A Process for the Creation of T-MATS Propulsion System Models from NPSS Data

Jeffryes W. Chapman[1]
Vantage Partners, LLC, Brook Park, Ohio 44142

Thomas M. Lavelle[2]
NASA Glenn Research Center Cleveland, Ohio 44135

Jonathan S. Litt[3] and Ten-Huei Guo[4]
NASA Glenn Research Center Cleveland, Ohio 44135

A modular thermodynamic simulation package called the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) has been developed for the creation of dynamic simulations. The T-MATS software is designed as a plug-in for Simulink® and allows a developer to create system simulations of thermodynamic plants (such as gas turbines) and controllers in a single tool. Creation of such simulations can be accomplished by matching data from actual systems, or by matching data from steady state models and inserting appropriate dynamics, such as the rotor and actuator dynamics for an aircraft engine. This paper summarizes the process for creating T-MATS turbo-machinery simulations using data and input files obtained from a steady state model created in the Numerical Propulsion System Simulation (NPSS). The NPSS is a thermodynamic simulation environment that is commonly used for steady state gas turbine performance analysis. Completion of all the steps involved in the process results in a good match between T-MATS and NPSS at several steady state operating points. Additionally, the T-MATS model extended to run dynamically provides the possibility of simulating and evaluating closed loop responses.

Nomenclature

Ath	= throat area (in^2)
Byp_Noz	= bypass nozzle
BPR	= Branch Pressure Ratio
C-MAPSS40k	= Commercial Modular Aero-Propulsion System Simulation 40,000 lbf thrust
C-MAPSS	= Commercial Modular Aero-Propulsion System Simulation
Core_Noz	= core nozzle
dP	= change in pressure or delta pressure (psia)
Eff	= efficiency
FAR	= fuel to air ratio
Fg	= gross thrust
HPC	= high pressure compressor
HPT	= high pressure turbine
ht	= enthalpy
LHV	= lower heating value
LPC	= low pressure compressor
LPT	= low pressure turbine

[1] Control Systems Engineer, 21000 Brookpark Rd., AIAA Member
[2] Engineer, 21000 Brookpark Rd., MS 86-1 AIAA Member
[3] Control Systems Engineer, 21000 Brookpark Rd., MS 77-1, AIAA Senior Member
[4] Control Systems Research Engineer, 21000 Brookpark Rd., MS – 77-1, AIAA Member

N	= shaft speed (rpm)
N1	= low pressure shaft speed (rpm)
N2	= high pressure shaft speed (rpm)
Nc	= corrected shaft speed (rpm)
NPSS	= Numerical Propulsion System Simulation
Pa	= ambient pressure (psia)
PI	= proportional-integral
PR	= pressure ratio
Pt	= total pressure (psia)
R-line	= uniquely defined line on the compressor map
s_Eff	= map scale factor for efficiency coming from a map
s_Nc	= map scale factor for corrected shaft speed going to a map
s_PR	= map scale factor for pressure ratio coming from a map
s_Wc	= map scale factor for corrected flow coming from a map
T-MATS	= Toolbox for the Modeling and Analysis of Thermodynamic Systems
T_std	= standard day temperature (518.67 degR)
Tt	= total temperature (degR)
W	= mass flow (pps)
Wb	= bleed mass flow (pps)
Wc	= corrected mass flow (pps)
Win	= component input mass flow (pps)

Subscripts

Des	= variable at the design point
Map	= map variables are inputs or outputs of an empirical map model
Perf	= performance variables are final modeled parameter

I. Introduction

In gas turbine control system design, a model is often used in place of hardware to simulate the system and verify controls requirements are met. This model can be obtained in many different ways. One such way is by developing the simulation utilizing the Numerical Propulsion System Simulation (NPSS), a widely used engineering modeling environment for the design and analysis of aircraft engines.[1] Using NPSS models can be a convenient approach because they are often created in the early stages of gas turbine development and they can be quite accurate, and are easily modifiable to incorporate additional turbine design information.[2] NPSS is a powerful gas turbine cycle modeling and analysis package; however, integrating NPSS models with control system models can be difficult because of the complexity of NPSS, which may not be compatible with the legacy codes and the simulation platform used in the development of the control.[3] Another approach is to use a model like C-MAPSS or C-MAPSS40k. Built in Simulink®, C-MAPSS and C-MAPSS40k are generic commercial engine simulations with industry standard control architecture.[4,5] Controls design using a pre-existing generic engine with a basic controller that requires no cross platform integration can drastically reduce simulation setup time and allow a controls designer to test many different ideas quickly. However, customization of the engine model from such a tool can be difficult.

To eliminate cross platform integration and allow for maximum flexibility, the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) can be used for gas turbine simulation development.[6,7] Written as a plug-in for MATLAB/Simulink, T-MATS contains a library of generic engine component models, a numerical solver, and controller blocks designed to aid in the creation of thermodynamic models and control systems. While T-MATS can be used to create custom models from engine data, legacy model data may also be leveraged for simulation generation. For example, the C-MAPSS simulation was re-created in T-MATS and results compared favorably within an acceptable error tolerance.[8]

This paper describes a process for developing nonlinear dynamic aircraft engine simulations with T-MATS using information from NPSS models. Specifically, a process for leveraging maps, constants, and steady state data from NPSS-developed models for the creation of MATLAB/Simulink simulations with T-MATS. This process is demonstrated on the JT9D, a publicly available NPSS model of a dual spool high-bypass turbofan engine.[9]

This process can be summarized by describing how engine component models are generated in T-MATS from NPSS. For components like the fan, compressors, and turbines, performance maps are directly utilized from NPSS.

For components that are not explicitly defined, like the inlet, NPSS is run at a steady state point and then the inputs are perturbed to form a partial relation between the input and output flow variables. Nozzle performance is based on thermodynamic equations that determine static and total conditions at the throat and exit utilizing input parameters taken from NPSS data. For the burner, the fuel thermodynamic properties are included in the model, as in NPSS. The rotor dynamics are added to the T-MATS model using a representative rotational moment of inertia and the associated equations of motion, as the change in rotor speed is a function of torque differential between the turbine and compressor on a given rotor. Conservation of energy is ensured internally for each component, however mass is ensured by adding a solver that reconciles inter-component imbalances at every computation step. This paper will focus on the specifics of the process involved with NPSS, including extracting pertinent information necessary to develop the T-MATS model for the JT9D, and the model matching, which will be performed at takeoff and cruise to show match variation due to changes in operating point (certain fixed boundary conditions). Scale factors that adjust the performance of the compressor and turbine are determined by two methods: a data derived method that back calculates the scale factors from NPSS, and the T-MATS iDesign tool, which generates the scale factors based on an operating point, cruise in this paper.

The paper is organized as follows. Section II offers a background on creating turbo-machinery simulations in T-MATS. Section III gives an overview of the procedure steps involved in this model development process. Section IV presents how the T-MATS simulation components and system architecture were selected and modified to match the NPSS model data. All model testing and results are detailed in section V. Section VI describes how the simulation was converted from steady state to dynamic. Conclusions are discussed in section VII.

II. T-MATS background

The T-MATS software is a MATLAB plug-in that introduces additional blocks and tools to the Simulink environment. The turbo-machinery block set, located within T-MATS, provides access to parameterized gas turbine modeling components such as turbines, burners, compressors, and nozzles that may be used to create simulations. Turbo-machinery components are assumed to be 0-D, where the component-level dynamics are not modeled. Internal component calculations contain a combination of physics based and empirical models. Algebraic relations and component maps are used to generate component or system steady state solutions at specific operating points. For example, compressor efficiency (Eff), corrected mass flow (Wc), and pressure ratio (PR) are determined from the compressor map as functions of corrected shaft speed (Nc) and R-line, where R-line is an uniquely defined line on the compressor map. Turbine Eff and Wc are determined from the turbine map as functions of Nc and PR, with input cooling flow being added either at the input or output of the component. Empirical calculations for enthalpy and entropy are functions of temperature and pressure and are based on the assumption that all component mass flows consist of two compounds, air and Jet-A aircraft fuel. Physics based equations for turbo-machinery components in T-MATS are predominately used to calculate component energy and mass requirements, such as nozzle flow error.

As stated above, a T-MATS model ensures the conservation of energy and mass. Conservation of energy is ensured internally to each component with energy balance equations. Although T-MATS ensures conservation of mass internally for some components, conservation across the system must also be upheld by using one of the T-MATS numerical solvers. Generally, this is done by monitoring conservation errors, or dependents, then using a T-MATS solver to adjust floating model variables, or independents, to eliminate the errors. For example, the dependent variable for a T-MATS compressor component is mass flow error (the difference between input mass flow and an internally calculated mass flow), and the independent variable is R-line. A model will be considered converged once the solver has updated the independents so that all dependents are at or near zero.

While the T-MATS and NPSS environments are different, they operate in a very similar manner, in terms of simulating a given turbo-machinery design. Both are modular, use component maps and thermodynamic properties for calculations, and both require solving for system dependents by adjusting independents as discussed above. However, NPSS contains more flexibility in its temperature and pressure to enthalpy and entropy calculations, and allows for many different compound options. It should be noted that in this paper the JT9D engine was simulated in NPSS using air and Jet-A fuel, which are the same compounds as those used in T-MATS. Currently, work is being done to add alternate compound capability to T-MATS with the integration of the Cantera thermodynamic modeling package; however this is beyond the scope of this paper.[10]

III. Process Overview

This process assumes that a particular engine architecture has already been simulated in NPSS and that the cycle analysis for this design has been completed. Once the T-MATS user decides on the NPSS engine architecture to simulate and the respective simulation components, the following process may be used to create a dynamic T-MATS model from steady state NPSS data and constants.

1. Converting maps and constants from NPSS into a useable format, while also modifying T-MATS components to be compatible with these maps and constants. In some cases it is necessary to perturb certain inputs to gain an understanding of how a component should be modeled.
2. Completing a component level model match by taking each portion of the NPSS model and recreating it in T-MATS.
3. Combining the model components into a system model with constant independent variables at the operating point.
4. Creating a steady state model by integrating a solver into the system model.
5. Converting the steady state T-MATS model to a dynamic T-MATS model.

In the final two steps of the process, several operating points at different power levels and flight conditions are selected for comparison purposes. These additional tests are used to ascertain the operating point dependence of the model match. It should be noted that for this paper a “good” matching was arbitrarily chosen to be anything under a 1% deviation between models. In the following section, this procedure is demonstrated using the JT9D NPSS model.

IV. T-MATS simulation creation

Because NPSS and T-MATS are both modular it was determined that the T-MATS model would use the same components as the NPSS model. Figure 1 shows a block diagram detailing placement of components and subsequent connections, which is derived directly from NPSS module station numbers, shown in Appendix A.

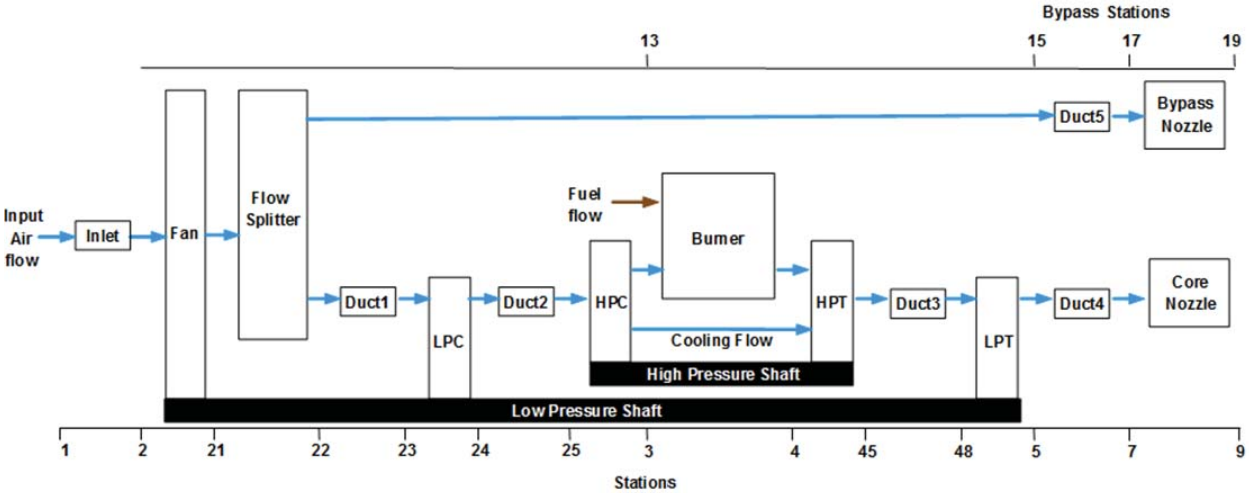

Figure 1. JT9D engine component setup.

As mentioned above, T-MATS and NPSS are both simulations that are developed around component maps. For example, both simulations define compressor maps that use R-lines and Nc to generate Eff, PR, and Wc. For this reason, component maps were copied as they were defined in NPSS and formatted to insure consistency in the variable names and table structures for use in T-MATS. These reformatted maps can be found in Appendix B.

Once the NPSS maps have been reformatted, it is necessary to integrate them with the T-MATS modeling structure using map scale factors. Map scale factors are multipliers that convert turbo-machinery map data to turbo-machinery performance data or vice versa and are typically used for shrinking or stretching a more generic model and/or converting Nc to a ratio value for map lookup. For this paper, map data are considered inputs or outputs of the component maps whereas performance (Perf) data are considered the actual values. For the creation of this T-MATS model, map scale factors were used to ensure a model match between NPSS and T-MATS. Shown in Table 1, for the arbitrarily chosen sea level static condition, are the map scale factors generated for the JT9D and the equations used to calculate them. Generating scale factors by deriving them from NPSS data essentially reformats the NPSS scale factors to be useable in T-MATS. It should be noted that there are differences between the data derived map scale factors in Table 1 and the map scale factors located within the NPSS data, as shown in Appendix A. These differences reflect variations in implementation between how T-MATS and NPSS use maps and scale factors.

Table 1. T-MATS data derived map scale factors, calculated from NPSS data.

Component	map scale factor	equation	Fan	LPC	HPC	HPT	LPT
Compressor	s_Nc	s_Nc = NcPerf / NcMap	3943.96	3662.66	6357.16		
Turbine	s_Nc	s_Nc = NcPerf × SQRT(T_std)/NcMap				34.87	18.47
Compressor or Turbine	s_Wc	s_Wc = WcPerf / WcMap	0.52	0.93	0.42	0.91	0.48
	s_PR	s_PR = (PRPerf-1) / (PRMap-1)	1.13	3.26	0.22	0.43	0.72
	s_Eff	s_Eff = EffPerf / EffMap	0.97	0.96	1.01	0.98	1.00

One such variation is in the turbine component equations, where NPSS and T-MATS calculate Nc differently. In T-MATS, Nc is referenced to a standard day temperature (T_std), while in NPSS this is not the case. This variation is taken into account in T-MATS by adjusting the turbine map scale factor for Nc (s_Nc) to include a T_std value, as shown in Table 1. It should be noted that the NPSS and T-MATS compressor components calculate Nc identically; therefore no additional multiplier was required for the compressor s_Nc definition.

Another implementation variation between the NPSS model and T-MATS has to do with the inclusion of bleed flow in the Wc component of a turbine map. For instance, in T-MATS there are three types of corrected flows used as input to the HPT turbine component: the HPT input mass flow, cooling flow to the input of the HPT, and cooling flow to the output of the HPT. In T-MATS, the summation of the HPT input mass flow and the cooling flow to the input of the HPT is compared to the internally calculated Wc to generate a mass flow error, which the solver utilizes to adjust the independent variables accordingly. In the NPSS data, this normalized flow error is based on component input Wc only. To resolve this variation, the T-MATS turbine block was modified to remove the HPT input cooling flow from the normalized flow error calculation. This adjustment moves the bookkeeping of the cooling flow from the turbine component to a component downstream, and must be done only because the T-MATS turbine map has cooling flow included in it and the NPSS turbine map does not.

As an alternative to using data to manually derive map scale factors for component maps, scale factors were also determined with the T-MATS tool iDesign. This tool is a time-saving feature that uses map scale factors and other key parameters to fit component maps to a design point automatically, effectively creating a model at a specific operating point. Map scale factors for the compressor and turbine may be generated by comparing the map values with performance values at a specified design point. Additionally, nozzle throat areas are generated by comparing nozzle mass flow at the design point with an internally calculated mass flow. The steady state design point selected for this study is detailed in Table 2, which in this case was chosen to be the cruise condition because a gas turbine is typically designed for this condition. Each step of the model creation process details the T-MATS model generated using data derived map scale factors as described earlier in this section. However, iDesign is also utilized during the final steps of the matching process as described in section V.

Table 2. iDesign inputs for map scale factor calculation.

Component	**NcDesMap**	**EffDes**	**PRDes**	**R-lineMap**
Fan	1.0	0.8674	1.644	2.37
LPC	0.993	0.8685	2.485	1.7716
HPC	0.996	0.8649	5.625	2.04
	NcDesMap	**EffDes**	**PRMap**	**Ndes, rpm**
HPT	99.0	0.9132	4.934	7392
LPT	105	0.9324	6.399	3667
	Wdes, pps			
Core_Noz	113.61			
Byp_Noz	562.59			
	Altitude, ft	**Mach Number**	**Temperature, degR**	
Flight Conditions	34000	0.8	448.43	

Additional constants required for T-MATS simulation creation are detailed in Appendix C and were determined from the NPSS data output located in Appendix A. The T-MATS components used for modeling were selected based on NPSS engine architecture, as described above, with modeling accuracy similar to that of the NPSS model. For example, the inlet total temperature does not change appreciably in the NPSS model, therefore the T-MATS model does not consider it.

V. Model Matching

Once the NPSS engine simulation is converted to T-MATS, model matching tests were performed that compared NPSS data with T-MATS generated outputs. This testing was broken down into several groups consisting of component level testing, system level testing without a solver, and system level testing with a solver. Component level tests used NPSS station data to determine how each NPSS module compares with its equivalent T-MATS component. The system level test without a solver brings the model together and validates the T-MATS component connections. For the final matching, the solver is enabled to verify the model will converge at the correct operating points.

A. Component Level Testing

A component level matching test was designed for each major part of the T-MATS JT9D high-bypass turbofan engine model, as detailed in Figure 1. Results from the component level testing using a takeoff operating point can be seen in Table 3. Stations immediately after each major engine component were selected for comparison purposes. Component data were gathered for the following parameters; flow (W), total temperature (Tt), total pressure (Pt), and gross thrust (Fg). If a parameter was not calculated at a particular station, either by T-MATS or NPSS, it was neglected, e.g., total pressure by the nozzle. In each component, the matching difference between the NPSS model and the T-MATS model was below 0.3%. The largest absolute difference was in the pressure output of the low pressure compressor (LPC). Thrust difference was negligible for both the core bypass and core nozzles. Engine flow difference was below 0.1% in all cases. These low error values are considered to demonstrate a sufficient component level match.

Table 3. Component level model match results.

Output	Parameter	NPSS	T-MATS	%Difference
Fan	W(pps)	1539.2	1539.21	0.0636%
	Tt(degR)	632.71	632.736	-0.0041%
	Pt(psia)	23.37	23.3669	0.0133%
Low Pressure Compressor (LPC)	W(pps)	245.29	245.2918	0.0743%
	Tt(degR)	821.38	822.4966	-0.1359%
	Pt(psia)	52.451	52.604	-0.2917%
High Pressure Compressor (HPC)	W(pps)	245.29	245.2904	0.0161%
	Tt(degR)	1398.6	1398.8	-0.0143%
	Pt(psia)	297.127	296.96	0.0562%
High Pressure Turbine (HPT)	W(pps)	228.18	228.1803	0.0127%
	Tt(degR)	2137.47	2136.97	0.0234%
	Pt(psia)	104.013	104.25	-0.2279%
Low Pressure Turbine (LPT)	W(pps)	250.25	250.25	-0.0009%
	Tt(degR)	1522.73	1522.82	-0.0059%
	Pt(psia)	22.575	22.577	-0.0089%
Nozzle core	W(pps)	250.25	250.251	0.0384%
	Fg(lbf)	11210	11210	0.0000%
Nozzle bypass	W(pps)	1293.91	1293.916	0.0495%
	Fg(lbf)	38674.8	38677.76	-0.0077%

B. System Level Matching (without solver)

Once component models are developed, they are assembled appropriately based on the engine architecture, shown in Figure 1, to build the engine plant model. Input flow characteristics and independents (e.g., R-line) are set to constants, taken from NPSS data and consistent with the takeoff operating point, for model match testing. These inputs consist of the values shown in Table 4.

Table 4. T-MATS JT9D simulation inputs for system level testing without a solver.

Plant Inputs	W	Pt	Tt	Fan R-line	LPC R-line	HPC R-line	BPR	HPT PR	LPT PR	N1	N2
Value	1539.2	14.69	545.67	2.0	1.769	2.081	5.275	2.694	4.558	3750	8000

Table 5 shows the results from the system level test with constant inputs. It can be seen that the average matching differences increased significantly from those of the component level test. However, all differences in this test still remain well under 1%, with the highest difference being in nozzle W. Matching differences of such a low magnitude constitute a good match and it also shows that the model components have been connected correctly. Note in this matching step, that independents are held constant and the solver that insures satisfaction of conservation properties is not activated. Flow difference values shown in the model matching results are internal component differences and should be considered on a component by component basis.

Table 5. Model matching results system level testing without a solver.

Output	Parameter	NPSS	T-MATS	%Difference
Fan	W(pps)	1539.20	1542.52	0.2158%
	Tt(degR)	632.71	632.30	-0.0648%
	Pt(psia)	23.37	23.32	-0.2139%
LPC	W(pps)	245.29	245.82	0.2150%
	Tt(degR)	821.38	822.10	0.0877%
	Pt(psia)	52.45	52.52	0.1316%
HPC	W(pps)	245.29	245.52	0.0938%
	Tt(degR)	1398.60	1399.00	0.0286%
	Pt(psia)	297.13	296.60	-0.1774%
HPT	W(pps)	250.25	250.67	0.1680%
	Tt(degR)	2137.47	2137.00	-0.0220%
	Pt(psia)	104.01	103.80	-0.2048%
LPT	W(pps)	250.25	250.67	0.1682%
	Tt(degR)	1522.73	1522.00	-0.0479%
	Pt(psia)	22.58	22.54	-0.1550%
Noz_core	W(pps)	250.25	250.92	0.2689%
	Fg(lbf)	11210.00	11190.00	-0.1784%
Noz_byp	W(pps)	1293.91	1297.62	0.2869%
	Fg(lbf)	38674.80	38585.00	-0.2322%

C. System Level Matching (with solver)

Once the plant model is verified to have a good match, a steady state simulation is created by adding the T-MATS solver and connecting the system independents and dependents.[6] In this example, the steady state independents are system input flow (W), Fan R-line, splitter bypass ratio, LPC R-line, HPC R-line, HPT PR, LPT PR, N1, and N2. The steady state dependents are normalized flow differences for the Fan, LPC, HPC, HPT, LPT, core nozzle, and bypass nozzle, along with high and low pressure shaft acceleration. During simulation, the solver will drive the dependents to zero by adjusting the independents, effectively moving the simulation towards a new operating point. This simulation was performed at takeoff conditions with altitude, Mach number, and temperature, set to 0 ft, 0, and 545.67 degR, respectively.

Table 6 shows the results from the system level test after allowing the solver to converge with the scale factors developed from NPSS data as well as with scale factors created with iDesign. It can be seen that the matching differences from the model using NPSS-derived scale factors increase slightly from that of the system level test that contained constant independent variables. These differences occur due to the operating point moving on the map, caused by the manipulation of independents by the solver, to meet conservation of mass. Flow differences listed in this section take into account that each component input flow is matched with its component calculated flow with a difference less than 0.0001%. Although there was an increase in the difference between the models, all errors remain less than 1%, with the highest difference being the LPC pressure (Pt). The matching differences from the T-MATS model that utilized scale factors from the iDesign tool are generally greater in magnitude. This increase can be attributed to the iDesign scale factors being generated for a cruise operating point rather than the takeoff condition being shown below.

Table 6. Model matching results for system level testing with solver.

Output	Parameter	NPSS	T-MATS, NPSS scale factors	%Difference	T-MATS, iDesign	%Difference
Fan	W(pps)	1539.20	1539.11	-0.0058%	1540.00	0.0520%
	Tt(degR)	632.71	632.78	0.0111%	632.90	0.0300%
	Pt(psia)	23.37	23.37	0.0000%	23.39	0.0685%
LPC	W(pps)	245.29	245.90	0.2487%	246.10	0.3302%
	Tt(degR)	821.38	823.00	0.1972%	823.30	0.2338%
	Pt(psia)	52.45	52.70	0.4747%	52.80	0.6654%
HPC	W(pps)	245.29	245.93	0.2593%	246.04	0.3066%
	Tt(degR)	1398.60	1401.00	0.1716%	1400.00	0.1001%
	Pt(psia)	297.13	298.00	0.2938%	298.20	0.3611%
HPT	W(pps)	250.25	250.90	0.2597%	251.00	0.2997%
	Tt(degR)	2137.47	2138.00	0.0248%	2139.00	0.0716%
	Pt(psia)	104.01	104.30	0.2759%	104.60	0.5644%
LPT	W(pps)	250.25	250.90	0.2597%	251.00	0.2997%
	Tt(degR)	1522.73	1523.00	0.0177%	1522.00	-0.0479%
	Pt(psia)	22.58	22.62	0.1993%	22.63	0.2436%
Noz_core	W(pps)	250.25	250.90	0.2597%	251.00	0.2997%
	Fg(lbf)	11210.00	11269.00	0.5263%	11280.00	0.6244%
Noz_byp	W(pps)	1293.91	1293.00	-0.0703%	1294.00	0.0070%
	Fg(lbf)	38674.80	38660.50	-0.0370%	38723.00	0.1246%

D. Model Match Testing Summary

Up to this point, the majority of testing has been completed at the takeoff operating point with an engine model created by utilizing component maps and relationships developed from NPSS data or from the T-MATS iDesign tool. For comparison purposes, additional testing at a cruise operating point was also conducted. Flight conditions and fuel inputs for the takeoff and cruise operating points are listed in Table 7.

Table 7. T-MATS JT9D simulation inputs for system level testing.

operating point	altitude, ft	Mach Number	Temperature, degR	fuel flow, pps
design point or cruise	34000.0	0.8	448.43	1.91
off-design point or takeoff	0.0	0.0	545.67	5.0

To gain a general understanding of the differences generated by these methods, the magnitudes of the difference values for all components were averaged and then tabulated. A comparison of average differences is shown in Table 8. The lowest differences generated in this study were from the component level simulations. System level simulations completed with and without a solver and NPSS derived scale factors generated comparable average difference values. Running at cruise with iDesign-generated map scale factors, the average difference is roughly half the amount generated by the simulation using map scale factors derived from NPSS data. However, when scale factors generated by iDesign at cruise are used at the takeoff operating point the average difference roughly doubles. Although there is a large rise, the values are still quite low, with an average difference of less than 0.25%, which shows that calculating map scale factors by data derivation or by the T-MATS iDesign tool can lead to difference values of less than 1% at the operating points tested.

Table 8. T-MATS JT9D simulation matching results.

Simulation Level	Solver	Operating Point	Scale Factor Derivation Method	Average Difference
Component	No	takeoff	NPSS derived	0.0550%
System	No	takeoff	NPSS derived	0.1558%
System	Yes	takeoff	NPSS derived	0.1891%
System	Yes	cruise	NPSS derived	0.1233%
System	Yes	takeoff	iDesign at cruise	0.2490%
System	Yes	cruise	iDesign at cruise	0.0910%

VI. Dynamic conversion

Once the matching was satisfactory, the model was converted from steady state to dynamic for transient testing. This was accomplished by introducing integrators to calculate the shaft speeds from the shaft accelerations and setting up the solver discussed earlier to converge the system during each time step.[6] Spool moments of inertia were unavailable from the steady state NPSS model, so values were assumed based on engine class as detailed in Appendix C. T-MATS contains a library of control component blocks that can be combined to build a controller for the open-loop system.[6] In this example, the input fuel flow was connected to a simple proportional-integral (PI) controller based on fan speed. A sample run of the T-MATS dynamic simulation is shown in Figure 2. At time 15s, the fan speed demand is reduced by 500 rpm. It can be observed that N1, N2, gross thrust, HPC exit temperature, and HPC exit pressure reduce as the fuel flow is reduced. Startup transients before 5 seconds are due to the system converging and then reaching a steady state value, and are typical with this type of simulation. Transient data for matching were unavailable and therefore, a direct dynamic model matching was not possible. It should be noted that the control system being utilized here is for demonstration purposes only; it was designed to adjust the fuel flow based on the fan speed error and does not take into account all of the operability and safety limits required in a realistic engine control design.

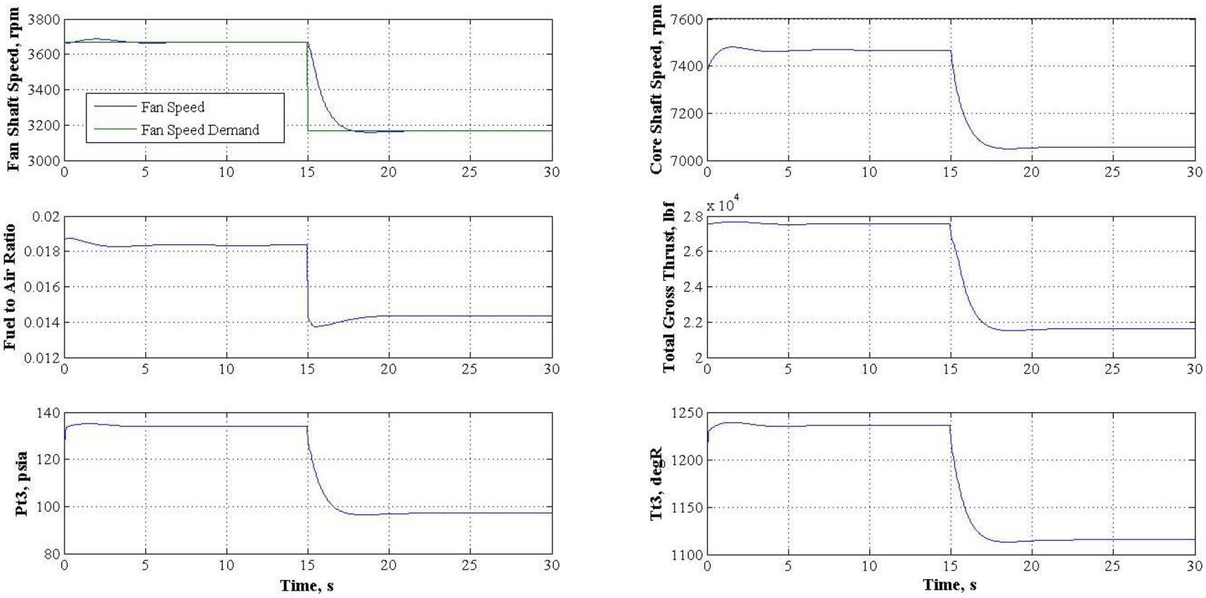

Figure 2. JT9D dynamic simulation.

VII. Conclusion

The T-MATS is a thermodynamic simulation system that allows developers to build custom thermodynamic system models within MATLAB/Simulink. The T-MATS Turbo-machinery modeling capability enables a user to create nonlinear steady state or dynamic gas turbine models. In creating a T-MATS simulation, the process described in this paper can be used to expedite the simulation development by leveraging NPSS data. As an example, a JT9D engine model, a publicly available dual-spool high bypass gas turbine, was created from a pre-existing steady state NPSS model. The process consisted of selecting components, determining maps and constants, and then performing component level testing. This was followed by system level testing without a solver, and finally performing system level testing with a solver. Demonstration of this process resulted in an acceptable level of matching error for all operating points tested. Once a reasonable match had been achieved, the steady state simulation was updated in a straightforward manner by inserting appropriate dynamics, as well as a simple PI fan speed controller. This approach demonstrates the capability of simulating and evaluating closed loop responses when starting with a steady state NPSS model and that the T-MATS package provides the ability to convert an industry standard NPSS simulation into the user-friendly Simulink environment for controls development.

Appendix

A. NPSS data output:

```
************************************************************************************************************************
Date:08/27/13   Time:10:15:44   Model:                   Rough JT9D Turbofan Performance Model   converge = 1   CASE:   0
Version:          NPSS_2.4        Gas Package:  Janaf      iter/pass/Jacb/Broy=  3/  6/ 1/ 1       Run by:         tlavelle

                         SUMMARY OUTPUT DATA
  MN alt     dTamb        W         Fn     TSFC      Wfuel      BPR     VTAS      OPR       T4      T41      T49
 0.000 0.0    27.00   1539.20    50012.9   0.3597   17987.64   5.2751     0.00   20.218   2730.0   2662.0   2142.4
                core size:19.92   Q:sdfsffafd  0.00   Maximum Thrust:    0.0    % Thrust:  0.0    Power code:  0.0

                          FLOW STATION DATA
                               W        Pt        Tt        ht      FAR       Wc        Ps        Ts     Aphy       MN      gamt
FS_1  Start.Fl_O         1539.20    14.696    545.67      0.30   0.0000  1578.75    14.696    545.67   -----.-   0.0000   1.39984
FS_2  Inlet.Fl_O         1539.20    14.578    545.67      0.30   0.0000  1591.49    12.015    516.34   5944.3    0.5329   1.39984
FS_21 Fan.Fl_O           1539.20    23.370    632.66     21.21   0.0000  1068.99    20.341    608.13   4515.2    0.4500   1.39800
FS_13 Splitter.Fl_O2     1293.91    23.370    632.66     21.21   0.0000   898.63    20.341    608.13   3795.7    0.4500   1.39800
FS_22 Splitter.Fl_O1      245.29    23.370    632.66     21.21   0.0000   170.35    20.341    608.13    719.5    0.4500   1.39800
FS_23 Duct1.Fl_O          245.29    23.312    632.66     21.21   0.0000   170.78    20.290    608.13    721.4    0.4500   1.39800
FS_24 LPC.Fl_O            245.29    52.451    821.24     66.89   0.0000    86.48    45.679    789.91    366.0    0.4500   1.39077
FS_25 Duct2.Fl_O          245.29    52.320    821.24     66.89   0.0000    86.69    45.565    789.91    366.9    0.4500   1.39077
FS_3  HPC.Fl_O            245.29   297.128   1398.32    212.41   0.0000    19.92   279.658   1376.22    119.9    0.3000   1.35605
FS_36 Bld3.Fl_O           223.21   297.128   1398.32    212.41   0.0000    18.13   279.658   1376.22    109.1    0.3000   1.35605
FS_4  Burner.Fl_O         228.21   280.786   2730.00    181.05   0.0224    27.40   279.007   2726.07    484.5    0.1000   1.28838
FS_45 HPT.Fl_O            250.28   104.217   2142.30     41.20   0.0204    71.73    98.312   2112.97    439.0    0.3000   1.30810
FS_48 Duct3.Fl_O          250.28   103.696   2142.35     41.20   0.0204    72.09    97.820   2113.02    441.2    0.3000   1.30810
FS_5  LPT.Fl_O            250.28    22.748   1529.55   -132.17   0.0204   277.66    20.978   1498.73   1470.0    0.3500   1.33466
FS_7  Duct4.Fl_O          250.28    22.521   1529.61   -132.17   0.0204   280.48    21.605   1513.77   2008.9    0.2500   1.33466
FS_9  Core_Nozz.Fl_O      250.28    22.521   1529.67   -132.17   0.0204   280.48    14.696   1372.96    855.7    0.8193   1.33465
FS_15 BypBld.Fl_O        1293.91    23.370    632.66     21.21   0.0000   898.63    20.341    608.13   3795.7    0.4500   1.39800
FS_17 Duct5.Fl_O         1293.91    23.195    632.66     21.21   0.0000   905.42    20.188    608.13   3824.4    0.4500   1.39800
FS_19 Byp_Nozz.Fl_O      1293.91    23.195    632.66     21.21   0.0000   905.42    14.696    555.46   2706.4    0.8345   1.39800

TURBOMACHINERY PERFORMANCE DATA
           Wc       PR      eff        Nc      TR     efPoly        pwr    SMN     SMW
Fan   1591.49    1.603   0.9038  3656.047  1.1594     0.9100   -45538.1  27.92   16.18
LPC    170.78    2.250   0.8658  3395.415  1.2981     0.8799   -15854.5  30.00   22.36
HPC     86.69    5.679   0.8625  6357.714  1.7027     0.8900   -50501.1  20.00   19.97
HPT     27.40    2.694   0.9144   153.112  1.2274     0.8987    50501.1
LPT     72.09    4.558   0.9288    81.019  1.4009     0.9135    61391.9

TURBOMACHINERY MAP DATA
        WcMap    PRmap   effMap     NcMap  R/Parm  s_WcDes  s_PRdes   s_effDes      s_NcDes
Fan   3051.46    1.420   0.9337     0.927  2.0000   0.5215   1.4369     0.9679    3943.9560
LPC    183.19    1.383   0.9018     0.927  1.7688   0.9322   3.2631     0.9600    3662.7989
```

```
HPC     206.12   22.630   0.8508     1.000   2.0805    0.4206   0.2163    1.0137   6357.7140
HPT      30.15    5.000   0.9328   100.000   5.0000    1.4087   0.4236    0.9803      1.5311
LPT     149.90    6.000   0.9276   100.000   6.0000    0.7453   0.7117    1.0013      0.8102

===INLETS====    eRam       Afs    Fram
Inlet          0.9920  -----.--     0.0   BLEEDS -   interstg   Wb/Win    BldWk     BldP          W        Tt       ht        Pt
                                          CstmrBld   HPC.C>     0.0000   0.3500   0.1465     0.0000   1027.77   117.83    88.184
====DUCTS====dPnorm      MN    Aphy       C_LPTexit  HPC.C>     0.0000   0.3500   0.1465     0.0000   1027.77   117.83    88.184
Duct1          0.0025  0.4500  719.55     C_LPTinlt  HPC.C>     0.0000   0.3500   0.1465     0.0000   1027.77   117.83    88.184
Duct2          0.0025  0.4500  366.01
Duct3          0.0050  0.3000  439.01     BLEEDS -     output   Wb/Win   hscale   Pscale          W        Tt       ht        Pt
Duct4          0.0100  0.35001470.05      C_HPTexit   Bld3.>    0.0350   1.0000   1.0000     8.5850   1398.32   212.41   297.128
Duct5          0.0075  0.45003795.70      C_HPTinlt   Bld3.>    0.0550   1.0000   1.0000    13.4908   1398.32   212.41   297.128
                                          Duct5Lk    BypBl>     0.0000   1.0000   1.0000     0.0000    632.66    21.21    23.370

==SPLITTERS==       BPR    dP/P 1    dP/P 2
Splitter         5.2751    0.0000    0.0000

===SHAFTS====     Nmech    trq in    pwr in
HP_Shaft         8000.0   33154.7   50501.1
LP_Shaft         3750.0   85983.3   61391.9


===BURNERS===     TtOut       eff    dPnorm     Wfuel      FAR
Burner          2729.98    0.9940    0.0550   4.99657  0.02238


===NOZZLES===        PR       Cfg      CdTh        Cv      Ath    MNth    Vact       Fg
Byp_Nozz          1.578    0.9975    1.0000    0.9975  2706.42   0.835   961.7  38674.0
Core_Nozz         1.532    0.9999    1.0000    0.9999   855.75   0.819  1457.6  11338.9
```

B. T-MATS formatted maps

Fan map:

```
MWS.FAN.NcVec = [ 0.500 0.600 0.700 0.800 0.850 0.900 0.950...
            1.000 1.050 1.100 1.150];

 MWS.FAN.RlineVec = [   1.000   1.200    1.400    1.600    1.800...
               2.000   2.200    2.400    2.600    2.800...
               3.000   3.200];


%========= Wc = f(NcMap, RlineMap) =================
 MWS.FAN.WcArray = [ 1129.0825  1258.8474  1382.1909  1498.6384  1607.7749...
             1709.2917  1802.8665  1888.3794  1965.6499  2034.6738...
             2095.4690  2148.1226;
             1396.9045  1546.6948  1685.6306  1813.0896  1928.5979...
             2031.8818  2122.7800  2201.2161  2267.4148  2321.6023...
             2364.1392  2395.5188;
             1636.3572  1816.0977  1977.3179  2119.2051  2241.3333...
             2343.6414  2426.3811  2490.0854  2535.5586  2563.8640...
             2576.1626  2576.9243;
             1884.0764  2099.0234  2284.7820  2440.4712  2565.9761...
             2661.9326  2729.5161  2770.3828  2786.6572  2787.1653...
             2787.1653  2787.1653;
             2049.3005  2269.7354  2456.5759  2609.2085  2727.9202...
             2813.8420  2868.7388  2894.8994  2898.0327  2898.0327...
             2898.0327  2898.0327;
             2288.3171  2492.5862  2662.3203  2797.5042  2898.9448...
             2968.1807  3007.2944  3018.8535  3018.8535  3018.8535...
```

```
                  3018.8535  3018.8535;
                  2562.6401  2733.0469  2872.2166  2980.6609  3059.4998...
                  3110.3328  3135.1262  3138.4663  3138.4663  3138.4663...
                  3138.4663  3138.4663;
                  2769.6670  2909.9534  3022.0933  3106.9241  3165.6892...
                  3199.9995  3211.6814  3211.7039  3211.7039  3211.7039...
                  3211.7039  3211.7039;
                  2954.6721  3056.6882  3137.9358  3199.1514  3241.2805...
                  3265.4429  3272.8789  3272.8789  3272.8789  3272.8789...
                  3272.8789  3272.8789;
                  3130.1790  3195.2559  3246.9822  3285.8452  3312.4382...
                  3327.4097  3331.4949  3331.4949  3331.4949  3331.4949...
                  3331.4949  3331.4949;
                  3275.6797  3310.4507  3338.3101  3359.5127  3374.2856...
                  3382.9170  3385.6724  3385.6724  3385.6724  3385.6724...
                  3385.6724  3385.6724];



%========= eff = f(NcMap, RlineMap) =================
 MWS.FAN.EffArray = [0.7900  0.8409  0.8835  0.9160  0.9360  0.9401...
                  0.9257  0.8894  0.8233  0.7152  0.5453  0.2792
                  0.7728  0.8298  0.8776  0.9141  0.9368  0.9421...
                  0.9271  0.8885  0.8179  0.7026  0.5223  0.2425
                  0.7237  0.7965  0.8579  0.9053  0.9352  0.9432...
                  0.9256  0.8776  0.7882  0.6405  0.4067  0.0380
                  0.6854  0.7705  0.8423  0.8978  0.9331  0.9431...
                  0.9239  0.8702  0.7703  0.6058  0.3474  0.0000
                  0.7003  0.7808  0.8483  0.9001  0.9327  0.9418...
                  0.9243  0.8763  0.7882  0.6463  0.4299  0.1059
                  0.7556  0.8178  0.8692  0.9080  0.9318  0.9379...
                  0.9248  0.8907  0.8305  0.7378  0.6034  0.4147
                  0.8143  0.8549  0.8878  0.9120  0.9262  0.9290...
                  0.9201  0.8991  0.8639  0.8121  0.7405  0.6453
                  0.8431  0.8697  0.8909  0.9059  0.9142  0.9150...
                  0.9081  0.8936  0.8704  0.8373  0.7928  0.7353
                  0.8613  0.8780  0.8909  0.8998  0.9042  0.9037...
                  0.8984  0.8884  0.8731  0.8519  0.8243  0.7893
                  0.8696  0.8785  0.8851  0.8892  0.8908  0.8896...
                  0.8856  0.8791  0.8697  0.8573  0.8416  0.8222
                  0.8644  0.8680  0.8704  0.8716  0.8714  0.8699...
                  0.8670  0.8629  0.8575  0.8507  0.8424  0.8326];

%========= PR = f(NcMap, RlineMap) =================
 MWS.FAN.PRArray = [ 1.1119  1.1119  1.1098  1.1058  1.0998  1.0919...
                  1.0822  1.0711  1.0586  1.0446  1.0292  1.0125;
                  1.1622  1.1638  1.1621  1.1570  1.1487  1.1372...
                  1.1228  1.1060  1.0869  1.0655  1.0420  1.0164;
                  1.2172  1.2249  1.2266  1.2221  1.2115  1.1950...
                  1.1733  1.1471  1.1168  1.0825  1.0447  1.0035;
                  1.2858  1.3036  1.3114  1.3090  1.2965  1.2742...
                  1.2433  1.2052  1.1606  1.1099  1.0537  1.0000;
                  1.3350  1.3543  1.3628  1.3603  1.3467  1.3224...
                  1.2887  1.2474  1.1988  1.1438  1.0830  1.0174;
                  1.4075  1.4214  1.4258  1.4204  1.4055  1.3813...
                  1.3489  1.3098  1.2643  1.2131  1.1567  1.0957;
```

```
                    1.4915  1.4971  1.4955  1.4869  1.4712  1.4487...
                    1.4202  1.3867  1.3485  1.3058  1.2591  1.2086;
                    1.5579  1.5567  1.5503  1.5386  1.5218  1.5000...
                    1.4738  1.4439  1.4106  1.3739  1.3341  1.2914;
                    1.6184  1.6123  1.6024  1.5888  1.5717  1.5510...
                    1.5273  1.5010  1.4722  1.4410  1.4076  1.3720;
                    1.6748  1.6655  1.6538  1.6399  1.6238  1.6054...
                    1.5852  1.5633  1.5399  1.5150  1.4886  1.4608;
                    1.7224  1.7122  1.7008  1.6882  1.6744  1.6595...
                    1.6434  1.6266  1.6089  1.5904  1.5711  1.5510];
```

LPC map:

```
MWS.LPC.NcVec = [ 0.300 0.400 0.500 0.600 0.700 0.800 0.900...
                  1.000 1.100 1.200 1.250];

 MWS.LPC.RlineVec = [    1.000    1.200    1.400    1.600    1.800    2.000...
                     2.200    2.400    2.600    2.800    3.000    3.200];


%========= Wc = f(NcMap, RlineMap) =================
 MWS.LPC.WcArray = [ 38.0744   42.9399   47.7510   52.5016   57.1863   61.7994...
                   66.3359   70.7905   75.1584   76.5663   76.5663   76.5663;
                   54.0383   60.0388   65.9233   71.6816   77.3038   82.7808...
                   88.1038   93.2648   98.2565  101.0545  101.0545  101.0545;
                   70.3200   77.5153   84.4949   91.2421   97.7419  103.9805...
                  109.9459  115.6273  121.0156  124.6409  124.6409  124.6409;
                   87.4860   95.6896  103.5393  111.0128  118.0907  124.7566...
                  130.9971  136.8019  142.1633  146.2312  146.2312  146.2312;
                  105.8588  114.8071  123.2285  131.0978  138.3948  145.1045...
                  151.2169  156.7268  161.6340  165.7319  165.7319  165.7319;
                  125.1164  134.6062  143.3572  151.3454  158.5548  164.9773...
                  170.6127  175.4677  179.5554  182.8951  183.0717  183.0717;
                  144.4910  154.5703  163.6243  171.6346  178.5959  184.5149...
                  189.4099  193.3090  196.2491  198.2745  198.4155  198.4155;
                  165.9141  176.2228  185.1849  192.7986  199.0806  204.0644...
                  207.7979  210.3410  211.7638  212.1506  212.1506  212.1506;
                  188.5677  198.3532  206.6834  213.5745  219.0613  223.1942...
                  226.0370  227.6647  228.1611  228.1611  228.1611  228.1611;
                  214.1402  222.1943  228.9021  234.2963  238.4220  241.3359...
                  243.1030  243.7959  243.8124  243.8124  243.8124  243.8124;
                  227.8569  234.5820  240.1193  244.5040  247.7802  250.0000...
                  251.2213  251.5216  251.5216  251.5216  251.5216  251.5216];



%========= eff = f(NcMap, RlineMap) =================
 MWS.LPC.EffArray = [.7256  .7656  .7978  .8195  .8274  .8164  .7494...
                    .5651  .1931  .0000  .0000  .0000;
                         .7474  .7848  .8147  .8351  .8430  .8339  .7757...
                    .6161  .3003  .0000  .0000  .0000;
                    .7610  .7984  .8286  .8496  .8586  .8516  .7977...
                    .6479  .3526  .0000  .0000  .0000;
                    .7744  .8117  .8421  .8637  .8738  .8685  .8183...
                    .6765  .3970  .0000  .0000  .0000;
                    .7872  .8240  .8542  .8759  .8866  .8827  .8360...
                    .7028  .4407  .0000  .0000  .0000;
                    .7965  .8329  .8627  .8843  .8953  .8924  .8485...
                    .7222  .4748  .0391  .0000  .0000;
                    .7997  .8368  .8673  .8896  .9013  .8991  .8561...
                    .7310  .4858  .0551  .0000  .0000;
                    .8034  .8405  .8712  .8937  .9058  .9042  .8628...
                    .7420  .5068  .0979  .0000  .0000;
                    .8214  .8533  .8793  .8981  .9079  .9062  .8724...
                    .7766  .5961  .2955  .0000  .0000;
                    .8425  .8663  .8853  .8985  .9047  .9025  .8778...
                    .8117  .6929  .5052  .2255  .0000;
                    .8540  .8731  .8880  .8981  .9024  .9000  .8800...
                    .8286  .7386  .6003  .4004  .1206];

%========= PR = f(NcMap, RlineMap) =================
 MWS.LPC.PRArray = [ 1.0423  1.0412  1.0393  1.0367  1.0333  1.0292  1.0234...
                    1.0151  1.0043  1.0000  1.0000  1.0000;
                    1.0760  1.0738  1.0704  1.0658  1.0600  1.0530  1.0434...
                    1.0297  1.0122  1.0000  1.0000  1.0000;
                    1.1215  1.1180  1.1127  1.1055  1.0965  1.0856  1.0707...
                    1.0497  1.0228  1.0000  1.0000  1.0000;
                    1.1789  1.1738  1.1660  1.1555  1.1423  1.1266  1.1052...
                    1.0753  1.0374  1.0000  1.0000  1.0000;
                    1.2494  1.2422  1.2312  1.2167  1.1986  1.1771  1.1481...
                    1.1078  1.0572  1.0000  1.0000  1.0000;
                    1.3353  1.3253  1.3105  1.2910  1.2669  1.2384  1.2002...
                    1.1476  1.0822  1.0056  1.0000  1.0000 ;
                    1.4411  1.4282  1.4088  1.3830  1.3512  1.3136  1.2632...
                    1.1942  1.1088  1.0101  1.0000  1.0000;
                    1.5724  1.5561  1.5313  1.4982  1.4572  1.4088  1.3440...
                    1.2556  1.1472  1.0233  1.0000  1.0000;
                    1.7323  1.7101  1.6785  1.6379  1.5888  1.5318  1.4572...
                    1.3572  1.2358  1.0982  1.0000  1.0000;
                    1.9360  1.9056  1.8662  1.8184  1.7625  1.6991  1.6190...
                    1.5142  1.3887  1.2471  1.0944  1.0000;
                    2.0507  2.0158  1.9729  1.9223  1.8645  1.8000  1.7201...
                    1.6176  1.4958  1.3584  1.2098  1.0546];

HPC map:
 MWS.HPC.NcVec = [ 0.500   0.600 0.700 0.750 0.800 0.850 0.900 0.925 0.950 0.975...
                   1.000   1.025 1.050 ];

 MWS.HPC.RlineVec = [  1.000    1.200    1.400    1.600    1.800    2.000...
                     2.200    2.400    2.600    2.800    3.000];

%========= Wc = f(NcMap, RlineMap) =================
 MWS.HPC.WcArray = [ 22.7411   24.0487   25.1548   26.0615   26.7738   27.2992...
                    27.6470   27.8286   27.8634   27.8634   27.8634;
                    31.7548   33.1181   34.2670   35.2054   35.9397   36.4783...
                    36.8308   37.0085   37.0362   37.0362   37.0362;
                    46.1066   47.4088   48.5066   49.4046   50.1096   50.6291...
                    50.9717   51.1469   51.1757   51.1757   51.1757;
                    56.7268   58.0480   59.1608   60.0704   60.7837   61.3084...
                    61.6527   61.8260   61.8517   61.8517   61.8517;
                    70.1448   71.5163   72.6688   73.6088   74.3429   74.8795...
                    75.2269   75.3943   75.4134   75.4134   75.4134;
                    89.3764   90.9746   92.3098   93.3900   94.2232   94.8199...
```

```
               95.1897   95.3442   95.3504   95.3504   95.3504;
              118.0620  120.1207  121.8253  123.1867  124.2166  124.9292...
              125.3385  125.4609  125.4609  125.4609  125.4609;
              138.5093  140.8966  142.8639  144.4238  145.5916  146.3836...
              146.8174  146.9192  146.9192  146.9192  146.9192;
              160.6243  162.5676  164.1805  165.4722  166.4536  167.1370...
              167.5334  167.6563  167.6563  167.6563  167.6563;
              181.7993  183.4993  184.9150  186.0545  186.9260  187.5389...
              187.9029  188.0273  188.0271  188.0271  188.0271;
              202.6315  203.5858  204.3958  205.0661  205.5998  206.0000...
              206.2702  206.4145  206.4418  206.4418  206.4418;
              209.9986  210.5917  211.1029  211.5321  211.8825  212.1554...
              212.3516  212.4735  212.5220  212.5227  212.5227;
              216.6847  217.0279  217.3287  217.5860  217.8015  217.9767...
              218.1106  218.2041  218.2586  218.2739  218.2739];



%========= eff = f(NcMap, RlineMap) =================
 MWS.HPC.EffArray = [ 0.6753  0.6913  0.7016  0.7050  0.7004  0.6864  0.6570...
               0.6044  0.5236  0.4075  0.2467;
                      0.6953  0.7094  0.7184  0.7214  0.7176  0.7058  0.6812...
               0.6378  0.5717  0.4783  0.3512;
               0.7248  0.7359  0.7429  0.7452  0.7424  0.7335  0.7154...
               0.6838  0.6366  0.5713  0.4848;
               0.7427  0.7533  0.7600  0.7627  0.7606  0.7533  0.7379...
               0.7108  0.6703  0.6147  0.5414;
               0.7634  0.7736  0.7804  0.7834  0.7822  0.7762  0.7630...
               0.7394  0.7041  0.6556  0.5920;
               0.7891  0.8008  0.8090  0.8134  0.8136  0.8092  0.7974...
               0.7754  0.7417  0.6950  0.6335;
               0.8139  0.8280  0.8385  0.8449  0.8469  0.8439  0.8333...
               0.8117  0.7779  0.7303  0.6671;
               0.8206  0.8356  0.8469  0.8541  0.8567  0.8544  0.8442...
               0.8229  0.7892  0.7416  0.6783;
               0.8403  0.8512  0.8593  0.8643  0.8660  0.8641  0.8566...
               0.8415  0.8179  0.7852  0.7423;
               0.8408  0.8492  0.8552  0.8588  0.8597  0.8578  0.8516...
               0.8394  0.8209  0.7954  0.7624;
               0.8470  0.8505  0.8529  0.8539  0.8536  0.8520  0.8483...
               0.8418  0.8324  0.8200  0.8043;
               0.8350  0.8364  0.8371  0.8370  0.8362  0.8346  0.8318...
               0.8275  0.8217  0.8141  0.8049;
               0.8202  0.8203  0.8201  0.8195  0.8185  0.8171  0.8152...
               0.8124  0.8088  0.8045  0.7992];

%========= PR = f(NcMap, RlineMap) =================
 MWS.HPC.PRArray = [  2.4769   2.4288   2.3620   2.2778   2.1774   2.0627...
               1.9284   1.7711   1.5958   1.4083   1.2146;
               3.4633   3.3778   3.2643   3.1248   2.9619   2.7787...
               2.5679   2.3253   2.0595   1.7802   1.4973;
               5.0821   4.9375   4.7554   4.5391   4.2923   4.0194...
               3.7106   3.3602   2.9800   2.5826   2.1813;
               6.3490   6.1658   5.9371   5.6667   5.3594   5.0204...
               4.6377   4.2042   3.7342   3.2431   2.7467;
               8.0021   7.7686   7.4792   7.1388   6.7532   6.3287...
```

```
                     5.8504   5.3097   4.7237   4.1114   3.4919;
                    10.4899  10.1976   9.8249   9.3786   8.8669   8.2989...
                     7.6539   6.9201   6.1229   5.2899   4.4495 ;
                    14.4564  14.0970  13.6074  12.9977  12.2808  11.4715...
                    10.5377   9.4621   8.2878   7.0614   5.8306;
                    17.4426  17.0500  16.4870  15.7661  14.9034  13.9183...
                    12.7692  11.4347   9.9718   8.4425   6.9104;
                    20.7403  20.2486  19.6093  18.8329  17.9324  16.9227...
                    15.7626  14.4263  12.9562  11.3983   9.8011;
                    23.8298  23.2601  22.5536  21.7200  20.7705  19.7178...
                    18.5212  17.1524  15.6466  14.0424  12.3810;
                    26.6962  26.0933  25.4175  24.6733  23.8656  22.9999...
                    22.0495  20.9930  19.8436  18.6163  17.3267;
                    27.6439  27.0687  26.4522  25.7969  25.1052  24.3798...
                    23.6033  22.7614  21.8600  20.9058  19.9057;
                    28.4663  27.9667  27.4460  26.9054  26.3460  25.7690...
                    25.1640  24.5225  23.8472  23.1399  22.4038];
```

HPT map:

```
MWS.HPT.NcVec = [ 60 70 80 90 100 110];

 MWS.HPT.PRVec = [  3.000    3.250    3.500    3.750    4.000    4.250    4.500...
                 4.750    5.000    5.250    5.500    5.750    6.000    6.250...
                 6.500    6.750    7.000    7.250    7.500    8.000];


%========= Wc = f(NcMap, PRMap) =================
 MWS.HPT.WcArray = [...
                     30.446   30.533   30.568   30.572   30.572   30.572   30.572...
                     30.572   30.572   30.572   30.572   30.572   30.572   30.572...
                     30.572   30.572   30.572   30.572   30.572   30.572 ;
                     30.299   30.413   30.480   30.516   30.529   30.530   30.530...
                     30.530   30.530   30.530   30.530   30.530   30.530   30.530...
                     30.530   30.530   30.530   30.530   30.530   30.530 ;
                     30.120   30.239   30.317   30.368   30.398   30.415   30.421...
                     30.421   30.421   30.421   30.421   30.421   30.421   30.421...
                     30.421   30.421   30.421   30.421   30.421   30.421 ;
                     30.014   30.124   30.201   30.253   30.288   30.311   30.325...
                     30.333   30.337   30.337   30.337   30.337   30.337   30.337...
                     30.337   30.337   30.337   30.337   30.337   30.337 ;
                     29.856   29.948   30.013   30.059   30.091   30.113   30.128...
                     30.139   30.145   30.149   30.150   30.150   30.150   30.150...
                     30.150   30.150   30.150   30.150   30.150   30.150 ;
                     29.799   29.870   29.920   29.955   29.979   29.997   30.009...
                     30.017   30.023   30.026   30.028   30.029   30.029   30.029...
                     30.029   30.029   30.029   30.029   30.029   30.029];



%========= eff = f(NcMap, PRMap) =================
 MWS.HPT.EffArray = [...
                       0.8460   0.8405   0.8349   0.8296   0.8249   0.8206   0.8165...
                       0.8127   0.8092   0.8060   0.8030   0.8002   0.7976   0.7953...
                       0.7931   0.7911   0.7892   0.7875   0.7858   0.7826 ;
                       0.8879   0.8842   0.8804   0.8769   0.8735   0.8701   0.8670...
                       0.8640   0.8614   0.8590   0.8568   0.8548   0.8529   0.8511...
```

```
              0.8495   0.8479   0.8460   0.8436   0.8414   0.8373 ;
              0.9125   0.9111   0.9096   0.9078   0.9055   0.9034   0.9014...
              0.8995   0.8979   0.8964   0.8950   0.8936   0.8924   0.8903...
              0.8877   0.8853   0.8830   0.8808   0.8787   0.8749 ;
              0.9228   0.9242   0.9250   0.9247   0.9240   0.9232   0.9224...
              0.9217   0.9210   0.9203   0.9197   0.9188   0.9162   0.9137...
              0.9113   0.9091   0.9070   0.9050   0.9031   0.8980 ;
              0.9215   0.9258   0.9289   0.9304   0.9313   0.9319   0.9323...
              0.9326   0.9328   0.9329   0.9330   0.9311   0.9288   0.9266...
              0.9245   0.9225   0.9206   0.9188   0.9161   0.9107 ;
              0.9106   0.9176   0.9232   0.9267   0.9292   0.9312   0.9327...
              0.9340   0.9351   0.9361   0.9369   0.9349   0.9329   0.9311...
              0.9293   0.9276   0.9259   0.9239   0.9212   0.9161];

LPT map:
MWS.LPT.NcVec = [ 60 70 80 90 100 110 120];

 MWS.LPT.PRVec = [  3.000    3.250    3.500    3.750    4.000    4.250    4.500...
           4.750    5.000    5.250    5.500    5.750    6.000    6.250...
           6.500    6.750    7.000    7.250    7.500    8.000];


%========= Wc = f(NcMap, PRMap) =================
 MWS.LPT.WcArray = [...
          153.812  153.812  153.812  153.812  153.812  153.812  153.812...
          153.812  153.812  153.812  153.812  153.812  153.812  153.812...
          153.812  153.812  153.812  153.812  153.812  153.812 ;
          153.511  153.511  153.511  153.511  153.511  153.511  153.511...
          153.511  153.511  153.511  153.511  153.511  153.511  153.511...
          153.511  153.511  153.511  153.511  153.511  153.511 ;
          152.799  152.982  153.052  153.061  153.061  153.061  153.061...
          153.061  153.061  153.061  153.061  153.061  153.061  153.061...
          153.061  153.061  153.061  153.061  153.061  153.061 ;
          150.995  151.316  151.518  151.647  151.729  151.781  151.814...
          151.834  151.846  151.852  151.856  151.858  151.859  151.859...
          151.859  151.859  151.859  151.859  151.859  151.859 ;
          148.751  149.107  149.349  149.517  149.635  149.719  149.779...
          149.822  149.852  149.872  149.885  149.894  149.898  149.899...
          149.899  149.899  149.899  149.899  149.899  149.899 ;
          145.352  145.680  145.905  146.061  146.169  146.244  146.293...
          146.324  146.339  146.344  146.344  146.344  146.344  146.344...
          146.344  146.344  146.344  146.344  146.344  146.344 ;
          140.863  141.131  141.310  141.428  141.503  141.547  141.567...
          141.569  141.569  141.569  141.569  141.569  141.569  141.569...
          141.569  141.569  141.569  141.569  141.569  141.569];



%========= eff = f(NcMap, PRMap) =================
 MWS.LPT.EffArray = [...
             0.8388   0.8309   0.8234   0.8159   0.8091   0.8030   0.7975...
             0.7924   0.7876   0.7832   0.7791   0.7753   0.7717   0.7684...
             0.7652   0.7623   0.7595   0.7568   0.7542   0.7495;
             0.8878   0.8813   0.8745   0.8685   0.8629   0.8577   0.8528...
             0.8484   0.8443   0.8404   0.8368   0.8334   0.8302   0.8272...
             0.8242   0.8210   0.8179   0.8150   0.8122   0.8071;
```

```
0.9201   0.9152   0.9105   0.9061   0.9018   0.8978   0.8940...
0.8905   0.8872   0.8840   0.8810   0.8776   0.8741   0.8707...
0.8676   0.8646   0.8618   0.8590   0.8565   0.8516;
0.9381   0.9360   0.9336   0.9310   0.9283   0.9257   0.9231...
0.9206   0.9182   0.9153   0.9119   0.9087   0.9056   0.9027...
0.8999   0.8973   0.8948   0.8924   0.8901   0.8858;
0.9447   0.9455   0.9456   0.9450   0.9440   0.9429   0.9417...
0.9404   0.9383   0.9355   0.9327   0.9301   0.9276   0.9252...
0.9229   0.9207   0.9186   0.9165   0.9146   0.9099;
0.9415   0.9454   0.9479   0.9495   0.9504   0.9510   0.9512...
0.9511   0.9492   0.9472   0.9452   0.9433   0.9414   0.9396...
0.9378   0.9361   0.9344   0.9326   0.9304   0.9262;
0.9295   0.9366   0.9419   0.9458   0.9487   0.9509   0.9526...
0.9538   0.9528   0.9517   0.9505   0.9493   0.9481   0.9468...
0.9456   0.9444   0.9432   0.9413   0.9395   0.9360];
```

C. T-MATS parameters

Table 9. Additional required T-MATS constants.

T-MATS Component	NPSS variable	T-MATS variable	Value	Description	Method
Inlet	eRam	eRambase_M	0.992	Base dP across inlet	Per Data
	eRam derived	X_eRamVec_M	[0.99 1 1.007 1.028 1.1 2]	Ram dP vector	Determined empirically by viewing change in pressure as Pt/Pa is changed
	eRam derived	T_eRamtbl_M	[1 1 1.001 1.002 1.003 1.003]	Ram dP pressure multiplier	Determined empirically by viewing change in pressure as Pt/Pa is changed
HPC	Bleeds - Wb/Win	C_FBD_M	[0.55 0.035]	Fractional bleed demand, for each bleed	Per Data, Bleed - output
	Bleeds - ht	C_BFht_M	[1 1]	Fractional bleed enthalpy	Calculated by assuming enthalpy rises linearly from component input to output.
	Bleeds - Pt	C_BFPt_M	[1 1]	Fraction bleed pressure	Calculated by assuming enthalpy rises linearly from component input to output.
Burner	LHV	LHV_M	18400	Lower Heating Value	Per Data, from model definition file
	dPnorm	dPnormBurner_M	0.055	dP across Burner	Per Data
	eff	Eff_M	0.944	Burner Efficiency	Per Data
HPT	C_HPT exit or inlt	CoolFlwPos_M	[0 1]	Location of input cooling flows	Per Data, HPT_in - 0 , HPT_exit - 1
Core Nozzle	Ath	N_Tarea_M	862.88	Nozzle throat area	Per Data
	Cv	CvArray_M	[0.999 0.999]	Cv value	Per Data, in this instance the value is constant
Byp Nozzle	Ath	N_Tarea_M	2706.4	Bypass nozzle throat area	Per Data
	Cv	CvArray_M	[0.9975 0.9975]	Cv value	Per Data, in this instance the value is constant
Duct1	dPnorm	dP_M	0.0025	dP loss across duct1	Per Data
Duct2	dPnorm	dP_M	0.0025	dP loss across duct2	Per Data
Duct3	dPnorm	dP_M	0.005	dP loss across duct3	Per Data
Duct4	dPnorm	dP_M	0.01	dP loss across duct4	Per Data
Duct5	dPnorm	dP_M	0.0075	dP loss across duct5	Per Data
HP Shaft	None	Shaft_Inertia_M	20	High Pressure Shaft Inertia	Assumption based on engine class. This is a dynamic criteria and NPSS data is steady state.
LP Shaft	None	Shaft_Inertia_M	100	Low Pressure Shaft Inertia	Assumption based on engine class. This is a dynamic criteria and NPSS data is steady state.

Acknowledgments

The authors would like to thank the NASA Aviation Safety Program's Vehicle Systems Safety Technologies Project for funding this work. Additionally the assistance of the NPSS team was critical in gaining a deeper understanding of the NPSS modeling process.

References

[1] Jones, S.M., "An Introduction to Thermodynamic Performance Analysis of Aircraft Gas Turbine Engine Cycles Using the Numerical Propulsion System Simulation Code," NASA/TM-2007-214690, March 2007.
[2] Jones, S.M., "Steady-State Modeling of Gas Turbine Engines Using The Numerical Propulsion System Simulation Code," GT2010-22350, June 2010.
[3] Nichols, L.D. and Chamis, C.C., "Numerical Propulsion System Simulation: An Interdisciplinary Approach," AIAA-91-3554, September 1991.
[4] May, R.D., Csank, J., Lavelle, T.M., Litt, J.S., and Guo T.H., "A High-Fidelity Simulation of a Generic Commercial Aircraft Engine and Controller," NASA/TM-2010-216810, October 2010.
[5] DeCastro, J.A., Litt, J.S., Frederick, D.K., "A Modular Aero-Propulsion System Simulation of a Large Commercial Aircraft Engine," NASA/TM-2008-215303, January 2008.
[6] Chapman, J.W., Lavelle, T.M., May, R.D., Litt, J.S., and Guo, T.M., "Toolbox for the Modeling and Analysis of Thermodyanmic Systems (T-MATS) User's Guide," NASA/TM-2014-216638, January 2014.
[7] Chapman, J.W., Lavelle, T.M., May, R.D., Litt, J.S., Guo, T-H., "Propulsion System Simulation Using the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS)," 2014 AIAA Joint Propulsion Conference, Cleveland, OH, Jul 28-30, 2014.
[8] Zinnecker, A.M., Chapman, J.W., Lavelle , T.M., and Litt, J.S., "Development of a twin-spool turbofan engine simulation using the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS)," 2014 AIAA Joint Propulsion Conference, Cleveland, OH, Jul 28-30, 2014.
[9] Daly, M., *Jane's Aero Engines*, Issue 28, Ed., Jane's Information Group, London, England, September 2010.
[10] Lavelle, T.M., Chapman, J.W., May, R.D., Litt, J.S., and Guo, T.H., "Cantera Integration with the Modeling and Analysis of Thermodynamic Systems (T-MATS)," 2014 AIAA Joint Propulsion Conference, Cleveland, OH, Jul 28-30, 2014.