Harmonic Balance Codes

Software Assurance Classification Report

Text highlighted in gray is unique/updated for each project.

**Prepared by:**

|  |  |
| --- | --- |
| **Name** | **Date** |
| Leslie J. Johnson, NASA LaRC  Software Assurance | 8/9/2013 |

**NASA Langley Research Center**

**Hampton, VA 23681**

**Revision and History Page**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision**  **No.** | **Change**  **No.** | **Description** | **Release**  **Date** |
|  |  | Initial Release | 8/9/2013 |

Table of Contents

SECTION PAGE

1. INTRODUCTION 4

1.1 Background 4

2. REFERENCE Documents 5

3. SUMMARY 5

3.1 Software Classification 5

3.2 Software Safety 5

3.3 Software Assurance Effort 6

Appendix A: ACRONYMS 6

# INTRODUCTION

This report contains the software assurance classification assessment which identifies and evaluates the characteristics of software in determining the software’s classification, software safety-criticality, and level of software assurance to be applied to a Project.

## Background

In 1979, four FORTRAN programs, designated XX, YY, ZZ, and AA, were written to help verify concepts and procedures which were being developed to establish foundations for adequate representation and treatment of airframe structures in design analysis of helicopter vibrations. The original FORTRAN programs were converted from mainframe-based programs to PC-based programs in 1996.

The current intention is to use an existing elastic-line finite-element model of a real helicopter airframe and to develop a simplified model of a real rotor in edgewise flow to provide a more substantial system for analysis by the four codes.

It is expected that some coding modifications will be needed to meet the numerical processing requirements of the new math models. While there is no intention to distribute any of the modified codes, it would seem prudent to prepare for such a possibility. Because the codes have a major computational feature in common, namely the use of matrix harmonic balance techniques to solve the different sets of periodic-coefficient differential equations of motion treated by the codes, it seems reasonable to consider them collectively for the classification assessment.

A brief description of the purpose of each of the programs is given below:

1. XX is a program for harmonic balance solution of linear, second-order, non-homogeneous matrix differential equations. MHB is intended for performing a full-up system analysis, that is, the user-supplied equations of motion to be solved are those of the coupled rotor-airframe system.
2. YY is a program that uses the linearized differential equations of motion for a rotor with a free hub to calculate both the vibratory loads acting at the hub assuming no hub motion and the hub load increments resulting from unit imposed harmonic displacements of the hub degrees of freedom by the method of harmonic balance.
3. ZZ is a program for rotor-airframe coupling and vibratory response analysis of rotorcraft systems by the method of harmonic balance.
4. AA is a program that couples rotor hub impedances computed internally using a user-supplied linearized math model of the rotor with a free hub to user-supplied airframe hub mobilities and calculates the vibratory responses of the coupled system and the associated interface forces by the method of harmonic balance. SHAKE is the implementation of the impedance/mobility matching technique for rotor-airframe coupling and vibratory response analysis.

# REFERENCE Documents

The following documents were used or referenced in the development of this report:

|  |  |
| --- | --- |
| **Document No.** | **Document Title** |
| NPR 7150.2A | NASA Software Engineering Requirements |
| NASA-STD-8739.8 | NASA Software Assurance Standard |
| NASA-STD-8719.13B | NASA Software Safety Standard |
| LAPD 5300.1 | Program/Product Assurance |
| LPR 7150.2 | LaRC Software Engineering Requirements |
| LPR 5300.1 | Product Assurance Plan |
| LMS-CP-4754 | Software Assurance (SA) for Development and Acquisition |
| LMS-CP-7150.6 | Class E Software |

# SUMMARY

The following paragraphs summarize the results and describe the details used to determine the software classification assessment for this report.

## Software Classification

According to LPR 7150.2, the software component for this Project is classified as Class E – Small Light Weight Design Concept and Research and Technology Software which is defined as

1. *Software developed to explore a design concept or hypothesis, but not used to make decisions for an operational Class A, B, or C system or to-be built Class A, B, or C system, or*
2. *Software used to perform minor desktop analysis of science or experimental data.*

As such, the Project shall follow the instructions and complete the compliance matrix in LMS-CP-7150.6, *Class E Software*, which applies to all Class E software that is not safety-critical.

## Software Safety

The Software Safety Litmus Test below is applied to all projects with software to determine if the software is safety-critical. If the software is determined to be safety-critical, then the project must adhere to the NASA-STD-8719.13, NASA Software Safety Standard.

A software component is considered safety-critical if it meets **any** of the following criteria:

|  |  |
| --- | --- |
| **Criteria:** | **Software**  **components** |
| 1. Resides in a safety-critical system (as determined by a hazard analysis) **AND** at least one of the following apply: | No |
| 1. Causes or contributes to a hazard |  |
| 1. Provides control or mitigation for hazards |  |
| 1. Controls safety-critical functions |  |
| 1. Processes safety-critical commands or data |  |
| 1. Detects and reports, or takes corrective action, if the system reaches a specific hazardous state |  |
| 1. Mitigates damage if a hazard occurs |  |
| 1. Resides on the same system (processor) as safety-critical software |  |
| 1. Processes data or analyzes trends that lead directly to safety decisions (e.g., determining when to turn power off to a wind tunnel to prevent system destruction) | No |
| 1. Provides full or partial verification or validation of safety-critical systems, including hardware or software subsystems. | No |

The software components in this Project do not reside in a safety-critical system; process data or analyze trends that lead directly to safety decisions or provide full or partial verification or validation of safety-critical systems.

The LaRC Safety and Mission Assurance Office have determined that the software components in this Project are not safety-critical.

## Software Assurance Effort

The software assurance effort is based on the software class and impacts from potential failure. In accordance with LMS-CP-4754 software assurance is not applicable for non-safety critical Class E software developments.

# Appendix A: ACRONYMS

|  |  |
| --- | --- |
| CP | Center Process |
| LaRC | Langley Research Center |
| LAPD | Langley Policy and Directives |
| LMS | Langley Management System |
| LPR | Langley Procedural Requirements |
| NASA | National Aeronautics and Space Administration |
| NPR | NASA Procedural Requirement |
| SA | Software Assurance |
| STD | Standard |