Proceedings of ASME Turbo Expo 2005:
Land Sea & Air
June 6-9, 2005, Reno-Tahoe, Nevada, USA

GT2005-68202

IMPLEMENTATION OF ENGINE LOSS ANALYSIS METHODS IN THE NUMERICAL PROPULSION SYSTEM SIMULATION

Bryce A. Roth, Erin M. McClure, Travis W. Danner
Georgia Institute of Technology
Atlanta, GA 30332-0150

ABSTRACT

This paper describes the implementation and application of a new set of thermodynamic loss analysis tools in the Numerical Propulsion System Simulation. This analysis tool set is intended to enable fast, accurate estimation of losses in an engine cycle model with minimal effort on the part of the user. The basic thermodynamic concepts and analysis methods are first described. Next, the implementation of the necessary thermodynamic calculation functions is described. These functions are intended to be used in conjunction with a general-purpose loss analysis element to facilitate estimation of all losses in an engine cycle model. The loss analysis element is described in detail and is subsequently used to analyze a mixed flow turbofan engine. Typical performance and loss results are presented. The resultant detailed loss information is not normally available when using standard cycle analysis methods. The information gained from this analysis is useful in that it yields insight into the underlying losses that contribute to the overall engine performance.

INTRODUCTION

The fundamental thermodynamic function of all aircraft engines is to convert work potential stored in the chemical bonds of the fuel into useful thrust work. The efficiency with which this occurs is typically measured in terms of a system-level metric such as specific fuel consumption. This overall system performance is the result a thermodynamic cycle consisting of many dozens of flow processes that occur inside an engine, each process contributing a small portion of loss that is ultimately manifested as an incremental reduction in system performance. It is logical to think that having detailed estimates of these various losses is a necessary prerequisite to reducing them and therefore improving the system as a whole. Unfortunately, all past and current cycle analysis codes are structured such that detailed loss information is not readily available and can only be obtained after great effort and inconvenience to the user.

This situation has been one of the biggest barriers to application of loss analysis methods to engine performance analysis [1,2]. Past cycle codes were not readily extensible or flexible enough to easily accommodate *post-facto* incorporation of loss calculation routines. As a result, the cycle analyst is usually forced to do this type of analysis by hand in a spreadsheet environment for a limited set of engine operating conditions (if it is performed at all).

The advent of modern cycle analysis codes implemented in object-oriented languages has largely removed this barrier. Such cycle analysis codes offer the ability to seamlessly add objects and functions required for loss analysis without the need for major modification of the existing code base. The research effort described in this paper represents a first step towards the construction of a suite of general tools to facilitate loss analysis in aircraft engine cycle models.

This paper begins with a discussion on the fundamental concepts and definitions of loss and work potential and how they can be applied to the analysis of engine performance. Next, the implementation of the necessary thermodynamics functions in the Numerical Propulsion System Simulation (NPSS) is described [3]. These functions are then used as the basis for constructing a general loss analysis element in the NPSS-native programming language. This element is described in detail and it is applied to the analysis of a mixed flow turbofan engine model to illustrate results obtained with these tools.

ANALYSIS OF LOSS IN ENGINE CYCLE MODELS

The first law of thermodynamics is a statement of conservation of energy. The second law of thermodynamics states that the entropy of a closed system can only remain the same or increase. Typical textbook presentations usually express the second law in inequality form as: $S_1 \leq S_2$ where S_1 is the total system entropy at some time and S_2 is total system entropy at a later time. Another way of stating this same thing is: $S_1 + \Delta S = S_2$ where S_1 and S_2 are as before, and ΔS is the entropy generated in the system between states 1 and 2. It has been shown that the entropy generation in the system is related to the reduction in maximum work that can be done by the system on its environment: $W_{lost} = T_{ref}\Delta S$ where W_{lost} is the work potential that was lost due to the generation of entropy in the system and T_{ref} is the reference (dead state) temperature of the surrounding environment [4]. Thus, an alternate and perhaps more intuitive expression of the second law of thermodynamics is $W_1 - W_{lost} = W_2$ where W_1 is the maximum work that the system could theoretically do on its environment at state 1 and W_2 is the maximum work potentially available at state 2.

This simple concept of work potential is the crux of the loss analysis methods discussed in this paper [5]. The central concept is that a fluid at a given temperature, pressure, and composition contains a thermodynamically precise and quantifiable amount of work potential. It is the transfer and loss of this work potential that is the quantity of fundamental interest for prime-movers such as aircraft engines.

For example, the conditions at a given flow station in an engine can be precisely quantified using standard cycle analysis methods. If the fluid at this flow station is at a temperature and pressure other than the ambient temperature and pressure, we know that it is possible to extract work from the flow passing through this flow station. The second law of thermodynamics sets an upper bound on the work that can be extracted from this flow. If this information is calculated for every flow station in an engine model, it can be used to deduce the losses inside each engine component. This information is useful not only for understanding component performance, but also in understanding the performance of the engine as a whole.

To gain a better understanding of how the loss analysis works, consider a single component in isolation. One can conceptualize any arbitrary component as a “black box” that has one or more fluid streams entering and exiting Figure 1). In addition, there may be shaft work entering or exiting the component, and possibly heat transfer through the component surfaces (we will neglect heat transfer in this discussion). In the NPSS environment, these streams are organized as four types of port objects: fuel ports, fluid ports, bleed ports, and shaft ports [6]. All energy and work transfer occurring into or out of a component must pass through one of these ports. Since NPSS uses 1-D flow station representations of the engine system, it is assumed that each port can be represented by its average station properties. For airbreathing propulsion applications, the thermodynamic state of the fluid is specified by total temperature (Tt), total pressure (Pt), and fuel-air ratio (FAR).* This, in addition to total mass flow rate (W), is sufficient to describe total energy and work potential flux passing through a fluid or bleed port. In the case of a shaft, the energy and work transfer are equal and are given by the shaft power. Let us ignore work potential flux in fuel ports for the present time. The total loss inside this component can be calculated by merely summing the net input and output streams of work potential. This procedure is repeated for every model element to yield total losses throughout the engine.

The loss analysis method described above was implemented by first extending the FlowStation object to include new member functions for calculating work potential, then implementing the port summing function in a separate analysis element. Since the NPSS FlowStation object is part of the code base, implementation of the desired functions required consent from the NPSS development community. These functions are currently available in the production NPSS system (version 1.6.1G and later).

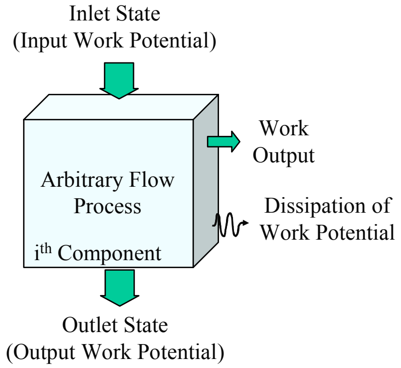

Figure 1: "Black Box" Representation of Component Flow Processes.

NEW FLOWSTATION MEMBER FUNCTIONS

In implementing new FlowStation member functions, a number of conditions had to be satisfied in order for the proposed modifications to be deemed acceptable for inclusion in a production NPSS release. Among these, the work potential functions can not impact or in any way alter existing flow station calculations; the source code modifications should be minimal and should also be transparent to the user; the number of functions to be implemented should be the minimum necessary and those implemented should be simple to use; the calculations should have negligible impact on calculation speed; and the functions should not contain any internal convergence loops or add any new failure modes to the FlowStation object. The present NPSS implementation of work potential calculations satisfies all these conditions.

As this project is the first time this type of loss analysis functionality has been implemented in a production cycle analysis code, only the simplest and most useful work potential functions were implemented. These are manifested as three new FlowStation member functions: `getExergy`, `getIdealWork`, and `getIdealEnthalpy`. The first two are used principally for loss analysis while the third is useful for efficiency calculations in various compression and expansion elements.

getExergy Flow Station Function

The getExergy FlowStation member function has the form:

```
real getExergy (real Tref, real Pref);
```

where Tref and Pref are the temperature and pressure used as the dead state point of reference for work potential calculations. Tref and Pref are typically taken to be ambient temperature and pressure for most analysis purposes. `getExergy` returns the mass-specific exergy content of the mass flux passing through that flow station. It is evaluated in NPSS as:

$$\begin{aligned} exergy = {} & h_t(T_t, P_t, FAR) - h_t(T_{t,ref}, P_{t,ref}, FAR) - \\ & T_{ref}[s(T_t, P, FAR) - s(T_{t,ref}, P_{t,ref}, FAR)] \end{aligned} \qquad (1)$$

where subscript ‘t’ denotes stagnation conditions, ‘h’ is mass-specific stagnation enthalpy, ‘s’ is mass-specific entropy, T_t, P_t, FAR define the state at the flow station of interest, and the reference state is given by $T_{t,ref}$, $P_{t,ref}$, and FAR.

---

* More precisely, thermodynamic state is specified by any two intensive thermodynamic parameters, relative velocity, and FAR (which determines composition of the fluid).

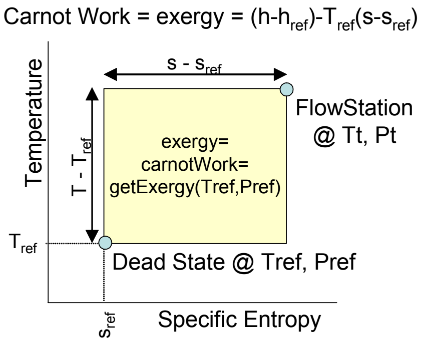

Figure 2: Geometric Interpretation of Exergy Expressed in a Temperature-Entropy Diagram.

Exergy physically corresponds to the work that would be obtained if one were to put a Carnot device between the FlowStation conditions and an infinite reservoir at Tref, Pref conditions (Figure 2). The value returned from `getExergy` is therefore equal to the mass specific work that would be produced by such a Carnot device operating between these two conditions. It represents an upper limit on the work that can be extracted from a flow when taking it from Tt, Pt into equilibrium with its environment at Tref, Pref.

It should be noted that the most general definition of exergy includes many components of work potential beyond temperature and pressure equilibrium with the environment. For example, chemical equilibrium, species concentration, voltage potential, gravitational potential, electromagnetic radiation, nuclear, and many other components all contribute to the total exergy of a substance. However, the vast majority of these additional exergy components are of little interest for cycle analysis applications and are ignored in the `getExergy` function calculations. Furthermore, `getExergy` assumes a constant gas composition (i.e. FAR) and therefore does not include exergy available due to differences in species partial pressures between flow station and reference conditions. Finally, `getExergy` calculations use the flow station total temperature and pressure, thereby implicitly including gas kinetic energy as part of the exergy calculation. Appendix A provides further discussion on these points.

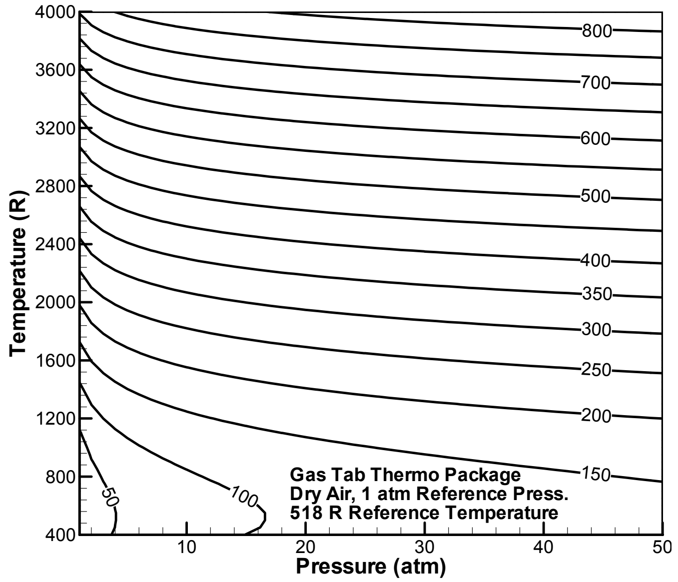

Figure 3: Contours of Flow-Specific Exergy as a Function of Temperature and Pressure (BTU/lbm).

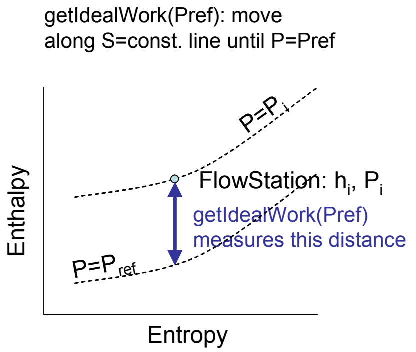

Figure 4: Definition of the `getIdealWork` Function Depicted on a Mollier Diagram.

Figure 3 is an exergy contour plot showing mass-specific exergy of dry air as a function of total pressure and temperature assuming a fuel-air ratio of 0.0 and reference temperature/pressure of 518R/14.7 psia, respectively. This plot was created by evaluating the `getExergy` function over a matrix of temperatures and pressures (at constant reference conditions). The range of temperatures and pressures shown on this plot are intended to be representative of those encountered in modern aircraft gas turbine engines. The exergy content of air at the temperatures and pressures typical of the conditions at station 4 is quite appreciable. It is, in fact, significantly higher than the amount of work realized from typical turbine expansion processes used today, suggesting that there is far more exergy work potential available in turbine inlet flows than is actually extracted in practice.

The source of this apparent discrepancy is the assumption of equilibrium temperature and pressure enforced by the definition of exergy. While it is relatively common to expand a high enthalpy flow to ambient pressure in an aircraft gas turbine, the attainment of equilibrium temperature implicitly requires the inclusion of some means for exhaust flow heat transfer in order to reach the reference temperature. The weight

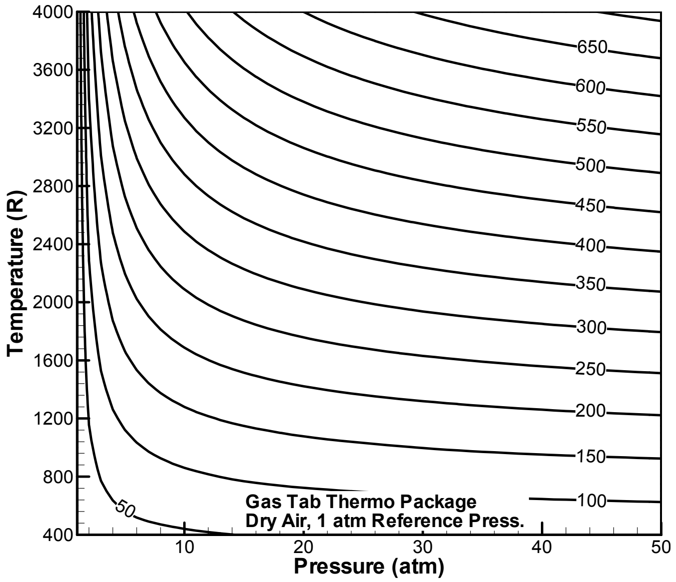

Figure 5: Contours of Gas Specific Power as a Function of Temperature and Pressure (BTU/lbm).

and volume constraints placed on aircraft engines typically make it impractical to recover waste heat to improve cycle efficiency. This is a serious limitation on the usefulness of exergy as a measure of *useable* flow work potential in aircraft engines and is the reason for the inclusion of the `getIdealWork` function as an alternative method for evaluating flow work potential.

getIdealWork Flow Station Function

The `getIdealWork` FlowStation member function has the form:

```
real getIdealWork (real Pref);
```

where Pref is the pressure used as the dead state point of reference for work potential calculations. Pref is typically taken to be ambient pressure for most analysis purposes. `getIdealWork` returns the mass-specific ideal compression or expansion work required to get from the current flow station pressure to the user-specified pressure, Pref, while moving along a *constant entropy line* (Figure 4). In other words, `getIdealWork` returns the mass specific work that would be produced if the flow at the flow station conditions were passed through an ideal turbine (or compressor) to expand (compress) it from Pt to Pref. The exhaust temperature corresponding to this imaginary expansion process is a fall-out from the calculations. The calculation is implemented in NPSS as:

$$idealWork = h_t(P_t, T_t, FAR) - h_t(P_{t,ref}, s = const., FAR) \quad (2)$$

where 's=const' implies a constant entropy process. Thus, `getIdealWork` can be viewed as a specialized form of the `getExergy` function wherein only pressure equilibrium is enforced when calculating the work potential.

The concept of an imaginary expansion or compression as a measure of work potential is not new and has been used for many years, though in limited ways. For example, the power output of core engines is typically quoted in terms of "gas horsepower," which is the same as `getIdealWork`. Other names for this quantity are "gas specific power," and "available energy."

Results from the `getIdealWork` function are shown in Figure 5. This is a contour plot of gas specific power as a function of temperature and pressure using air as the working fluid and assuming the reference pressure is sea-level ambient. The plot was generated by evaluating the getIdealWork function over a matrix of temperature/pressure combinations. Note that the magnitude of work potential is relatively close to that actually extracted using modern turbomachinery and that the shape of the contours is different from that of the exergy plot.

getIdealEnthalpy Flow Station Function

The `getIdealEnthalpy` FlowStation member function has the form:

```
real getIdealEnthalpy (real Sref);
```

where Sref is a reference entropy level. The `getIdealEnthaply` function is similar to the getIdealWork function except that instead of moving along a constant-entropy line to reach Pref, `getIdealEnthaply` moves along a constant pressure line (P=Pt) to reach Sref. In other words, `getIdealEnthalpy` returns the mass-specific enthalpy that would have resulted from ideal compression (expansion) along the S=Sref line until reaching the current flow station pressure—which is the intersection of the P=Pt and S=Sref lines. The definition of getIdealEnthalpy is illustrated in Figure 6.

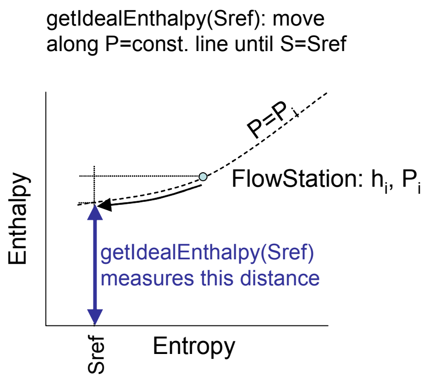

Figure 6: Definition of the getIdealEnthalpy Function Depicted on a Mollier Diagram.

The `getIdealEnthalpy` function is a useful utility for evaluating component efficiency. For example, compressor efficiency in the compressor element would be evaluated as:

```
eff=(Fl_O.getIdealEnthalpy(Fl_I.S)-
    Fl_I.ht)/(Fl_O.ht-Fl_I.ht)
```

This function eliminates the need to instantiate auxiliary FlowStation objects in the compressor element (or map subelement) to do calculations for "ideal" compression conditions.[†] Finally, take note of the difference between the `getIdealWork` and `getIdealEnthalpy` functions: `getIdealWork` returns the vertical distance between two constant pressure lines on an H-S diagram while `getIdealEnthalpy` returns the distance between the H=0 axis and a constant pressure line. One is a delta-enthalpy while the other is an absolute enthalpy.

MODEL LOSS ANALYSIS ELEMENT

The implementation of the `getIdealWork` and `getExergy` functions in the FlowStation object greatly simplifies the calculation of thermodynamic losses in engine components. As mentioned previously, the components can be viewed as "black boxes" and the loss in the component calculated based solely on fluid and shaft port conditions entering and leaving the component. Returning to the earlier compressor example, the calculation of loss in the compressor element (assuming no bleed) would be given by:

```
compressorLoss=(Fl_O.ht-Fl_I.ht)-
                Fl_O.getIdealWork(Fl_I.Pt);
```

This basic calculation procedure is easily generalized and automated such that every element in an engine model can be analyzed quickly and easily. The logical approach for doing this is to create a special loss analysis element that can be appended to the model in much the same way as the "`EngPerf`" (engine performance summary) element is currently used. This was the genesis for the development of the `NetFlux` analysis element.

The objective of `NetFlux` is to give NPSS users the capability to accurately analyze losses in any NPSS model with minimal effort. The only effort required to get loss analysis results is the instantiation of the `NetFlux` element and the

---

[†] Incidentally, the same calculation can be done using getIdealWork:
`eff=Fl_O.getIdealWork(Fl_I.Pt)/(Fl_O.ht-Fl_I.ht)`

definition of the thermo quantity to be conserved. `NetFlux` can also be used for checking conservation of mass, momentum, or energy in a model and readily reveals any areas where the model is failing to satisfy conservation equations.

There are three main parts to the `NetFlux` element. The first is the `PortList` object, which is a generic data structure to contain all input and output flux quantities through all element ports. `NetFlux` creates one `PortList` object for each element in the model. These objects are instantiated inside `NetFlux` and are named according to the name of the top-level element they represent.

There are two member functions in `NetFlux`: `verify()` and `calculate()`. The verify function instantiates all `PortList` objects, one for each element in the model. Because the `PortLists` are instantiated in the verify function, `NetFlux` should be instantiated only after all elements containing ports are instantiated. The calculate function simply loops through all ports in each element and sums the total flux entering and exiting each type of port (fluid, fuel, and shaft). The `NetFlux` element currently supports all types of ports except thermal port. It then stores these values in the proper `PortList` object and sums the total flux in all ports to obtain net generation/destruction of the flux quantity.

While `NetFlux` was created to analyze model losses, it is also a general utility element for checking conservation of any thermo property in an engine model. For example, the default settings for the `NetFlux` element will cause it to sum the total mass flow into and out of each element fluid/bleed port. It will then calculate the net mass flux generated or destroyed in each model element, which presumably should be zero for a steady-state model. In the case of transient models, this element calculates the rate of change of fluid storage in all elements at each time step. One can check conservation of energy in an engine model by simply specifying total enthalpy as the input for the `NetFlux` element. It will automatically multiply by total flow rate, account for shaft ports, etc. in order to yield calculations for net energy input/output from each element.

One `NetFlux` element must be instantiated for each thermo quantity to be conserved in the model. In general `NetFlux` can track any thermo property that can be accessed using a FlowStation member function. Most will produce nonsensical results, with `getht()`, `getimp()`, `getIdealWork()`, and `getExergy()` being the main FlowStation functions of interest.

MIXED FLOW TURBOFAN ANALYSIS EXAMPLE

A generic mixed flow turbofan (MFTF) is used as a case study to demonstrate typical results available from the NPSS loss the analysis functions and elements described herein. This model is implemented in the NPSS-native programming language and is representative of current low bypass MFTF engines in terms of model fidelity and engine performance levels. A schematic diagram of this model is shown in Figure 7. The element names are denoted in capital letters while the connecting flow stations are denoted with numbers. Figure 7 also shows that this model features two spools, an afterburner, and several cooling flow circuits.

This model was modified to perform loss calculations with the addition of a few lines of NPSS code:

```
Element NetFlux MASS {
   fluxType="";
   switchMassSpecific="True";
   switchActive="On";
}
solverSequence.remove("MASS");
postsolverSequence.append("MASS");
```

This code acts to instantiate a new element of type `NetFlux` called “MASS” which is used to check model conservation of mass (the default is `fluxType=“″`, which causes the element to conserve mass). This MFTF example will demonstrate application of the NetFlux element to check conservation of mass and energy as well as calculate loss in exergy and ideal work. All cases are analyzed at sea level static conditions and intermediate rated power setting. Results can just as easily be obtained at any other flight condition or power setting.

Conservation of Mass

To start, let us use `NetFlux` to check conservation of mass in this model. Results from this analysis are shown in the first column of Table 1 while the corresponding FlowStation mass flows are shown in the first column of Table 2. This table

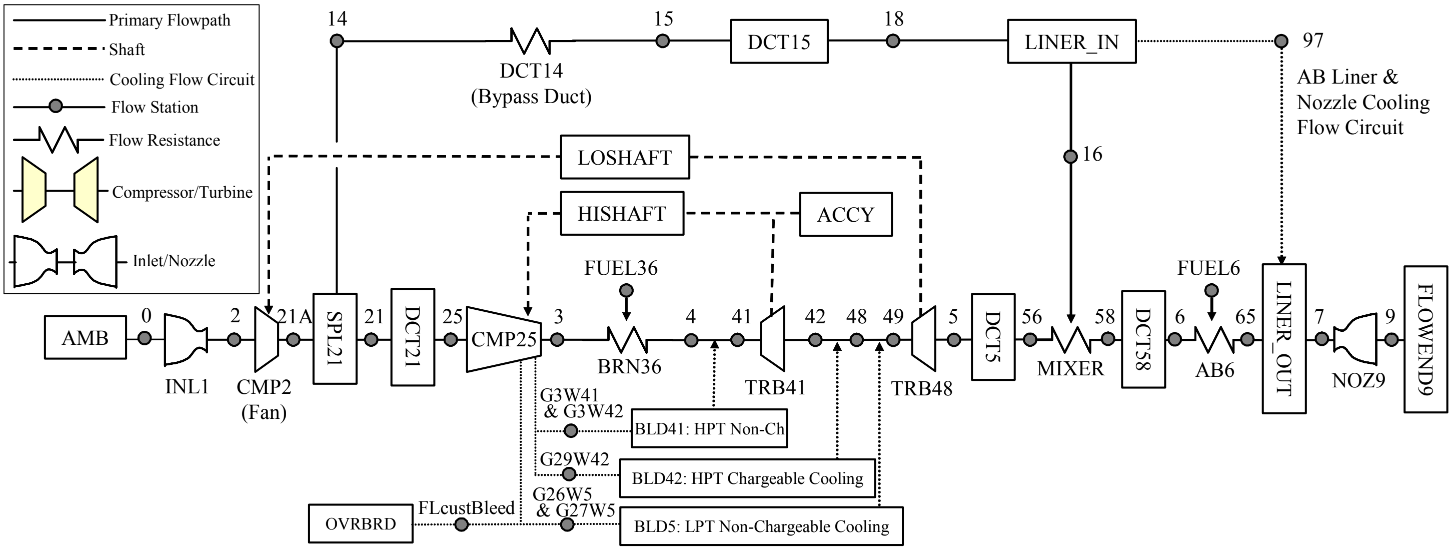

Figure 7: Schematic Diagram of Mixed Flow Turbofan Cycle Model.

shows the net flux for each MFTF model element, where net flux out of an element is defined as positive. The element names in the first column of Table 1 and the flow station numbers in the first column of Table 2 correspond to those defined in Figure 7. Note that the `FlowStart` element “AMB” has a net creation of mass flow while the `FlowEnd` has a net destruction. Furthermore, 3.5 lbm/s of fuel mass is injected in the combustor while 0.8 lbm/s flow is extracted for customer use. Summing the net mass flux in/out of each element, the total across all elements is shown to be zero implying that this model accurately conserves mass flow.

Conservation of Energy

The NetFlux element can be used to provide an independent check of conservation of energy by setting the `fluxType` variable to sum net enthalpy into/out of each element:

`fluxType=”ht”;`

The analysis results are shown in the second column of Table 1, where the components named on each row correspond to the model components shown in Figure 7. Note that the net energy flux through the model is not zero due to the way the engine is modeled. First, the model has a net energy output of 15.1 BTU/s into the accessories section. From the model’s point of view, the accessories appear as an energy sink. Likewise, the high and low spool windage losses appear as a net destruction of energy in the model. This energy is implicitly removed by the oil system, the details of which are not modeled here. The 278.8 BTU/s used for customer bleed appears as a net energy sink for the same reasons. Note also the apparent large loss in energy in the burner BRN36. This is due to the way incomplete combustion is handled in the burner element calculations—a portion of the fuel’s lower heating value (LHV) disappears due to incomplete combustion. As one might expect, the AMB and FLOWEND9 elements have large net creation and destruction of energy, respectively. The `FuelStart` element FUEL36 also has a large positive energy flux due to the LHV of the fuel flow it contains. Summing all energy sources and sinks in the model shows that there is 21.5 BTU/s more energy leaving the model than entering. This error is small relative to the total energy flux in the model and is due to residual errors remaining after the cycle balance calculations. This error can be driven closer to zero by simply tightening the solver tolerances.

Gas Specific Power Losses

NetFlux can be used to calculate available energy (gas horsepower) losses throughout the model by setting the `fluxType` variable to:

`fluxType=”getIdealWork(AMB.Ps)”;`

The results of this analysis are shown in the third column of Table 1. For example, these results show that the pressure drop due to the afterburner flameholders causes a loss of 478.6 BTU/s (677 HP), an appreciable penalty to overall performance. The accessories yield a loss of 15.1 BTU/s, as before. Although there is a large energy flux in the AMB element, there is no appreciable work loss.

Closer scrutiny of the ideal work losses listed in Table 1 reveals some surprising and counterintuitive results. First, the BRN36 element appears to have a large loss of gas specific power. This is a manifestation of the bookkeeping system

Table 1: Net Creation/Destruction of Mass, Energy, Ideal Work, and Exergy in Each Model Element.

	Component	Mass	Energy	IdealWork	Exergy
		lbm/s	BTU/s	BTU/s	BTU/s
Primary Flowpath	AMB	245.2	30450.9	-0.2	0.0
	INL1	0.0	0.0	0.0	0.0
	CMP2	0.0	0.0	-1219.4	-1207.4
	SPL21	0.0	0.0	0.0	0.0
	DCT21	0.0	0.0	-60.6	-58.6
	CMP25	0.0	0.0	-2183.2	-1977.0
	FUEL36	3.5	65068.8	65068.8	65068.8
	BRN36	0.0	-780.8	-29500.0	-16500.0
	BLD41	0.0	0.0	277.8	-641.7
	TRB41	0.0	0.0	-2893.9	-1099.3
	BLD42	0.0	0.0	-916.1	-800.2
	BLD5	0.0	0.0	-503.4	-376.8
	TRB48	0.0	0.0	-1196.8	-450.6
	DCT5	0.0	0.0	-481.4	-180.6
	MIXER	0.0	0.0	233.9	-2332.0
	DCT58	0.0	0.0	-494.4	-224.8
	FUEL6	0.0	0.0	0.0	0.0
	AB6	0.0	0.0	-478.6	-216.0
	LINER_OUT	0.0	0.0	3.9	-219.1
	NOZ9	0.0	0.0	0.0	0.0
	FLOWEND9	-247.9	-94400.0	-25300.0	-38400.0
Bypass Flowpath	DCT14	0.0	0.0	-60.5	-58.4
	DCT15	0.0	0.0	-60.9	-58.4
	LINER_IN	0.0	0.0	0.0	0.0
	ACCY	0.0	-15.1	-15.1	-15.1
	HISHAFT	0.0	-60.1	-60.1	-60.1
	LOSHAFT	0.0	-6.4	-6.4	-6.4
	OVRBRD	-0.8	-278.8	-153.8	-156.1
	TOTAL	0.0	-21.5	-0.2	30.2

Table 2: Flow Station Conserved Quantities for SLS MFTF Example Case.

	Flow Station	Mass	Energy	IdealWork	Exergy
		lbm/s	BTU/pps	BTU/pps	BTU/pps
Primary Flowpath	Station 0	245	124	-6	-6
	Station 2	245	124	-6	-6
	Station 21A	245	183	54	54
	Station 21	163	183	54	54
	Station 25	163	183	53	53
	Station 3	122	374	232	233
	Station 4	125	875	509	613
	Station 41	146	802	470	553
	Station 42	160	573	233	321
	Station 42	146	591	239	335
	Station 49	166	476	133	223
	Station 5	160	482	135	228
	Station 56	166	476	130	222
	Station 58	239	387	107	161
	Station 6	239	387	105	160
	Station 65	239	387	103	159
	Station 7	247	380	102	155
	Station 9	247	380	102	155
Bypass Flowpath	Station 14	81	183	54	54
	Station 15	81	183	53	53
	Station 16	73	183	52	52
	Station 18	81	183	52	52
Cooling Flow Ckts	G26W5	1	330	174	178
	G27W5	4	330	174	178
	G29W42	1	374	232	233
	G3W41	21	374	232	233
	G3W42	11	374	232	233
	Station 97	8	183	52	52
	FlcustBleed	8	340	187	190

chosen for the `NetFlux` element. The present implementation assumes that the entire lower heating value of incoming fuel in fuel ports is available as ideal work (and exergy). Of course, only a fraction of the fuel's heating value is realized as available energy after combustion. As a result, the combustor's fuel input port appears to have a large ideal work input and a much smaller ideal work output at the outlet with a net loss to match. The available energy input due to combustion is confounded with available energy loss due to combustor pressure drop and incomplete combustion. The analysis methods described in Ref. 2 are required if one desires to estimate these individual loss contributions.

A second counterintuitive result is the apparent generation of available energy in the bleed element BLD41 and in the MIXER element. The generation of available energy in a component appears at first glance as though it should violate one or more laws of thermodynamics. In fact, it does not. Rather, it is a manifestation of the definition of available energy. It can be shown that ideal work will, in general, be created whenever a high and a low temperature stream are mixed. This stands in contrast to exergy where the mixing of hot and cold streams always results in destruction of exergy.

The explanation for this apparent generation of available energy lies in the fact that the available energy of the "warm" mixed stream is higher than the average of the "hot" and "cold" unmixed streams. This is depicted in the Mollier diagram of Figure 8. This figure shows two constant pressure lines, one at ambient pressure and the other at a higher pressure. The vertical distance between the contours is the available energy. By conservation of energy, the enthalpy of the mixed stream (assuming equal mass flow rates and pressures) is simply the average of the enthalpies of the hot and cold streams. This corresponds to an entropy slightly higher than the mass average entropy of the incoming streams. As a result the 'Δh' of the mixed flow is greater than the average of the Δh of the two separate streams, resulting in an apparent increase in available energy. The vertical Δh distances are compared at left in the figure—note that $2\Delta h_{mix}>\Delta h_{hot}+\Delta h_{cold}$. Thus, the generation of available energy is purely a thermodynamic gas property phenomena and occurs because the lines of constant pressure on a Mollier diagram both curve and diverge. Proper calculation of loss in the mixer and bleed elements requires an additional correction to account for this phenomena.

A final curious result shown in Table 1 is the nozzle loss (or lack thereof). The calculated nozzle loss is zero for both the ideal work and exergy calculations. In fact, nozzle losses are modeled using a nozzle thrust coefficient, so it was puzzling that the NetFlux element returned zero loss for this element. Upon further inspection, it was found that the nozzle element correctly applies a thrust decrement due to thrust coefficient, but this is never translated into a reduced stagnation pressure in the nozzle exit flow station. This is a flaw in the nozzle element calculations, revealed by analysis results from the NetFlux element. The NetFlux element is very useful for finding these types of errors in model calculations.

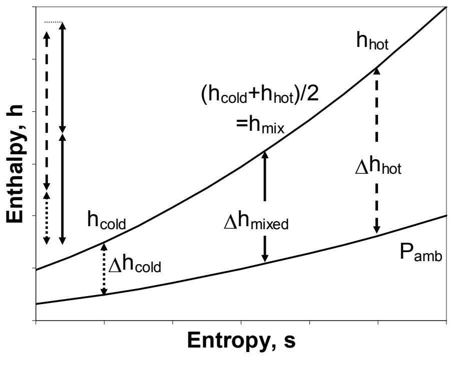

Figure 8: "Creation" of Available Energy Via Mixing of Cold and Hot Streams.

Exergy Losses

To calculate exergy losses in the model, the `fluxType` variable should be set to:
`fluxType=”getExergy(AMB.Ts,AMB.Ps)”;`
The exergy losses are shown in the last column of Table 1 with the corresponding flow station exergies shown in the last column of Table 2. The flow exergy always decreases in all components, including those components where two streams are mixed. Note also from Table 2 that the flow exergy is always higher than the corresponding ideal work (available energy). This is consistent with the intuitive expectation that ideal work should be less since it is defined as work available through pressure equilibrium only whereas exergy includes pressure and temperature equilibrium with the environment.

All shaft power losses are the same in all three cases of energy, ideal work, and exergy. Shaft work, flow velocity, and potential energy (height) are all forms of energy with perfect order. All of the energy stored in any of these forms is therefore available to do work. As a result, the definitions for exergy, ideal work, and energy all reduce to identical forms for these modes of work transfer.

It should be obvious from this example that the `fluxType` variable can be set to any character string which, when evaluated by FlowStation, will return a real value. For example, one could check model conservation of momentum by specifying the impulse function as the `fluxType` to be conserved.

FUTURE WORK

The proof-of-concept implementation described in this paper is the basic infrastructure needed for fast and accurate loss analysis of engine cycle models. Because it is implemented in an NPSS-native environment, it provides a good starting point for further developments needed to "productionize" the analysis tool. It is hoped that this tool or something similar can become a part of the standard NPSS distribution at some point in the future.

There are several areas still needing further development in order to make the tool truly "production ready." First, the current implementation ignores exergy contained in fuel ports. A more thorough handling of fuel exergy that accounts for fuel composition, temperature, and pressure is desirable. This is especially relevant to proper loss analysis of SCRAM and rocket propulsion systems (wherein large heat fluxes are absorbed by the fuel). Also, additional logic in the `NetFlux` element to handle thermal ports will be required for accurate analysis of these systems.

A more general treatment of ideal work for situations involving mixing of unlike streams is also needed. At present, the user must manually calculate theoretical "ideal work generation" due to mixing of streams. An automated means for doing this would be preferred. Finally, the use of `NetFlux` for checking conservation of momentum via the impulse function is not well-validated and needs further work.

CONCLUSIONS

Work potential analysis uses absolute loss as a measure of performance instead of examining and comparing relative efficiencies as is conventionally done. This allows developers to directly compare the losses within the components of a system against each other, as well as compare overall losses of any two systems regardless of their architecture. Additionally, it is simple to determine how much of the losses within components and systems are inherent to the thermodynamic cycle, and how much of that loss can eventually be eliminated through component improvements. Attention may then be focused *not* on areas of the propulsion system where the greatest losses exist, but on the areas where the greatest room for *reduction of loss* exists. This type of knowledge is essential to making well-informed tradeoffs between design alternatives.

This paper described and demonstrated a new general loss analysis tool that directly facilitates the above objectives. This tool consists of three new FlowStation member functions and a new element added to the NPSS system. This tool is trivially simple to use and can currently provide loss results measured relative to available energy or exergy ideal. The system is extensible so that additional figures of merit can be implemented with minimal difficulty. Finally, this analysis tool has the added side-benefit of providing a simple and easy means for summarizing and checking conservation of mass, momentum, and energy in any arbitrary cycle model.

ACKNOWLEDGMENTS

We would like to thank NASA Glenn Research Center for supporting elements of this work under contract NAS 3-03115. In addition, we would like to thank Mr. Tom Lavelle at NASA and Mr. Mike Lucas at GE Aircraft Engines for their assistance.

APPENDIX A—EXERGY AND ITS RELATION TO DEFINITION OF DEAD STATE

There are some subtleties associated with the definition of the exergy dead state that one should be aware of before using it as an analysis tool. By definition, exergy is the total work that could be extracted from a substance in bringing it into total equilibrium with its environment. We have implicitly assumed that the environment is essentially infinite (or is large enough that it is not appreciably changed by extracting work from the substance). This general definition of exergy includes *all* possible means of work production including mechanical, temperature, chemical, electrical, magnetic, nuclear, etc. Most of these modes are of little interest for gas turbine applications, and are ignored. The ones of interest here are: kinetic energy, internal energy, and chemical composition.

The kinetic energy component of exergy is accounted for by doing all the work potential calculations using stagnation properties. The internal energy component of exergy is manifested as the work available by taking the substance to pressure and temperature equilibrium. One could imagine that this is achieved in a two-step process: first using an isentropic turbine to reach pressure equilibrium (this is what `getIdealWork` does) followed by a Carnot engine to reach temperature equilibrium.

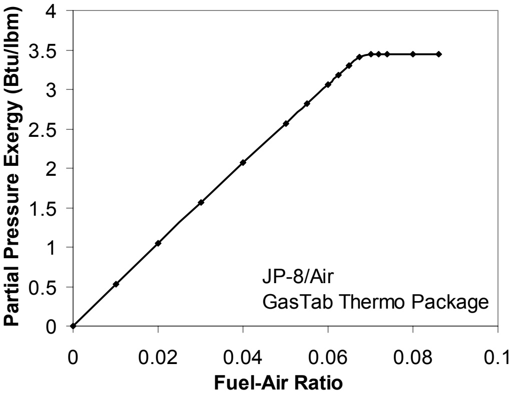

Figure 9: Difference in Calculated Exergy (Vitiated Versus Non-vitiated Dead State Assumptions).

Even after temperature and pressure equilibrium with the environment are reached, there is still a small component of exergy available. This is due to the fact that the chemical composition of the working substance will not, in general, be in equilibrium with its environment. Specifically, the environment is non-vitiated and therefore any substance with a fuel-air-ratio other than 0.0 will contain a component of exergy due to the differences in gas composition. Imagine a third step using a semi-permeable membrane to get additional PdV work by equilibrating partial pressures with the environment. Equilibrium is reached only when the partial pressures of all constituents are equal to the environment (equal partial pressures of CO_2, H_2O, etc.).

This third process is equivalent to defining the dead state as having FAR=0. Therefore, the chemical component of exergy is a function of FAR only, as shown in Figure 9. Note that its contribution to exergy is relatively small in comparison to the other two and is only a significant consideration at temperatures and pressures close to the dead state.

The exergy available by equilibration of chemical composition is of little interest in virtually all practical cycle analysis applications. Therefore, the `getExergy()` function implemented in NPSS defines the dead state *only in terms of Tt, and Pt* (leaving FAR to float at current conditions). This is equivalent to deleting the partial pressure component of exergy. One can easily include it *a posteriori* by applying a small delta as a function of FAR.

REFERENCES

[1] Roth, B.A., Mavris, D.N., "A Comparison of Thermodynamic Loss Models Suitable for Gas Turbine Propulsion," *J. of Propulsion and Power*, Vol. 17, No. 2, pp. 324-332, Mar-Apr 2001.

[2] Roth, B.A., "A Work Potential Perspective of Engine Component Performance," *J. of Propulsion and Power*, Vol. 18, No. 6, pp. 1183-1190, Nov-Dec 2002.

[3] Vivek Sanghi, B. K. Lakshmanan, V. Sundararajan, “Survey of Advancements in Jet-Engine Thermodynamic Simulation,” *J. of Propulsion and Power*, 2000 Vol.16 No.5 Pp. 797-807.

[4] Bejan, A., *Advanced Engineering Thermodynamics, 2nd Ed.*, Wiley, 1997.

[5] Roth, B.A., "The Role of Thermodynamic Work Potential in Aerospace Vehicle Design," Cleveland, 16th ISABE, Sept. 2003.

[6] Evans, A.L., et al., “Numerical Propulsion System Simulation's National Cycle Program,” AIAA Paper 98-3113, 1998.