IAF SPACE SYSTEMS SYMPOSIUM (D1)
Interactive Presentations - IAF SPACE SYSTEMS SYMPOSIUM (IP)

Author: Mr. Sudeesh Balan
ISRO Satellite Centre (ISAC), India, sudeesh@isac.gov.in

Mrs. Daffini Manoja J
U R RAO SATELLITE CENTRE (URSC), India, daffinimanoja@gmail.com
Mrs. Deepika Jindal
U R RAO SATELLITE CENTRE (URSC), India, deepika@isac.gov.in

VERIFICATION OF ATTITUDE AND ORBIT CONTROL SYSTEM ON-BOARD AUTONOMY SOFTWARE USING MODEL CHECKING

Abstract

Satellite on-board software is mission critical in nature and needs rigorous verification. However due to the complexity, stringent project deadlines and large states spaces, exhaustive testing is not possible using conventional testing methods. This can however be addressed through model checking which is a formal method used for system verification and validation. Given a finite state model of a system and formal property, this technique systematically checks whether this property holds for a given state in that model. This paper addresses how the real time aspects of the Attitude and Orbit Control System together with the autonomy features are being modeled and used for the verification and validation. Attitude and Orbit Control System is responsible for maintaining the attitude of the satellite and for performing fault detection, isolation, and recovery decisions of the satellite. Designing and verifying real-time systems requires a model of the tasks constituting the control program and model of the real time environment. As the interaction between tasks of the control program and environment is assumed to be time-sensitive it is important that our model also takes care the overhead introduced by the particular algorithm applied for scheduling tasks.

The application of Model Checking involves: 1) Modeling phase where system behavior is described using a model description language. We used abstraction techniques to control the state-space of the model to a reasonable size. 2) Property Identification phase where properties derived from the system requirements are identified. Linear Temporal Logic / Computational Tree Logic are used to specify these properties. 3) Analysis Phase where the Model Checker tests for property violation and lists counter example if any. The counter example describes an execution path that leads from the initial system state to a state that violates the property being verified. The model checker can explore all reachable states. So, any state or transition can be guaranteed to be covered. Even the subtle errors that remain undiscovered using conventional testing can potentially be revealed using model checking.

The on-board software for Attitude and Orbit Control System is written in Ada and the modeling is done using NuSMV2. Properties are derived from system specifications. We propose a methodology for deriving properties from specifications, verify with model checker and analyze the observations in a Software In Loop Simulator to get further insights into the performance.