Kerbal Space Program - Bug #24582

ManeuverNode and/or PatchedConicSolver code has single precision roundoff error (Quaternion or floating point Math somewhere)

01/14/2020 07:06 PM - lamont

Status:	Ready to Test	**Start date:**	01/14/2020
Severity:	Low	**% Done:**	80%
Assignee:	victorr		
Category:	Physics		
Target version:	1.11.0		
Version:	1.8.1	**Language:**	English (US)
Platform:	Linux, OSX, Windows	**Mod Related:**	No
Expansion:	Core Game		

Description

This is a dup of https://bugs.kerbalspaceprogram.com/issues/14076 but since that is three years old and stalled hard, I thought I'd open this with a more actionable request.

For reference that bug addresses:

https://github.com/MuMech/MechJeb2/issues/866

The bug is that when a ManeuverNode is placed via `vessel.patchedConicSolver.AddManeuverNode` and then retrieved via `vessel.patchedConicSolver.maneuverNodes[0].GetBurnVector()` that there is a difference in the computed burn vectors, which can be significant.

I've gone through the MechJeb code which converts the burn vector into the frenet frame and there's no loss of precision in anything MechJeb is doing there, and it is using Vector3d.Dot to do the projection of the vector onto the coordinate axis.

Debugging the before and after of placing a maneuver node looks like:

```
[LOG 10:31:39.939] placed dV = [1624.97467726249, -997.372898790216, 946.492620930603]
[LOG 10:31:39.939] placed UT = 299088.096336164

[LOG 10:31:39.939] actual dV = [1625.12725957217, -997.372899802009, 946.236477800571]
[LOG 10:31:39.939] actual UT = 299088.096336164
```

That is logging what I want, vs. what I read back from GetBurnVector() after doing the round trip through KSP. Note that the timestamp is identical so there's no rotation-frame issues here.

While this deviation is small as with all single-precision things in KSP this can have large effects, particularly on flyby altitudes on interplanetary trips, and makes implementing a flyby transfer planner very difficult.

I'm not sure this easily replicates with the stock maneuver node editor since you have to be working in the burn vector frame, not in the frenet frame. The prograde/radial/normal values of the maneuver node will not change. But if you call GetBurnVector() and use those values to re-generate the frenet values and then write the ManeuverNode again, then if you are under the inverse rotation threshold (i.e. very low kerbin parking orbit) then the rotation being applied to the burn vector will cause the loss of precision on the conversion to the Quaternion to become apparent on every single tick.

That does suggest a solution to the maneuver node editor problem which is to make certain to work in the frenet frame consistently and not do any conversions to the burn vector, but that can't help at all with trying to precisely place maneuver nodes for flybys where the problem isn't in trying to maintain the values of the Quaternion but that the single-precision values are just not sufficiently accurate.

This has been reported against KSP 1.2.x-1.8.x and should replicate against every version of KSP.

And please don't just ask to replicate this against stock unmodded KSP, the point is that this API makes modding goals harder / impossible.

History

#1 - 01/14/2020 07:26 PM - lamont

That's actually an old MJ bug, the more recent one is https://github.com/MuMech/MechJeb2/issues/1002, I'm going to close 866 as a dup.

#2 - 01/15/2020 06:43 PM - lamont

Well the title of this is likely inaccurate.

And the DeltaV values written to the node are faithfully recovered from the node.

There still is some single-precision floating point roundoff error happening, but it seems to not necessarily be in the ManeuverNode object but possibly in the patched conic solver somewhere else that consumes the maneuver node (which is consistent with the effect showing up mostly on interplanetary transfers with SOI changes where the solver would be involved).

#3 - 01/15/2020 06:52 PM - lamont

- Subject changed from ManeuverNode has single precision roundoff error (nodeRotation should be a QuaternionD not a Quaternion) to ManeuverNode and/or PatchedConicSolver code has single precision roundoff error (Quaternion or floating point Math somewhere)

#4 - 01/18/2020 06:07 AM - lamont

Yeah PatchedConicSolver.CheckNextManeuver and ManeuverNode.GetPartialDv both look like they use Quaternion.LookRotation instead of QuaternionD.LookRotation (probably because no QuaternionD.LookRotation exists -- which really should get implemented).

This will show up as 1e-8 level loss of precision if you just:

1. Take a desired burn vector in cartesian coordinates as a Vector3d
2. Convert that to the KSP maneuver node prograde/radial/normal frenet frame
3. Create or update a maneuver node with those values
4. Use ManeuverNode.GetBurnVector() and compare the output with the burn vector that you wanted in step #1

#6 - 03/04/2020 04:55 PM - victorr

- Status changed from New to Confirmed

- Assignee set to victorr

- % Done changed from 0 to 10

#7 - 01/05/2021 01:10 AM - JPLRepo

- Status changed from Confirmed to Ready to Test

- Target version set to 1.11.0

- % Done changed from 10 to 80