DR

Dean R Edstrom JR | Elliott S Mccrory; John S Diamond; Alexander A Valishev | 2:24 PM

Re: Orbit for IOTA thru ACNET

2:34 PM, August 31, 2018

The structure that we want the data in is just a set of arrays and a device to specify which turn we're looking at. For initial commissioning, I made a set of orbit devices that can accomodate two consecutive orbits. These are:

```
N:IBPMOH[]
N:IBPMOV[]
N:IBPMOS[]
```

Where each element is filled according to:

Turn n	**Turn n+1**	
N:IBPMOx[0]=N:IBA1Cx[n]	N:IBPMOx[21]=N:IBA1Cx[n+1]	
N:IBPMOx[1]=N:IBA2Rx[n]	N:IBPMOx[22]=N:IBA2Rx[n+1]	
N:IBPMOx[2]=N:IBA3Rx[n]	N:IBPMOx[23]=N:IBA3Rx[n+1]	
N:IBPMOx[3]=N:IBB1Rx[n]	N:IBPMOx[24]=N:IBB1Rx[n+1]	
N:IBPMOx[4]=N:IBB2Rx[n]	N:IBPMOx[25]=N:IBB2Rx[n+1]	
N:IBPMOx[5]=N:IBC1Rx[n]	N:IBPMOx[26]=N:IBC1Rx[n+1]	
N:IBPMOx[6]=N:IBC2Rx[n]	N:IBPMOx[27]=N:IBC2Rx[n+1]	
N:IBPMOx[7]=N:IBD1Rx[n]	N:IBPMOx[28]=N:IBD1Rx[n+1]	
N:IBPMOx[8]=N:IBD2Rx[n]	N:IBPMOx[29]=N:IBD2Rx[n+1]	
N:IBPMOx[9]=N:IBE1Rx[n]	N:IBPMOx[30]=N:IBE1Rx[n+1]	**Etc**
N:IBPMOx[10]=N:IBE2Rx[n]	N:IBPMOx[31]=N:IBE2Rx[n+1]	
N:IBPMOx[11]=N:IBE2Lx[n]	N:IBPMOx[32]=N:IBE2Lx[n+1]	
N:IBPMOx[12]=N:IBE1Lx[n]	N:IBPMOx[33]=N:IBE1Lx[n+1]	
N:IBPMOx[13]=N:IBD2Lx[n]	N:IBPMOx[34]=N:IBD2Lx[n+1]	
N:IBPMOx[14]=N:IBD1Lx[n]	N:IBPMOx[35]=N:IBD1Lx[n+1]	
N:IBPMOx[15]=N:IBC2Lx[n]	N:IBPMOx[36]=N:IBC2Lx[n+1]	
N:IBPMOx[16]=N:IBC1Lx[n]	N:IBPMOx[37]=N:IBC1Lx[n+1]	
N:IBPMOx[17]=N:IBB2Lx[n]	N:IBPMOx[38]=N:IBB2Lx[n+1]	
N:IBPMOx[18]=N:IBB1Lx[n]	N:IBPMOx[39]=N:IBB1Lx[n+1]	
N:IBPMOx[19]=N:IBA3Lx[n]	N:IBPMOx[40]=N:IBA3Lx[n+1]	
N:IBPMOx[20]=N:IBA2Lx[n]	N:IBPMOx[41]=N:IBA2Lx[n+1]	

where 'n' is the primary turn being displayed on the orbit display (n=1...7000 or whatever the length of the BPM buffer is) and x=H,V,S. It would also be good to have an N:IBPMOZ[] with the longitudinal position of each BPM around the ring, so that an accurate azimuthal plot can be made using a standard array plotting package... This could just be a cache device, but it probably makes sense to just host it in the BPM front end along with the other ACNET devices.

The user-selected integer number of the primary orbit for display ('n') should also be an acnet device- given the other devices, this should be N:IBPMON, so if I set N:IBPMON to 1 the front end will fill the N:IBPMOx devices with the first two turns of BPM data, but if I set it to 100, it will fill the device with turns 100 and 101. It would be nice if this is read automatically when the TBT buffer is filled (to get the latest data), but also on changes to N:IBPMON (so that we can cycle through the orbits after injection).

If you'd like to see the acl that I was filling these devices with previously, it can be found on a clx xterm: /export/home1/edstrom/acl/fast_iota_orbit.acl