Where We Are with HCC

Created by Adam Sweet 4/10/2000

1.0 Old HCC to new HCC

Benoit has been developing the next version of HCC. There are several changes, both internal and external, to the language. The biggest external change is the change in syntax - the syntax of the new version is similar to Java. This means that we will need to change the current models over, but that should be a "do-once" task.

My understanding is that the new HCC is usable now, although there are still some bugs and the API isn't fully developed. It runs under Linux, Solaris, Irix, and probably most any other Unix system (Win NT version will be done later). Benoit has a CVS repository set up for HCC, and has converted many of the example programs to the new format. He has also converted some more complicated models, to test how they work under the new HCC. I have not been using it yet, mostly since we haven't declared an official release.

2.0 Current models

All of the current models are done in the old version of HCC. A brief description is below.

2.1 Rover

The rover model is a simulation of the Marsokhod planetary rover, which is a type of rover that the IMG has used for field tests at Ames. It is probably the largest model that we have. The idea behind this simulation is that HCC would be receiving a list of time-stamped rover commands from another source, and would then simulate the response of the rover to those commands. We haven't done work on it for some time, as the group has been creating a new style of rover (the Fido-style rocker-bogie type) and there hasn't been a driving force to create a different simulation. However, many components of the model should be adaptable to a new rover.

The current version of the simulation is fairly tightly integrated with the Marsmap VR environment, both for display of the results and to calculate certain values for the simulation. It could be separated, but would take some

effort. The current rover HCC files are not well documented yet, but I will do that before summer. Some of the functionality is

- Driving commands: BaseDrive, BaseAbort, etc.
- Imaging commands: ICPanTilt, ICSnap, ICMosaic
- Arm commands
- Rover dynamics with a simple wheel soil model
- Wheel faults: stuck, slipping, encoder broken, encoder skipping

2.2 PSA

The PSA is a small sphere that will autonomously fly itself around on the space station, performing several labor-saving tasks. It will use fans for propulsion, have a variety of environmental sensors, and a videocamera for teleoperation/inspection.

The current PSA model is simple, just an implementation of the dynamics of the PSA. That is, it implements the equations of motion for a rigid sphere with six additional rotating components (the fans providing thrust). As of now, there are no failure modes or command messages that can be sent to the program. There is a simple path planner, which defines a desired trajectory for the PSA, and a simple controller (P control, perfect state info), allowing the PSA to follow that path. It can be run as a standalone program, or as part of the VR environment. There is a considerable amount of additional functionality added by the VR program, but HCC does not depend on any of this (yet). The files are fairly well documented. The PSA model is probably the best one to use for getting up and running with HCC.

2.3 ISPP

ISPP stands for “in situ propellant production”. This is a broad category of devices that convert Mars resources to resources usable by a sample return mission or a robotic/human colony. We have some models of two ISPP devices, the sorbtion pump and the RWGS. The sorption pump takes in CO2 from the Mars atmosphere, and stores it for other systems. Our model for the sorbtion pump is fairly simple, and is fairly well documented. The RWGS system essentially uses a chemical process to convert the CO2 on Mars to O2.

This model is more complex, and only halfway documented. Neither system interfaces with a virtual reality display – since both are immobile devices, it didn't seem that there was much point to doing it.