HARP-Opt v3.0 Development Plan

Danny Sale and Michael Lawson

May 25, 2012

1 Objective

This document describes the development for HARP-Opt version 3.0, which will be released by September 30, 2012. The objectives of this development effort are:

- Make the WT-Perf FORTRAN program used by HARP-Opt "callable" as a MATLAB function (note that WT-Perf will still be implemented in FORTRAN). This will remove the need for computationally expensive disk input-output each time WT-Perf is run by HARP-Opt.
- Remove the graphical users interface (GUI) from the HARP-Opt code. Make the code operate using a text input file.
- Develop a stand alone GUI, referred to as HARP-GUI herein, that writes the text input file for HARP-Opt
- Implement a patter search optimization algorithm to improve repeatability of the results

These development objectives are described in more detail in the following section.

2 Discussion of Development Objectives

2.1 Interoperate the callable version of WT-Perf into HARP-Opt

The current version of HARP-Opt runs WT-Perf as an external program. This requires HARP-Opt to write WT-Perf input files and read WT-Perf output files from disk. This is a computationally expensive operation that contributes significantly to the total HARP-Opt run time.

Andrew Platt is currently developing a version of WT-Perf that is callable as a FORTRAN module/function (details are to be determined). The MathWorks provides tools to make this possible.

A significant amount of the HARP-Opt code is dedicated to file input-output, so the ability to call WT-Perf from MATLAB should significantly simplify the code.

2.2 Platform Independence

The new release of HARP-Opt should be platform independent (i.e. able to run on any operating system). This will require several things:

- Sub-programs that are called by HARP-Opt will need to be compilable from any operating system.
- Anything else?

2.3 No GUI for the Main HARP-Opt Code!

GUIs and graphics of any kind should be completely removed from the code. I have already removed the status bar from the airfoil interpolation operation. The graphical output that is displayed during the optimization run should be replaced with a text based output. This should simplify the code. This will also allow the program to be run from the command line and for batch jobs consisting of many HARP-Opt runs to be on high-performance-computing (HPC) resources.

2.4 GUI to Write Input Files

Develop a HARP-Opt pre-processor GUI (HARP-Opt-Pre, need a better name, but this will work for now) to write an input file for the HARP-Opt code. HARP-Opt-Pre may be developed so it calls the HARP-Opt code, however, the code of the HARP-Opt-Pre will be completely separate from the HARP-Opt code.

3 Long Term Development Goals

- **Open-Source:** Move to all open-source packages. Coding will be in Python and optimization will by in some open-source optimization package, such as pyOpt or DAKOTA
- **Fatigue Analysis Capabilities:** Consider fatigue loads in the optimization process.
- Danny, I'm sure you have a bunch of stuff you want to add here
- Andrew Ning has some improvements to WT-Perf that will allow for the use of gradient search methods to be used.