Gas Turbine Propulsion Toolbox: User Manual

Christopher Chinske

February 21, 2020

1 Introduction

The Gas Turbine Propulsion Toolbox (GTPT) is a collection of programs that can be used to study gas turbine engine performance. The programs are implemented as GNU Octave functions and scripts. However, the programs should run correctly in MATLAB.

GTPT is based on the engine models contained in *Aerothermodynamics of Gas Turbine and Rocket Propulsion* by Gordon C. Oates. GTPT reproduces some of the functionality of the Engine Cycle Analysis Program (ECAP) and the Engine Off-Design Performance Program (EOPP), which are distributed with the book by Oates.

GTPT also includes example code for engine performance modeling (EPM). For example, if you know the specific fuel consumption for an engine at a particular reference point, and you know the engine's pressure ratio and bypass ratio, then GTPT can be used to determine a feasible set of parameters (e.g., component pressure ratios, polytropic efficiencies, etc.) that match the engine's known performance. These parameters can then be used in an engine performance model to estimate the engine's performance at an off-design point.

2 How to Use This Manual

Commands are in mono-spaced typeface. For example:

```
[f_mdot, s, inputs] = ideal_turbojet
```

Commands that could have more than one form are shown as `<example>`
Arguments that are optional are shown as `[example]`

3 Concept of Operations

GTPT operations can be divided into four categories:

1. Ideal Cycle Analysis
2. Non-Ideal Cycle Analysis
3. Off-Design Performance
4. Engine Performance Modeling.

In general, GTPT can be used to analyze turbojet and turbofan engines.

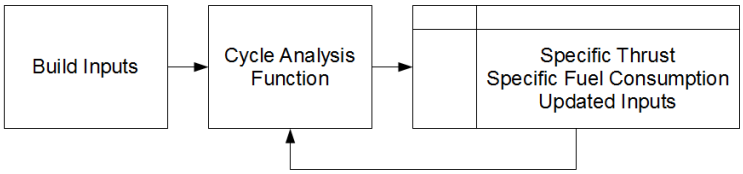

Figure 1: General program flow

Figure **??** shows the general program flow for Ideal Cycle Analysis, Non-Ideal Cycle Analysis, and Off-Design Performance. The general process is as follows.

Build Inputs The user builds a structure array where the fields of the structure array are the input parameters. In many cases, the cycle analysis function will prompt the user for the input parameters if an input structure array is not provided as an argument to the function. See ICD-GTPT-001 for a detailed description of the interfaces for each function.

Cycle Analysis Function The user calls the cycle analysis function.

Output In general, each cycle analysis function will output specific thrust and specific fuel consumption. Some functions will output an updated input structure array.

Figure **??** shows the process for generating an engine performance model based on limited engine reference data. GTPT includes example code for determining a feasible set of parameters and for engine performance modeling. First, optimization functions interface with a non-ideal cycle analysis function to determine a feasible set of parameters. Then, these parameters are used to build an engine performance model that interfaces with an off-design performance function.

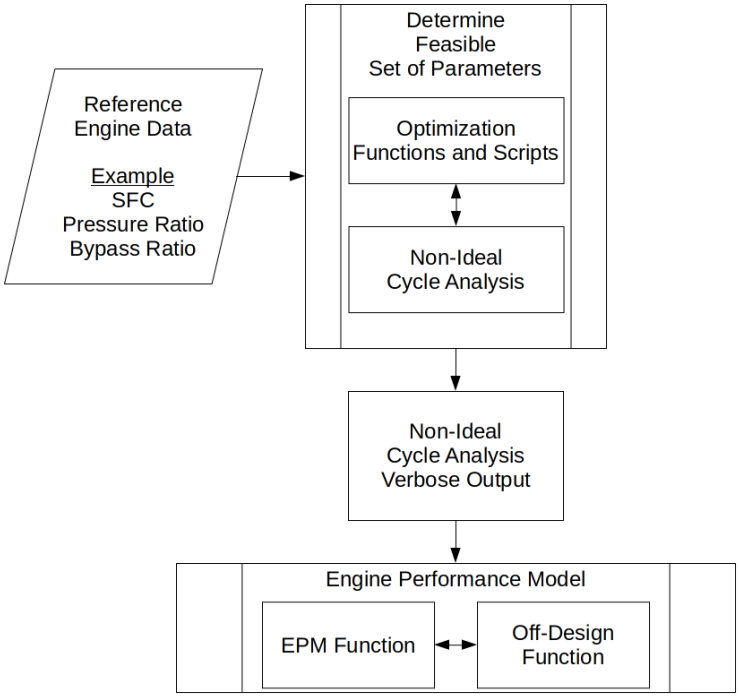

Figure 2: Engine performance modeling

4 Procedures

4.1 General

To display a short message describing the calling syntax of a function, type:

```
help <function>
```

Be sure that your load path is correctly configured. In particular, for engine performance modeling, the following directories must be on your load path.

- optimize
- epm

4.2 Build Input Structure Array

4.2.1 General

In many cases, the cycle analysis function will prompt you for the input parameters if an input structure array is not provided as an argument to the function. For example:

```
[f_mdot, s, inputs] = ideal_turbojet
```

Alternatively, you can build an input structure array by typing:

```
inputs = build_inputs
```

BUILD_INPUTS may prompt you for inputs that do not apply to your particular problem. For example, bypass ratio is not required for an analysis of a turbojet engine.

BUILD_INPUTS_OD_TURBOJET and BUILD_INPUTS_OD_TURBOFAN should be used to build the input structure array for OD_TURBOJET and OD_TURBOFAN, respectively.

Also, you may always choose to manually build the input structure array.

4.2.2 Vector Input

In general, each input parameter is a single value. However, one input parameter may be a vector. This allows computation of the outputs (primarily specific thrust and specific fuel consumption) as a function of an input variable.

4.3 Ideal Cycle Analysis

- Build the input structure array.
- Call the ideal cycle analysis function.

List of Functions
IDEAL_TURBOJET
IDEAL_TURBOJET_AB
IDEAL_TURBOFAN
IDEAL_TURBOFAN_AB
IDEAL_TURBOFAN_MIXED

4.4 Non-Ideal Cycle Analysis

- Build the input structure array.
- Call the non-ideal cycle analysis function.

List of Functions
NONIDEAL_TURBOJET
NONIDEAL_TURBOFAN
NONIDEAL_TURBOFAN2

4.5 Off-Design Performance

- Build the input structure array.
- Call the off-design performance function.

List of Functions
OD_TURBOJET
OD_TURBOFAN

4.6 Engine Performance Modeling

- Configure your load path
  - `addpath optimize`
  - `addpath epm`
- Create an optimization script based on optimize/s_seek.m
  - This script defines the lower bounds and upper bounds of the free (i.e., unknown) variables.
  - The function SQP is used as the solver.
- Create an objective function based on optimize/phi.m
  - This function defines the fixed (i.e., known) variables.
  - This function calls a cycle analysis function.
  - This function computes the absolute error between the output of the cycle analysis function and the known measure of performance.
  - The absolute error is returned as the objective function to be minimized.
- Run the optimization script
- Use the output from the optimization script to build an engine performance model based on epm/example_epm.m