PRECAST (Portability, reproducibility and exception handling of control software on host and target platforms) Abstract

There is great interest in the space-software community for model based algorithm and software development but only partial benefits are obtained if the verification and validation has to be carried out on the *target* embedded system, as the development environment is not representative enough of it.

This especially applies to the development of numerical computation algorithms, such as the ones used for GNC/AOCS systems as well as for scientific algorithms.

Numerical reproducibility issues among different hardware and software environments have accompanied the development of numerical software since its historical beginnings and despite the advent of standards such as the IEEE-754 for floating-point arithmetic and other standards like the C99 and POSIX have greatly clarified the situation, numerical reproducibility issues between the *host* systems used for development (normal PCs) and the embedded *target* systems (on-board processors) impede a proper and early validation on host, as well as the investigation of problems observed during AIT phases on target systems.

These numerical reproducibility issues affect the accuracy and the error signaling behavior, including exceptions and special-value generation (e.g., NaN and INF).

We analyzed the numerical reproducibility issues when using Matlab and Simulink for algorithm development on PC platforms and the implications of autocode generation when running the results on host and target platforms. As host platforms we analyzed the differences among the x86-64 processors and for the target side especially the LEON processor family and their FPUs.

This evaluation has been based on the use of the MLFS library as a possible solution for numerical reproducibility of elementary mathematical procedures and its test suite BLTS to assess the exceptions behavior.

We have been able to produce guidelines which when applied yield reproducible numerical results (to the last bit) on Model in the loop (MIL - host), Software in the loop (SIL - host), and Processor in the loop (PIL - target) executions of a non-linear system including elementary mathematical procedures as well as elementary arithmetical operations.

Although we cannot prove our guidelines as universally applicable, understanding and applying them can imply a significant win in advancing on model based numerical algorithms development for the space industry.