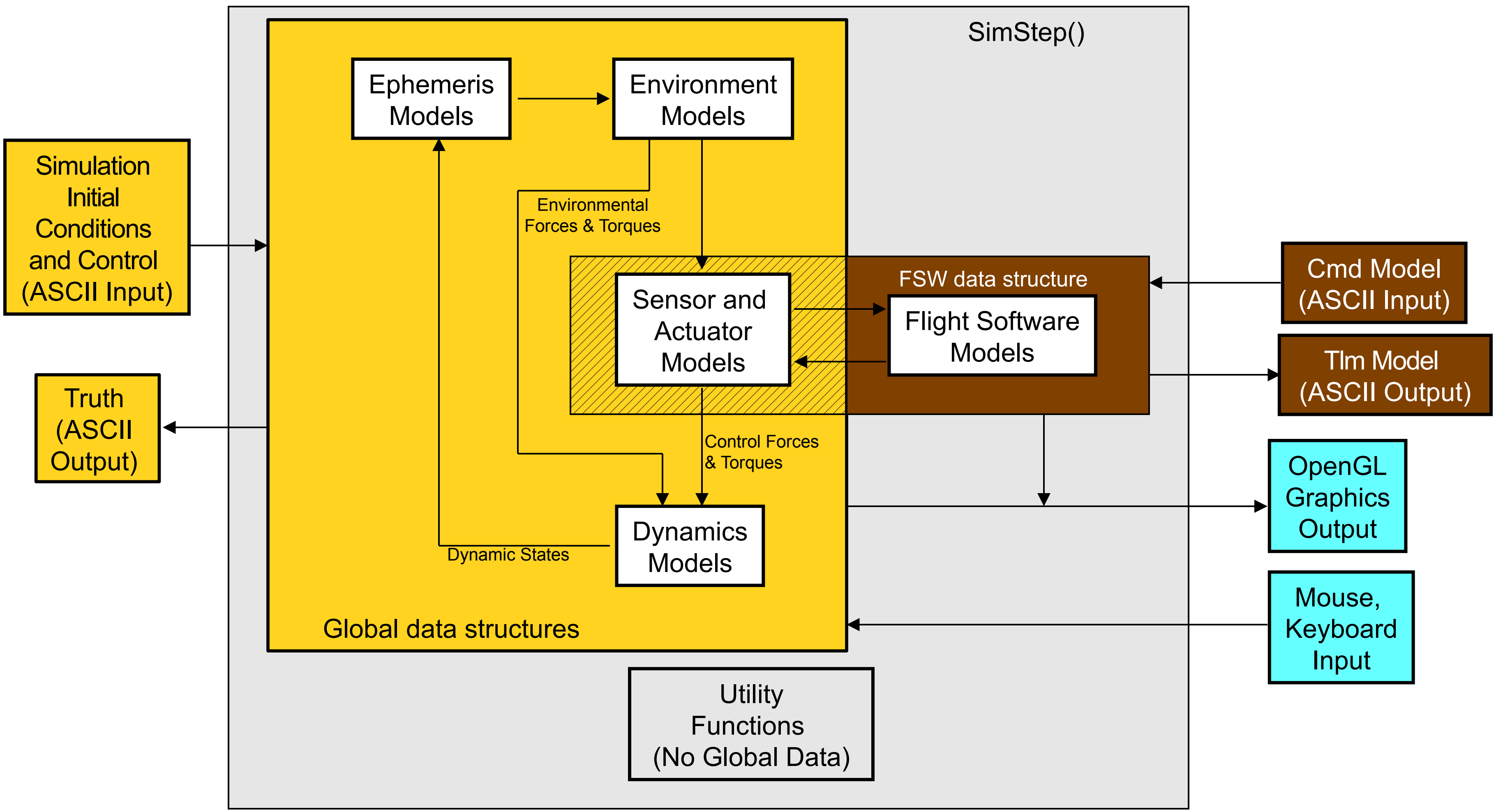
- The C language has a well-deserved reputation for giving you enough rope to hang yourself. Like any sharp tool, some care in handling is needed.
- If it weren't fast, powerful, portable, ubiquitous, and free, we wouldn't put up with it.
- The good news is, you can get a quick prototype running in 42 with zero C coding
  - Use “prototype FSW”, command script
- The next steps are to add your own sensor, FSW, and actuator models
  - Simple existing models show where to insert yours
  - Global data structures make needed signals easy to find
  - Choose your level of fidelity and difficulty: n00b -> l33t
- You can get into the guts of the environment and dynamics if you want to, but you probably won't ever need to

Command Script

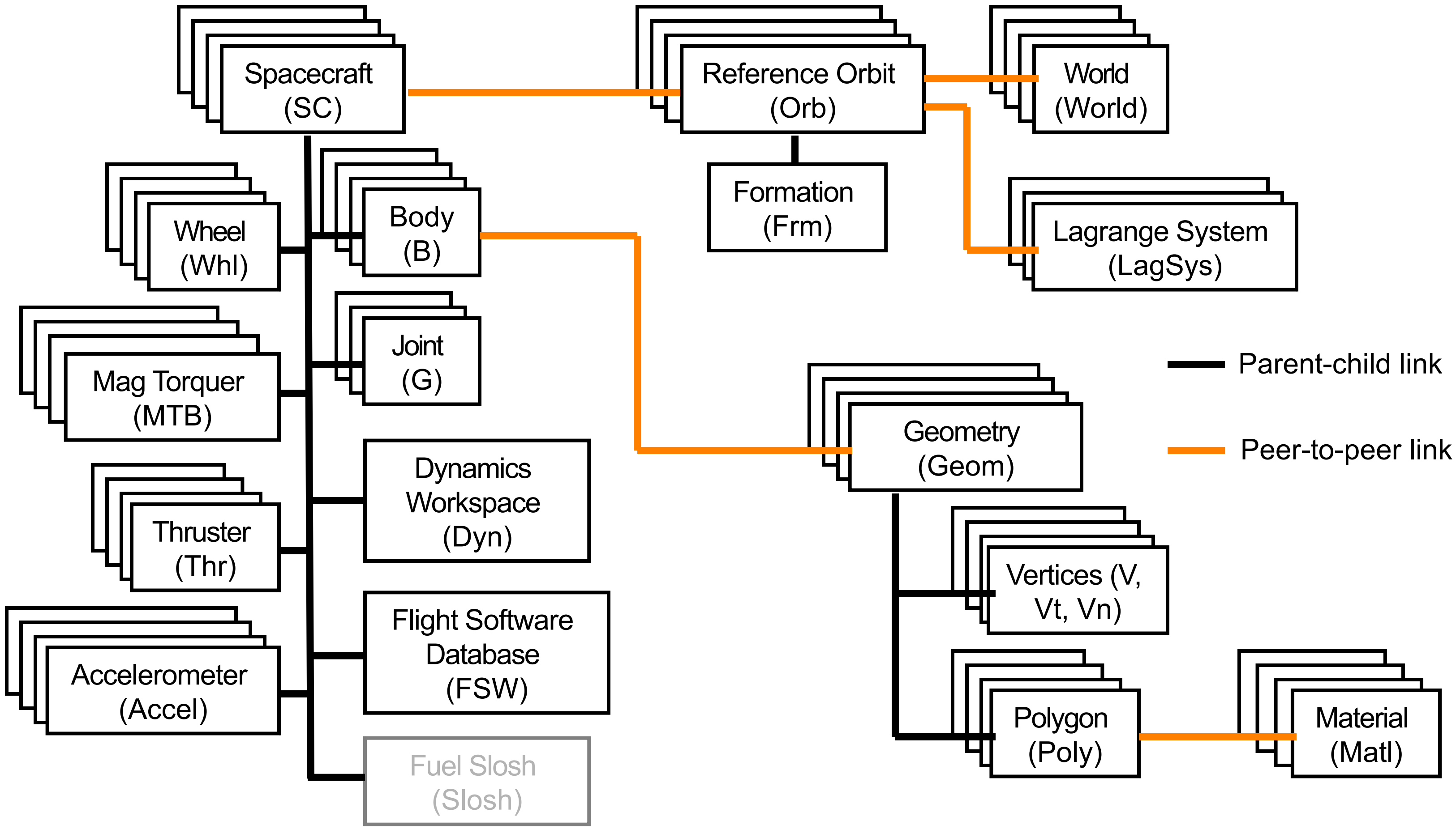
- The command script allows changing configuration (sim, FSW, or GUI) in the course of a sim run, from an ASCII input file.
- Good for rapid prototyping
- Inject sensor or actuator failures
- Model ground commands
- Script graphics for movie generation

```
<<<<<<<<<<<<<<<<<  42:  Command Script File  >>>>>>>>>>>>>>>>>
0.0 SC[0] qrl = [0.0 0.0 0.0 1.0]
0.0 Point SC[1].B[0] Primary Vector [0.0 0.0 -1.0] at SUN
0.0 Align SC[1].B[0] Secondary Vector [0.0 1.0 0.0] with L-frame Vector
[0.0 1.0 0.0]
0.0 Point SC[2].B[0] Primary Vector [1.0 0.0 0.0] at SC[1]
0.0 Align SC[2].B[0] Secondary Vector [0.0 1.0 0.0] with L-frame Vector
[0.0 1.0 0.0]
0.0 Point SC[2].B[1] Primary Vector [0.0 0.0 1.0] at SUN
0.0 Point SC[2].B[2] Primary Vector [0.0 0.0 1.0] at SUN
0.0 SC[3] Cmd Angles = [30.0 45.0 60.0] Seq = 123 wrt L Frame
EOF
```

42's Architecture



Global Data Structure Relationships



The Three Layers of Code

- 42 uses three “layers” of code:
  - Low-level toolkit functions:
    - Everything comes in and goes out through arguments and return value
    - No global variables
    - Segregated into *kit.c files (mathkit.c, dcmkit.c, etc)
  - Top-level frame outline:
    - Follows outline, does minimal math/logic itself
    - Access to all global variables
  - Mid-level workhorse functions
    - Most likely to be tailored to a particular need
    - Access to global variables

Toolkit Code Example

```
/**********************************************************************/
/*  Convert quaternion to direction cosine matrix                    */

void Q2C(double Q[4], double C[3][3])
{
      C[0][0] = 1.0-2.0*(Q[1]*Q[1]+Q[2]*Q[2]);
      C[0][1] = 2.0*(Q[0]*Q[1]+Q[2]*Q[3]);
      C[0][2] = 2.0*(Q[2]*Q[0]-Q[1]*Q[3]);
      C[1][0] = 2.0*(Q[0]*Q[1]-Q[2]*Q[3]);
      C[1][1] = 1.0-2.0*(Q[2]*Q[2]+Q[0]*Q[0]);
      C[1][2] = 2.0*(Q[1]*Q[2]+Q[0]*Q[3]);
      C[2][0] = 2.0*(Q[2]*Q[0]+Q[1]*Q[3]);
      C[2][1] = 2.0*(Q[1]*Q[2]-Q[0]*Q[3]);
      C[2][2] = 1.0-2.0*(Q[0]*Q[0]+Q[1]*Q[1]);
}
```

Top-level Code Example

```
ReportProgress();
ManageFlags();

/* Read and Interpret Command Script File */
CmdInterpreter();

/* Update Dynamics to next Timestep */
for(Isc=0;Isc<Nsc;Isc++) {
   if (SC[Isc].Exists) Dynamics(&SC[Isc]);
}
OrbitMotion();
SimComplete = AdvanceTime();

/* Update SC Bounding Boxes occasionally */
ManageBoundingBoxes();

Ephemerides(); /* Sun, Moon, Planets, Spacecraft, Useful Auxiliary Frames */
for(Isc=0;Isc<Nsc;Isc++) {
   S = &SC[Isc];
   if (S->Exists) {
      Environment(S);    /* Magnetic Field, Atmospheric Density */
      Perturbations(S);  /* Environmental Forces and Torques */
      Sensors(S);
      FlightSoftWare(S);
      Actuators(S);
      PartitionForces(S); /* Orbit-affecting and "internal" */
   }
}
Report();  /* File Output */
```

Mid-level Code Example

```
/* Find Attitude Command */
FindCLN(FSW->PosN,FSW->VelN,CRN,wln);
C2Q(CRN,qrn);
MxV(CRN,FSW->svn,svr);

/* Form Error Signals */
QxQT(FSW->qbn,qrn,qbr);
RECTIFYQ(qbr);

/* PD Control */
for(i=0;i<3;i++) {
   FSW->Tcmd[i] = -FSW->Kr[i]*FSW->wbn[i]-FSW->Kp[i]*(2.0*qbr[i]);
   FSW->Twhlcmd[i] = -FSW->Tcmd[i];
}

/* Momentum Management */
for(i=0;i<3;i++) {
   Herr[i]=FSW->Hw[i]-FSW->Hwcmd[i];
}
VxV(Herr,FSW->bvb,HxB);
for(i=0;i<3;i++) FSW->Mmtbcmd[i] = FSW->Kunl*HxB[i];
```

Interfaces to Matlab

- 42 generates ASCII output files, which may be loaded into Matlab (or whatever) for post-processing and plotting
- Using Matlab's `mcc` utility, m-files may be translated into C, and compiled and linked into 42
  - As done by FACET

Matlab + 42 = Monte Carlo

- 42 can be called from within Matlab using the `system` command
- Use Matlab as the MC executive
  - Generate initial conditions, parameters
  - Write to 42's input files
  - Run 42
  - Process and save data
  - Repeat
- Use 42 as the high-speed, high-fidelity component

Matlab/42 Example

```
for Irun=1:Nrun,

   % Compute initial attitude
   CRN = TRIAD(tvn(Irun,:),svn,[0 0 1],[1 0 0]);
   qrn = C2Q(CRN);

   % Write target to file
   Outdata = [TrgRA(Irun) TrgDec(Irun)];
   save -ascii ./MOMBIAS/TargetRaDec.inp Outdata

   % Write initial attitude to file
   line = sprintf('%f %f %f %f ! Quaternion\n', qrn(1),qrn(2),qrn(3),qrn(4));
   OverwriteLineInFile('./MOMBIAS/GLAST.inp',21,line);

   % Run 42 for three days.
   system('./42 MOMBIAS');

   % Record pointing histogram.
   load ./MOMBIAS/AngleToGo.42
   [HistCount(Irun,:),HistAng(Irun,:)] = hist(AngleToGo,20);

end
```

42 Walkthrough
Models, Terminology, Input Files

Environment Models

- Planetary Ephemerides
  - From Meeus, “Astronomical Algorithms”
  - Good enough for GNC validation, not intended for mission planning
    - Use GMAT or ODTBX for that
- Gravity Models have coefficients up to 18th order and degree
  - Earth: EGM96
  - Mars: GMM-2B
  - Luna: GLGM2
- Planetary Magnetic Field Models
  - IGRF up to 10th order (Earth only)
  - Tilted offset dipole field
- Earth Atmospheric Density Models
  - MSIS-86 (thanks to John Downing)
  - Jacchia-Roberts Atmospheric Density Model (NASA SP-8021)
- Simple exponential Mars atmosphere density model
  - New models easily incorporated as the state of the art advances

Dynamics Models

- Full nonlinear "6DOF" (actually N-DOF) dynamics
- Attitude Dynamics
  - One or many bodies
    - Tree topology (no kinematic loops)
  - Each body may be rigid or flexible
  - Joints may combine rotational and translational DOFs
    - May be gimballed or spherical
  - Slosh may be modeled as a pendulum (lo-fi, quick to implement and run)
    - 42 may run concurrently with Star-CCM CFD software for hi-fi slosh
  - Wheels embedded in Body[0]
  - Torques from actuators, aerodynamic drag, gravity-gradient, solar radiation pressure, joint torques
- Orbit Dynamics
  - Two- or three-body orbits
  - Encke or Euler-Hill (Clohessy-Wiltshire) for relative orbit motion (good for formation flying, prox ops)
  - Forces from actuators, aerodynamic drag, non-spherical gravity, third-body gravity, solar radiation pressure

Reference Frames are Important!

- In any dynamics problem beyond the spinning top, a systematic approach to reference frames and the relationships between them is vital
- For 42, we define several fundamental reference frames, and notational conventions to keep quaternions and direction cosines sorted out

Reference Frames (1 of 2)

- Heliocentric Ecliptic (H)
  - Planet positions expressed in this frame
- Each world has an inertial (N) and rotating (W) frame
  - For Earth, N = ECI (True of date), W = ECEF
  - N is the bedrock for orbits, S/C attitude dynamics
  - Full Disclosure: Although True-of-Date <-> J2000 conversions are provided, the distinction is not always rigorously made
    - Star vectors provided in J2000 (from Skymap), converted to H
    - Planet ephemerides are assumed given in true-of-date H
    - Transformation from N to W is simple rotation, implying N is True-of-Date
    - TOD ↔ J2000 conversions in envkit.c

Reference Frames (2 of 2)

- Each reference orbit has a reference point R
  - For two-body orbit, R moves on Keplerian orbit
  - For three-body orbit, R propagates under influence of both attracting centers (as point masses)
  - S/C orbit perturbations integrated with respect to R
- Associated with each R is a LVLH frame (L) and a formation frame (F)
  - F is useful for formation-flying scenarios
  - F may be offset from R, may be fixed in N or L
- Each spacecraft has one or more Body (B) frames and one LVLH frame (L)
  - L(3) points to nadir, L(2) points to negative orbit normal
  - SC.L is distinct from Orb.L, since SC may be offset from R

Notation for Quaternions, DCMs

- The rotation from frame *A* to frame *B* may be described by the direction cosine matrix

$$ {}^{B}C^{A}_{ij} = \hat{b}_i \cdot \hat{a}_j $$

- Given the components of a vector in *A*, its components in *B* may be found by the multiplication

$$ {}^{B}v = {}^{B}C^{A}\,{}^{A}v $$

- In C, we write the DCM as CBA to preserve order of superscripts, eg

```
MxV(CBA,va,vb)
```

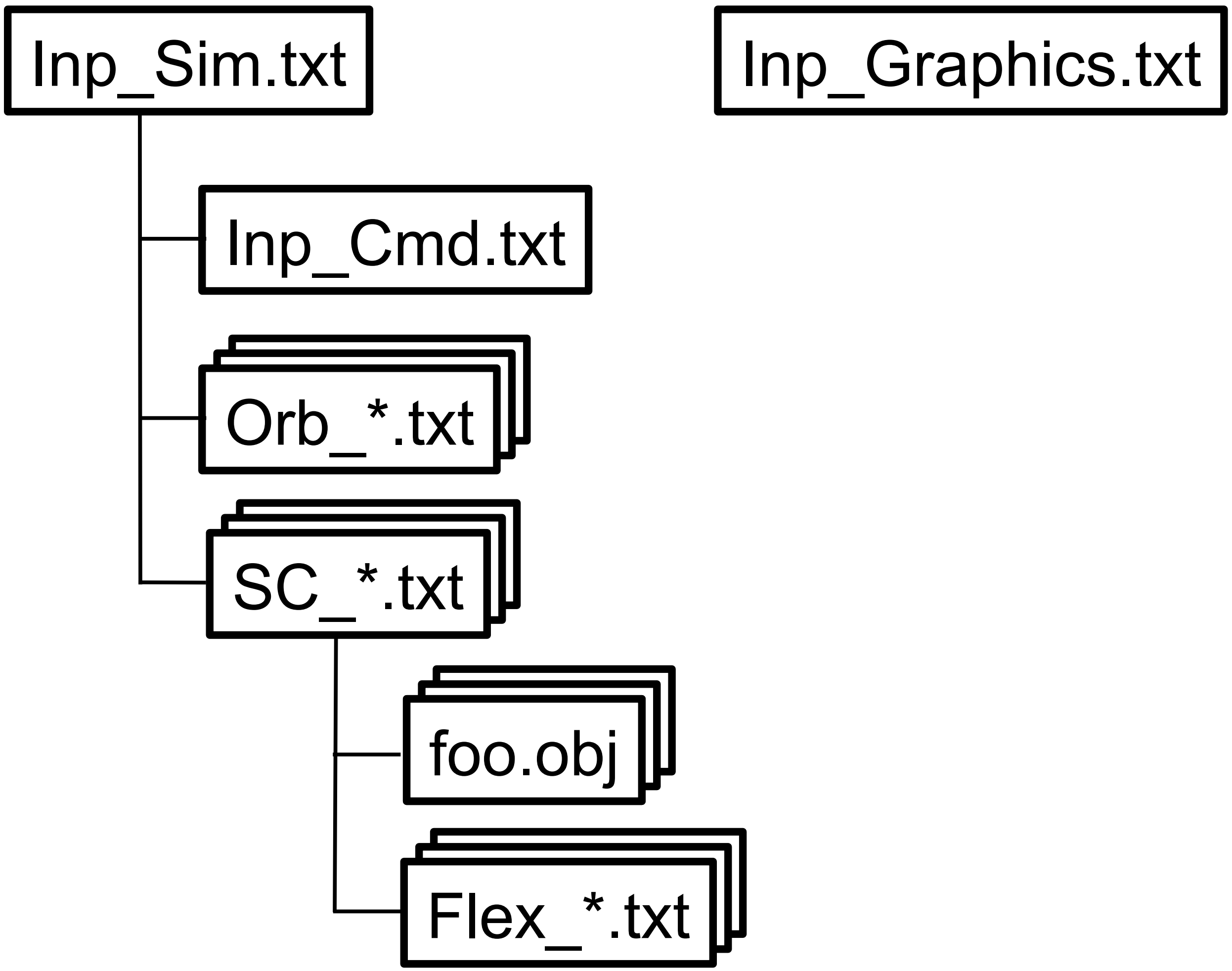
- Quaternions are another way to describe rotations. We use a parallel notation:

```
QxV(qba,va,vb)
```

- These and similar conventions promote concise, *unambiguous* code

Input File Overview

Input File Hierarchy



Top-level Input File (Slide 1 of 3)

```
<<<<<<<<<<<<<<<<<  42: The Mostly Harmless Simulator  >>>>>>>>>>>>>>>>>
************************** Simulation Control **************************
FASTER_THAN_REAL                ! Time Mode (FASTER_THAN_REAL, REAL_TIME,...)
20000.0      0.1                ! Sim Duration, Step Size [sec]
10.0                            ! File Output Interval [sec]
TRUE                            ! Graphics Front End?
Inp_Cmd.txt                     ! Command Script File Name
**************************  Reference Orbits  **************************
4                               ! Number of Reference Orbits
TRUE   Orb_EOS.txt              ! Input file name for Orb 0
TRUE   Orb_ISS.txt              ! Input file name for Orb 1
TRUE   Orb_FGST.txt             ! Input file name for Orb 2
TRUE   Orb_Eros.txt             ! Input file name for Orb 3
*****************************  Spacecraft  *****************************
5                               ! Number of Spacecraft
TRUE  0 SC_LDCM.txt             ! Existence, RefOrb, Input file for SC 0
TRUE  1 SC_ISS.txt              ! Existence, RefOrb, Input file for SC 1
TRUE  1 SC_Dragon.txt           ! Existence, RefOrb, Input file for SC 2
TRUE  2 SC_FGST.txt             ! Existence, RefOrb, Input file for SC 3
TRUE  3 SC_IonCruiser.txt       ! Existence, RefOrb, Input file for SC 4
```

Top-level Input File (Slide 2 of 3)

```
**************************** Environment  ****************************
06 28 2011                    !  Date (Month, Day, Year)
16 45 00.00                   !  Greenwich Mean Time (Hr,Min,Sec)
0.0                           !  Time Offset (sec)
USER_DEFINED                  !  Solar Flux, AP values (TWOSIGMA_KP, NOMINAL ...)
230.0                         !  If USER_DEFINED, enter desired F10.7 value
100.0                         !  If USER_DEFINED, enter desired AP value
IGRF                          !  Magfield (NONE,DIPOLE,IGRF)
8   8                         !  IGRF Degree and Order (<=10)
2   0                         !  Earth Gravity Model N and M (<=18)
2   0                         !  Mars Gravity Model N and M (<=18)
2   0                         !  Luna Gravity Model N and M (<=18)
TRUE                          !  Aerodynamic Forces & Torques
FALSE                         !  Gravity Gradient Torques
FALSE                         !  Solar Pressure Forces & Torques
FALSE                         !  Gravity Perturbation Forces
FALSE                         !  Passive Joint Torques
FALSE                         !  Thruster Plume Forces & Torques
FALSE                         !  Output Environmental Torques to Files
```

Top-level Input File (Slide 3 of 3)

```
******************** Celestial Bodies of Interest ********************
TRUE                             !   Mercury
TRUE                             !   Venus
TRUE                             !   Earth and Luna
TRUE                             !   Mars and its moons
TRUE                             !   Jupiter and its moons
TRUE                             !   Saturn and its moons
TRUE                             !   Uranus and its moons
TRUE                             !   Neptune and its moons
TRUE                             !   Pluto and its moons
TRUE                             !   Asteroids and Comets
**************** Lagrange Point Systems of Interest *****************
FALSE                            !   Earth-Moon
TRUE                             !   Sun-Earth
FALSE                            !   Sun-Jupiter
************************ Ground Stations ***************************
5                                             ! Number of Ground Stations
TRUE   EARTH  -77.0   37.0   "GSFC"           ! Exists, World, Lng, Lat, Label
TRUE   EARTH  -155.6 19.0   "South Point"    ! Exists, World, Lng, Lat, Label
TRUE   EARTH  115.4 -29.0   "Dongara"        ! Exists, World, Lng, Lat, Label
TRUE   EARTH  -71.0 -33.0   "Santiago"       ! Exists, World, Lng, Lat, Label
TRUE   LUNA   45.0   45.0   "Moon Base Alpha" ! Exists, World, Lng, Lat, Label
```

Reference Orbit Definition (1 of 2)

```
<<<<<<<<<<<<<<<<<  42: Orbit Description File   >>>>>>>>>>>>>>>>>
Low Earth Orbit             !  Description
CENTRAL WORLD               !  Orbit Type (CENTRAL, THREE_BODY), Center Type (WORLD, MINORBODY)
::::::::::::::  Use these lines if Body-Centered Orbit  ::::::::::::::::
EARTH                       !  Orbit Center
KEP                         !  Use Keplerian elements (KEP) or (RV)
PA                          !  Use Peri/Apoapsis (PA) or min alt/ecc (AE)
400.0   400.0               !  Periapsis & Apoapsis Altitude, km
200.0  2.0                  !  Min Altitude (km), Eccentricity
0.0                         !  Inclination (deg)
0.0                         !  Right Ascension of Ascending Node (deg)
0.0                         !  Argument of Periapsis (deg)
180.0                       !  True Anomaly (deg)
0.0  0.0  0.0               !  RV Initial Position (km)
0.0  0.0  0.0               !  RV Initial Velocity (km/sec)
```

Reference Orbit Definition (2 of 2)

```
:::::::::::::  Use these lines if Three-Body Orbit  ::::::::::::::::
SUNEARTH                        !  Lagrange system
LAGDOF_MODES                    !  Propagate using LAGDOF_MODES or LAGDOF_COWELL
MODES                           !  Initialize with MODES or XYZ
L2                              !  Libration point (L1, L2, L3, L4, L5)
800000.0                        !  XY Semi-major axis, km
45.0                            !  Initial XY Phase, deg
CW                              !  Sense (CW, CCW), viewed from +Z
0.0                             !  Second XY Mode Semi-major Axis, km (L4, L5 only)
0.0                             !  Second XY Mode Initial Phase, deg (L4, L5 only)
CW                              !  Sense (CW, CCW), viewed from +Z (L4, L5 only)
400000.0                        !  Z Semi-axis, km
60.0                            !  Initial Z Phase, deg
1.05   0.5   0.0                !  Initial X, Y, Z (Non-dimensional)
0.0    0.0   0.0                !  Initial Xdot, Ydot, Zdot (Non-dimensional)
******************* Formation Frame Parameters ************************
L                               !  Formation Frame Fixed in [NL]
0.0   0.0   0.0   123           !  Euler Angles (deg) and Sequence
L                               !  Formation Origin expressed in [NL]
0.0   0.0   0.0                 !  Formation Origin wrt Ref Orbit (m)
```

Example Spacecraft Input File (1 of 4)

```
<<<<<<<<<<<<<<<<<  42: Spacecraft Description File   >>>>>>>>>>>>>>>>>
SpaceX Dragon                  !  Description
"Dragon"                       !  Label
GenScSpriteAlpha.ppm           !  Sprite File Name
PROTOTYPE_FSW                  !  Flight Software Identifier
************************* Orbit Parameters ****************************
ENCKE                          !  Orbit Prop FIXED, EULER_HILL, or ENCKE
CM                             !  Pos of CM or ORIGIN, wrt F
-200.0  0.0  100.0             !  Pos wrt Formation (m), expressed in F
0.0  0.0  0.0                  !  Vel wrt Formation (m/s), expressed in F
*************************** Initial Attitude ***************************
NAN                            ! Ang Vel wrt [NL], Att [QA] wrt [NLF]
0.01    0.02    0.03           ! Ang Vel (deg/sec)
0.0    0.0    0.0    1.0       ! Quaternion
0.0    0.0    0.0    123       ! Angles (deg) & Euler Sequence
***************************  Dynamics Flags  ***************************
DYN_JOINT                      ! Rotation STEADY, KIN_JOINT, or DYN_JOINT
TRUE                           ! Assume constant mass properties
FALSE                          ! Passive Joint Forces and Torques Enabled
FALSE                          ! Compute Constraint Forces and Torques
REFPT_CM                       ! Mass Props referenced to REFPT_CM or REFPT_JOINT
FALSE                          ! Flex Active
FALSE                          ! Include 2nd Order Flex Terms
2.0                            ! Drag Coefficient
```

Example Spacecraft Input File (2 of 4)

```
*************************************************************************
************************* Body Parameters *******************************
*************************************************************************
1                             ! Number of Bodies
================================ Body 0 =================================
200.0                         ! Mass
100.0  200.0  300.0           ! Moments of Inertia (kg-m^2)
0.0  0.0  0.0                 ! Products of Inertia (xy,xz,yz)
0.0  0.0  0.0                 ! Location of mass center, m
IonCruiser.obj                ! Geometry Input File Name
NONE                          ! Flex File Name
```

Example Spacecraft Input File (3 of 4)

```
************************************************************************
************************** Joint Parameters ***************************
************************************************************************
         (Number of Joints is Number of Bodies minus one)
============================== Joint 0 ================================
0 1                           ! Inner, outer body indices
1   213   GIMBAL              ! RotDOF, Seq, GIMBAL or SPHERICAL
0   123                       ! TrnDOF, Seq
FALSE  FALSE  FALSE           ! RotDOF Locked
FALSE  FALSE  FALSE           ! TrnDOF Locked
0.0    0.0    0.0             ! Initial Angles [deg]
0.0    0.0    0.0             ! Initial Rates, deg/sec
0.0    0.0    0.0             ! Initial Displacements [m]
0.0    0.0    0.0             ! Initial Displacement Rates, m/sec
0.0   0.0  0.0   312          ! Bi to Gi Static Angles [deg] & Seq
0.0   0.0  0.0   312          ! Go to Bo Static Angles [deg] & Seq
0.0   0.0  0.0                ! Position wrt inner body origin, m
0.0   0.0  0.0                ! Position wrt outer body origin, m
0.0   0.0  0.0                ! Rot Passive Spring Coefficients (Nm/rad)
0.0   0.0  0.0                ! Rot Passive Damping Coefficients (Nms/rad)
0.0   0.0  0.0                ! Trn Passive Spring Coefficients (N/m)
0.0   0.0  0.0                ! Trn Passive Damping Coefficients (Ns/m)
```

Example Spacecraft Input File (4 of 4)

```
*************************** Wheel Parameters ***************************
0                              ! Number of wheels
=============================  Wheel 0  ================================
0.0                            ! Initial Momentum, N-m-sec
1.0   0.0   0.0                ! Wheel Axis Components, [X, Y, Z]
0.14   50.0                    ! Max Torque (N-m), Momentum (N-m-sec)
0.012                          ! Wheel Rotor Inertia, kg-m^2
**************************** MTB Parameters ****************************
0                                    ! Number of MTBs
==============================  MTB 0  =================================
180.0                          ! Saturation (A-m^2)
1.0   0.0   0.0                ! MTB Axis Components, [X, Y, Z]
************************* Thruster Parameters **************************
0                              ! Number of Thrusters
==============================  Thr 0  =================================
 1.0                           ! Thrust Force (N)
-1.0  0.0  0.0                 ! Thrust Axis
 1.0  1.0  1.0                 ! Location in B0, m
**************************** CMG Parameters ****************************
0                              ! Number of CMGs
===============================  CMG 0  ================================
1                              ! CMG DOF (typically 1 or 2)
0.0  0.0  0.0  123             ! Initial Gimbal Angles [deg] and Seq
0.0  0.0  0.0                  ! Initial Gimbal Angle Rates, deg/sec
-90.0  0.0  -54.74  123        ! Static Mounting Angles [deg] and Seq
0.12                           ! Rotor Inertia, kg-m^2
75.0                           ! Momentum, Nms
1.0  0.0  0.0                  ! Max Gimbal Angle Rates, deg/sec
```

Geometry Definition

- Geometry uses Wavefront "Object" format
  - ASCII, human-readable
  - Can hand-generate models *in theory*
  - In practice, you'll want mechanical help
- I use Wings3D solid modeling software
  - Free, multi-platform
- 42 can support myriad-polygon models, limited mainly by your patience
  - I get impatient at about 10,000 polys
- Note that aerodynamic and solar-pressure force and torque computations use these models
  - Interior polygons, self-shadowing are error sources

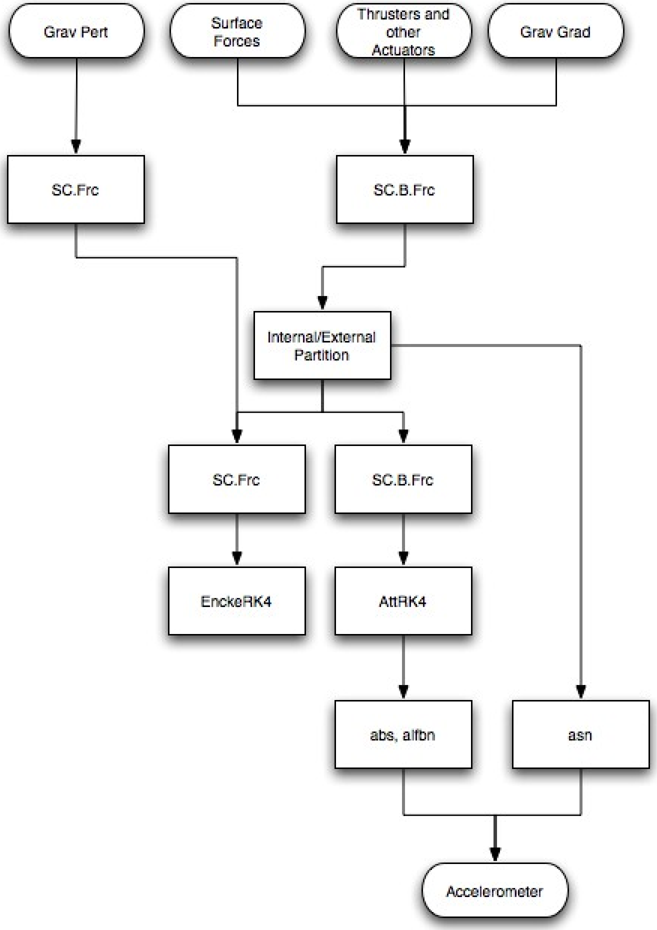
```
# Exported from Wings 3D 0.99.00b
mtllib Altair.mtl
o cylinder9
#12 vertices, 8 faces
v 2.11050391 1.21850000 -1.80666667
v 2.11050391 -1.21850000 -1.80666667
[...]
vn 0.0000000e+0 9.1113913e-17 1.00000000
vn 0.50000000 0.86602540 0.0000000e+0
[...]
g cylinder9_SHINY_WHITE
usemtl SHINY_WHITE
f 1//1 6//16 5//13 4//10 3//7 2//4
f 1//2 7//20 12//35 6//17
[...]
```

Excerpt from Altair.obj

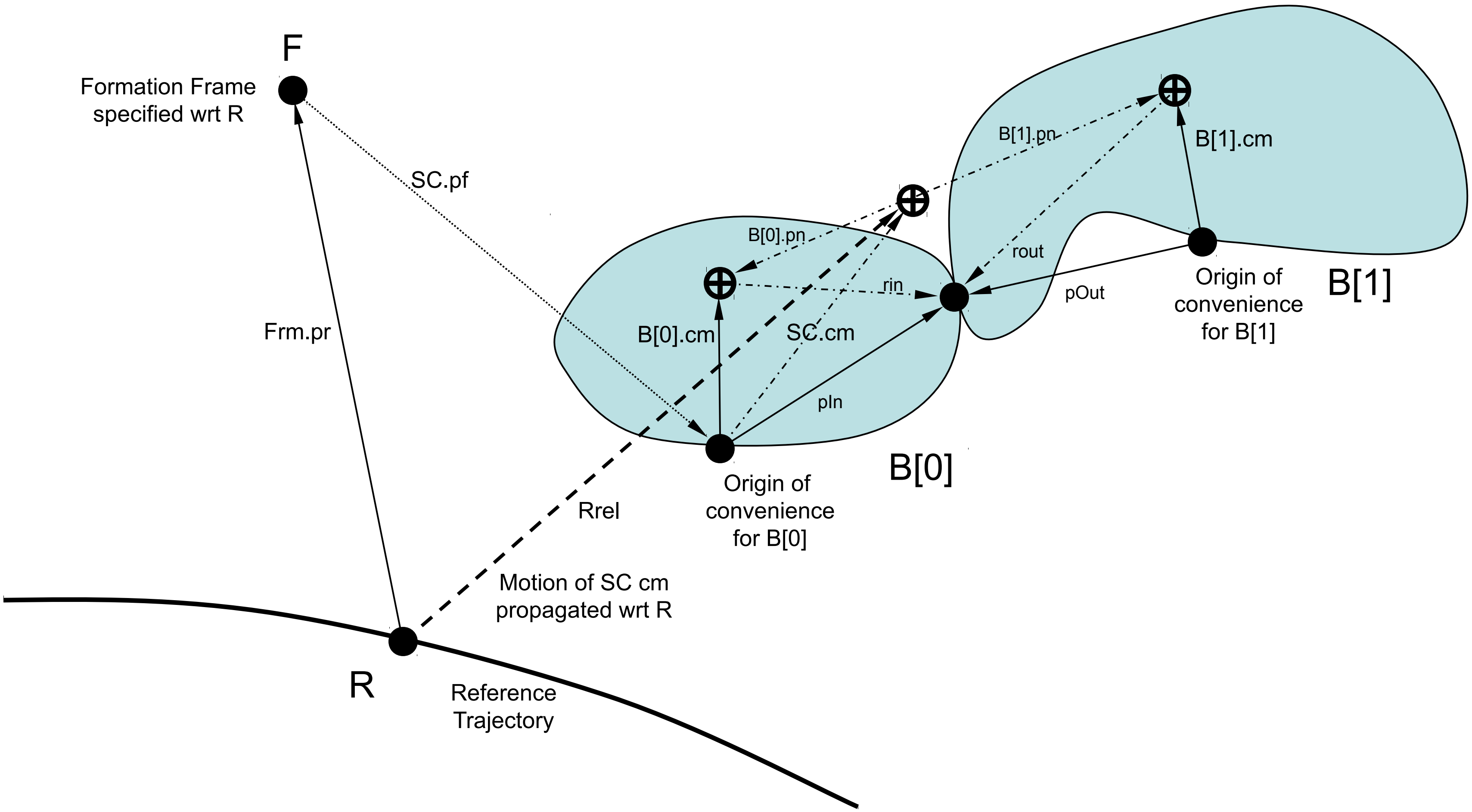
Reference Slides

Force Flow

- Careful accounting needed to decouple orbit and attitude equations of motion
- Accelerometer model also requires careful accounting



Bookkeeping of Translational States



Geometry for Surface Forces

Moment of forces taken about
Body's mass center