Propulsion System Simulation Using the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS)

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

Thomas M. Lavelle[2]
National Aeronautics and Space Administration, Glenn Research Center, Cleveland, Ohio 44135

Ryan D. May[3]
Vantage Partners, LLC, Brook Park, Ohio 44142

Jonathan S. Litt[4], and Ten-Huei Guo[5]
National Aeronautics and Space Administration, Glenn Research Center, Cleveland, Ohio 44135

A simulation toolbox has been developed for the creation of both steady-state and dynamic thermodynamic software models. This paper describes the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS), which combines generic thermodynamic and controls modeling libraries with a numerical iterative solver to create a framework for the development of thermodynamic system simulations, such as gas turbine engines. The objective of this paper is to present an overview of T-MATS, the theory used in the creation of the module sets, and a possible propulsion simulation architecture. A model comparison was conducted by matching steady-state performance results from a T-MATS developed gas turbine simulation to a well-documented steady-state simulation. Transient modeling capabilities are then demonstrated when the steady-state T-MATS model is updated to run dynamically.

Nomenclature

Alt	= altitude
Ath	= throat area
BPR	= branch pressure ratio
C-MAPSS40k	= Commercial Modular Aero-Propulsion System Simulation 40,000 lbf thrust
Cv	= specific heat at constant volume
dT	= delta temperature from standard day
dt	= delta time
DLL	= dynamic-link library
Eff	= efficiency
f(x)	= generic function
FAR	= fuel to air ratio
Fg	= gross thrust
g	= acceleration of gravity
γ	= specific heat ratio

[1] Control Systems Engineer, 3000 Aerospace Parkway, AIAA Member
[2] Engineer, 21000 Brookpark Rd., MS 86-1, AIAA Member
[3] Controls Engineer, 3000 Aerospace Parkway, AIAA Member
[4] Control Systems Engineer, 21000 Brookpark Rd., MS 77-1, AIAA Senior Member
[5] Control Systems Research Engineer, 21000 Brookpark Rd., MS 77-1, AIAA Member

ht	= total enthalpy
hs	= static enthalpy
I	= shaft inertia
J	= Jacobian
k	= iteration
l	= iteration number
MN	= Mach number
N	= shaft speed
n	= iteration step counter
Nc	= corrected shaft speed
Ndmd	= demanded shaft speed into controller
Ndot	= derivative of shaft speed with respect to time
Nmech	= mechanical shaft speed
NPSS	= Numerical Propulsion System Simulation
NR	= Newton-Raphson
PI	= proportional integral
PLA	= power lever angle
PR	= pressure ratio
Pa	= ambient pressure
Ps	= static pressure
Pt	= total pressure
R	= specific gas constant
R-line	= unique line on the compressor map
ρ	= density
S	= entropy
SPR	= stall pressure ratio
SS	= steady-state
t	= time
T-MATS	= Toolbox for the Modeling and Analysis of Thermodynamic Systems
Trq	= torque
Ts	= static temperature
Tt	= total temperature
W	= mass flow rate
Wc	= mass flow rate corrected to local conditions
Wf	= mass flow rate corrected to local conditions
x	= generic variable
X_il	= generic inner loop variable
X_ol	= generic outer loop variable

I. Introduction

As computer technology has improved, advancements in propulsion system modeling have become synonymous with increased complexity. In many cases, this complexity stems from the integration of components created with different development tools. Control system design and research has been particularly affected by this type of integration complexity due to the need for incorporating legacy models into a dynamic simulation package. An effective means of reducing this model integration complexity is to build the complete system in a single development package, such as MATLAB/Simulink® (The MathWorks, Inc.). MATLAB is a design tool commonly used among control engineers and researchers and contains the graphical programming tool Simulink, which is designed for the simulation of dynamic systems.

A number of thermodynamic simulation tools that model propulsion systems have been developed, which integrate with Simulink with varying levels of difficulty. These tools include Numerical Propulsion System Simulation (NPSS), GasTurb®, TurboLib®, and Commercial Modular Aero-Propulsion System Simulation 40,000 (C-MAPSS40k). The NPSS software is a powerful and highly flexible platform in which propulsion system simulations can be created modularly.[1] Based on a code similar to C, NPSS offers modifiable gas turbine

components that may be combined into larger systems. While NPSS is very powerful, integration of an NPSS model with other platforms, such as Simulink, can be time consuming.[2] GasTurb, another gas turbine modeling program, offers a graphical user interface that may be used to create engine simulations.[3] While providing a more visual interface than NPSS, this product also requires a user to work with a stand-alone software package for simulation development. Additionally, integration with other modeling software can be challenging, as GasTurb requires the creation of specialized dynamic-link library (DLL) files for cross-platform integration. Alternatively, the Simulink plug-in ThermoLib is available to expand the Simulink block sets to include thermodynamic modeling capability directly in Simulink, eliminating the need for system integration.[4] ThermoLib blocks, however, have limited code access; and, as a result, updating components to meet modeling requirements can become complicated. Another Simulink-based tool is the C-MAPSS40k software developed by NASA that combines a realistic turbine engine simulation with an industry-standard controller.[5] Not only does C-MAPSS40k solve many model integration issues by allowing a designer to work totally in the MATLAB/Simulink environment, it also provides the user full access to the development code. Despite this code access, C-MAPSS40k is not a generic tool, as it is a model of a particular gas turbine engine that may not represent the system a developer is interested in.

To fill the need for a thermodynamic transient and controls design simulation package that is built in Simulink, has component-level flexibility, and maintains a generic simulation framework, the Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) software package was developed by NASA Glenn Research Center. T-MATS is an unrestricted, open source Simulink toolbox (plug-in) that can be used by any industry professionals or academics interested in the development of custom thermodynamic and controls simulations. The T-MATS software package contains specialized modeling libraries that add to the Simulink block sets, which include numerical iterative solvers and advanced turbo-machinery component models, enabling a developer to create complex propulsion system simulations, such as a gas turbine engine.[6]

Installation of T-MATS creates a new Simulink library with the following block sets: *Effectors and Controls*, containing controller algorithms and control system hardware simulation blocks; *Numerical Methods*, offering self-contained numerical method operations; *Solvers*, providing flexible iterative solvers; and *Turbo-machinery*, consisting of specialized gas turbine simulation functionality. By using these block sets, a user can rapidly create complex simulations by dragging blocks into a Simulink model and connecting them appropriately. Furthermore, T-MATS block design details are parameterized, which allows a developer to quickly provide the required information. In addition to block sets for thermodynamic modeling, T-MATS contains *Model Development Tools* designed to help reduce the time required to create a Simulink simulation.

The remainder of this paper discusses the modeling theory, application to a gas turbine engine simulation, and a model matching between T-MATS and a "truth" model from NPSS. Section II presents an architecture overview for model development using the T-MATS software package. Simulink blocks and tools added to MATLAB/Simulink upon installation of T-MATS are discussed in Section III. Section IV details a model matching case for T-MATS blocks by comparing gas turbine data generated from an NPSS model, component by component, with a T-MATS model developed with the same maps and constants. A summary may be found in Section V. Finally, information for obtaining T-MATS is given in Appendix B.

II. System Model Architecture Overview

Simulations can be divided into two major categories: steady-state and dynamic. A steady-state simulation is a model that is unchanging with time, and can be used for analysis and design in the absence of requirements on system transients. A dynamic simulation is one that changes with time and can be used to model systems for controls development, a key feature of T-MATS. In thermodynamic modeling, both types of simulations are useful and T-MATS contains all the tools required for setting up either.

In setting up simulations, T-MATS makes use of a generic and modifiable multi-loop architecture, which contains an "outer" loop iterated over time, t, and an "inner" loop set up to solve for internal variable imbalances over discrete iterations, k. In turbo-machinery systems, the form and solution method for these imbalances is often dictated by plant composition.[7] A diagram of this architecture, with a single "inner" loop (blue) and an "outer" loop (green), can be seen in Figure 1. The inner loop plant contains all components that have thermodynamic imbalances, such as turbines, compressors, and nozzles. Plant imbalance measurements, or dependents, $f(x(k))$, are routed from the inner loop plant to the iterative solver. The iterative solver then adjusts the plant dependents by making changes to the inner loop plant dependent effectors, or independents, $X\_il(k+1)$. In a typical gas turbine simulation, dependents often take the form of component flow errors, while each independent may specify input mass flow, compressor R-line, bypass pressure ratio (*BPR*), or turbine pressure ratio (*PR*). The iterative solver works in tandem with a "while loop" to pause the simulation at each time step, allowing the iterations to reach plant convergence (i.e.,

drive the dependents to zero) before the next time step. Components within the "outer" loop act to integrate the system within the time domain and may or may not be time dependent. These include outer loop effectors that feed signals from outside the "inner" loop, such as environmental conditions like altitude (*Alt*) or Mach number (*MN*), and system commands, such as power lever angle (PLA). Controllers, shaft dynamics, and other time-dependent portions of the simulation are located within the outer loop plant, which uses a feedback to the inner loop plant (*y(t)*) to generate inner loop plant inputs (*X_ol(t+dt)*). The architecture shown in Figure 1 is appropriate for the creation of a dynamic or steady-state simulation. (To create a steady-state simulation, the portion shaded in green can be removed since the simulation is not time dependent.)

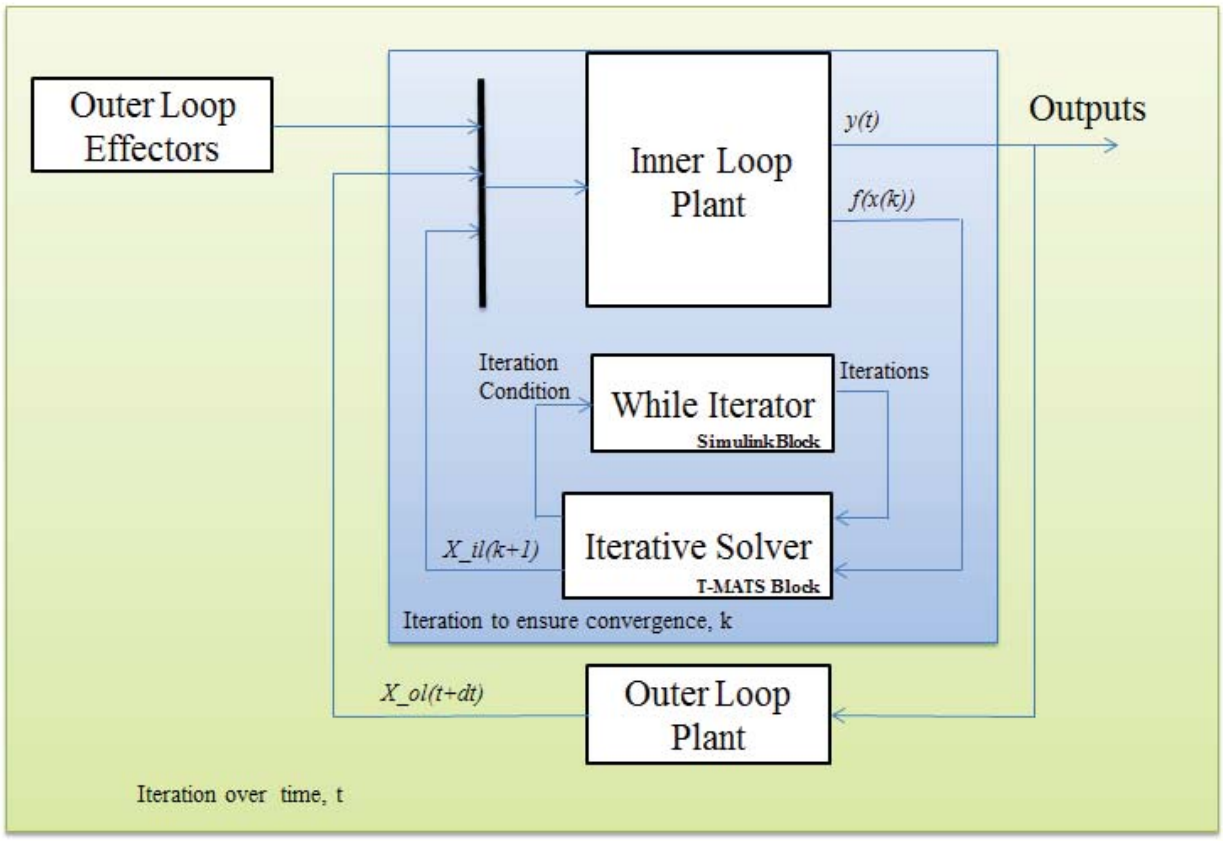

Figure 1. T-MATS sample model architecture.

III. T-MATS Components and Theoretical Development

The T-MATS software package comes with components and tools that can be used for implementing models of thermodynamic systems. This section will review the components for modeling effectors, controls, and turbo-machinery, the iterative solvers, and the model development tools, discussing some of the theory behind their development, and detailing how they may be used in the simulation creation process.

A. Turbo-Machinery Components

Turbo-machinery, as it relates to T-MATS, is defined as the components that compose a gas turbine engine. A simple example of a turbo-machinery system is a turbojet, which contains a compressor, a burner, a turbine, and a nozzle in series (see the example in Section IV for further details of a turbojet model). The T-MATS software package contains the turbo-machinery component models listed in Table 1, which may be combined to create a 0-D component-level model of a gas turbine engine. A 0-D simulation is one in which the component-level dynamics are not modeled. It is important to note that T-MATS is completely open source and could be altered for higher order simulations, but the included turbo-machinery components are only valid for 0-D analyses. The following sub-sections describe the theory behind each T-MATS component used for gas turbine simulation development.

Table 1. Description of components in the *Turbo-machinery* block set.

T-MATS Block	Description
Ambient	Converts Environmental conditions into measurable parameters (e.g., *Alt* to ambient pressure, *Pa*)
Burner	Transfers energy of a fuel burn to a gas stream
Compressor	Compresses gas stream based on input power
Duct	Models effects of a gas stream moving through a duct (pressure loss)
Inlet	Conditions gas stream for entry into system
Nozzle	Constricts the gas stream, modeling thrust
Shaft	Connection for rotating components
Splitter	Forks a gas stream
Turbine	Extract energy from a gas stream, creating a torque on a shaft
Valve	Split gas stream in relation to a fractional input

1. Ambient component

The ambient component is used for converting the environmental variables altitude (*Alt*), temperature different from standard day (*dT*), and Mach number (*MN*) to the thermodynamic variables enthalpy, temperature, and pressure. Static pressure (*Ps*) and temperature (*Ts*) are determined from an empirical model based on *Alt* and *MN*. Total temperature (*Tt*) and pressure (*Pt*) are calculated based on the isentropic relations shown in Eqs. (1) and (2). It should be noted that the relationships in Eqs. (1) and (2) assume the specific heat ratio (γ) is a constant and modified forms of these equations should be used when applying the equations to high *MN* flow. Work is currently underway to update the ambient component to eliminate this issue.

$$\frac{\mathrm{Tt}}{Ts} = 1 + MN^2 * \left(\frac{(\gamma-1)}{2}\right) \tag{1}$$

$$\frac{\mathrm{Ps}}{Pt} = \left(\frac{Ts}{Tt}\right)^{\gamma/(\gamma-1)} \tag{2}$$

Enthalpy is determined empirically from a T-MATS-provided table that relates enthalpy to temperature based on an assumed gas composition of air and Jet-A fuel. It is important to note that, in other components, entropy is also calculated empirically based on temperature, pressure, and the same gas compositions.

2. Burner component

The burner component, or combustor, acts to move energy from a fuel source into the gas stream. The magnitude of this energy transfer is modeled by performing a weighted average of the enthalpy of the gas stream (ht_{air}) and the lower heating value (*LHV*) of the fuel source taking into account burn efficiency (*Eff*), based on fuel to air ratio (*FAR*) as shown in Eq. (3). The total enthalpy (ht_{total}) of the air and mixed flow are used to empirically calculate an exit temperature using tables, as described above for the ambient component. Module exit pressure and burn efficiency are based on simple scale factor multipliers on input pressure and enthalpy respectively.

$$ht_{total} = ht_{air} * (1 - FAR) + LHV * FAR * Eff \tag{3}$$

3. Compressor component

The compressor component represents a mechanical device that increases the pressure of a gas, which can take many forms, such as a centrifugal or multi-stage axial compressor, or even a fan. Whatever form the physical compressor takes, T-MATS assumes the compression process is isentropic and can be represented by a map that typically defines the relationship between corrected mass flow rate (*Wc*) and compressor *PR*, and contains the unique values for corrected shaft speed (*Nc*) and efficiency (*Eff*) at any given point. In the T-MATS compressor component, *PR*, *Wc*, and *Eff* are empirically calculated using *Nc* and R-line, an unique line on the compressor map relating *Wc* and *PR*. Once pressure and *Eff* have been determined, temperature can be calculated based on enthalpy and entropy tables. The power required for the compressor to operate is a function of the change in enthalpy across

the compressor (*Δht*) and mass flow (*W*) as seen in Eq. (4). The shaft torque required to maintain current shaft speed, which must be provided from a power generating component such as a turbine, is calculated as a function of power and shaft speed (*Nmech*) as seen in Eq. (5). In some compressors, air is removed (or bled) and used for turbine cooling. This cooling flow is modeled by removing associated mass flow and power from the core flow as dictated by where the bleed occurs.

$$Power = W * (\Delta ht) \tag{4}$$

$$Torque = \frac{Power}{Nmech} \tag{5}$$

Using compressor maps is an effective way of determining how a compressor will function when running in expected operating regions. Maps do not, however, give an accurate representation of conditions outside of these expected norms or indicate what may cause this operation. One such condition occurs when the *PR* builds too high for a given mass flow rate and the compression process fails, causing the potentially damaging phenomenon of compressor stall. In T-MATS, the stall line, or the line on a compressor map at which a stall will occur, is defined by a stall pressure ratio (*SPR*) that is a function of corrected flow into the compressor. The stall margin provides a measure of how close the compressor is to the stall condition through the percent difference between the current pressure ratio (*PR*) and the *SPR*, as shown in Eq. (6). It should be noted that if the compressor component crosses the stall line, stall margin will become negative but no other effects will be observable in the simulation since T-MATS does not contain a stall model.

$$\text{StallMargin}[\%] = \left(\frac{SPR-PR}{PR}\right) * 100 \tag{6}$$

In general, outputs of the compressor component can be determined from the thermodynamic state of the flow feeding into it from upstream. In order to determine the unique operating point, however, the R-line must be specified. This value can be determined from a mass flow imbalance resulting from comparison of the input mass flow and the mass flow interpolated from the compressor maps, as described above. Many turbo-machinery components built in T-MATS contain similar internal imbalances that may be reconciled by implementing a solver external to the components. These internal imbalances are generally flow imbalances (or errors) through components; in the steady-state case, the acceleration of the engine shaft(s), *Ndot*, are considered imbalances. Each T-MATS component that is prone to flow imbalance contains an output that depends on independent turbo-machinery inputs. A list of component independent and dependent variables is located in Table 2. When using a solver, the number of independents must match the number of dependents. Changes in mass flow rate, or *Ndot* in steady-state cases, have an effect on the entire system, therefore a one-to-one relationship between independent and dependent variables does not typically hold.

Table 2. Turbo-machinery independent and dependent variables.

T-MATS Component	Independent Variable	Dependent Variable
Ambient	Mass Flow Rate (*W*)	none
Compressor	R-line	Normalized Mass Flow Rate Error
Nozzle	none	Normalized Mass Flow Rate Error
Shaft	*Ndot* (SS only)	*NMech* (SS only)
Splitter	*BPR*	none
Turbine	Pressure Ratio (*PR*)	Normalized Flow Error

Central to modeling the operation of the compressor described above is the compressor map, which needs to be developed accurately in order to achieve realistic compressor performance. If generic maps are available, the iDesign tool can be used to automatically derive the map scale factors for compressors and turbines, and the throat areas of the nozzles, to meet particular engine requirements. Map scale factors are constants used to scale compressor and turbine maps to either normalize map inputs or to change the size of hardware components (for example, if a compressor for a simulation needs to be half the size of the compressor for which the map was created). The T-MATS iDesign feature calculates these map scale factors, along with nozzle throat areas, by fitting each component to the overall system model at a known steady-state, or design, point. Though generated at the design condition, scale factors created by this tool may be used to run the simulation in off-design conditions. (The technique of design-point analysis is beyond the scope of this document; for further information, see Ref. 1.) The

use of this type of map scaling is a prerogative of the user: some may use it often while others may prefer to develop maps that are tuned to the specific application. It is up to the designer to determine what is appropriate for a specific application.

4. Duct component

The duct component governs a transition between two gas turbine components. In T-MATS this is modeled as a simple fractional pressure loss.

5. Inlet component

An inlet acts to guide the flow to the front of the engine face and is modeled empirically as a drop in pressure based on total pressure relative to ambient pressure. This modeling scheme assumes changes in total temperature across the nozzle are negligible, which may not be true in some cases, such as at supersonic speed.

6. Nozzle component

The nozzle component acts to generate thrust for a gas turbine by forcing the output air through a narrow passage. Modeling the nozzle requires a combination of iterative solving, empirical lookups, and physic-based equations. The first step in determining the thrust is to iteratively solve (internal to the nozzle block) for the velocity, static pressure, and static temperature (*Ts*) of the flow at the throat of the nozzle by assuming Mach number (*MN*) is equal to 1 then using the relationships shown in Eqs. (7) and (8), where total and static enthalpy (*ht* and *hs* respectively) are determined by table lookup, assuming the expansion process is isentropic and specific heat ratio (γ) and the specific gas constant (*R*) are known.

$$velocity = \sqrt{2 * (ht - hs)} \tag{7}$$

$$MN = \frac{velocity}{\sqrt{\gamma * R * Ts}} \tag{8}$$

If the calculated static pressure is greater than the ambient pressure, the nozzle is assumed to be choked and the velocity, static pressure, and static temperature computed iteratively are accurate. However, if the calculated static pressure is less than ambient pressure, the nozzle is not choked and *MN*, static temperature, and velocity are recalculated based on ambient pressure.

Once velocity, static pressure (*Ps*), and static temperature (*Ts*) have been determined, the exit mass flow and thrust can be calculated as shown in Eqs. (9), (10), and (11), where *R* is the specific gas constant, ρ is density, *Ath* is throat area, *Pa* is ambient pressure, *g* is the acceleration of gravity, and *W* is mass flow.

$$\rho = \frac{Ps}{R * Ts} \tag{9}$$

$$W = \rho * velocity * Ath \tag{10}$$

$$thrust = \left(\frac{W}{g}\right) * velocity + (Ps - Pa) * Ath \tag{11}$$

7. Shaft component

The shaft component supplies a connection medium for turbo-machinery. Shaft acceleration (*Ndot*) is calculated as the sum of the input torques divided by the shaft inertia (*I)* as described in Eq. (12).

$$Ndot = \frac{\sum(Compoent\ Torques)}{2 * \pi * I} \tag{12}$$

8. Splitter component

The splitter component divides the flow into two separate streams (*W_branch1* and *W_branch2*) based on a *BPR*, as shown in Eqs. (13) and (14), and is typically used to model a multi-stream system, such as high bypass engines.

$$W\_branch2 = \frac{W * BPR}{(1 + BPR)} \tag{13}$$

$$W\_branch1 = \frac{W}{(1+BPR)} \tag{14}$$

9. *Turbine component*

The turbine component is designed to extract energy from a mass flow by directing air across a series of blades to spin a rotor or shaft. As with a compressor, the turbine is assumed to be isentropic and can be represented by a map that is typically defined as the relationship between *Wc* and turbine *PR*, and contains the unique values for *Nc* and *Eff* at any given point. In the T-MATS turbine component, *Nc* and turbine *PR* are used to empirically calculate *Wc* and *Eff*. Turbine temperature, enthalpy, torque, and power are calculated in the same manner as for the compressor, detailed in Eqs. (4) and (5), with the exception that power and torque are produced by the turbine then used to drive a compressor(s). In many cases, the mass flow entering a turbine is directly from a burner and has a temperature high enough to cause structural damage. For this reason, cooling flow is often added to turbine blades to create a protective layer of high pressure, lower temperature gas. Because the turbine is 0-D, additional gas cannot be added to the interior of the turbine component, so T-MATS models this cooling air as a combination of mass flows added either before or after the turbine.

10. *Valve component*

The valve component is a variable bypass orifice governed by a valve position that can open and close to increase or decrease bypass flow. Mass flow through the valve component is governed empirically by looking up mass flow values based on the pressure ratio across the valve. Valve position is taken into account with a flow multiplier that governs the opening and closing of the orifice. The model assumes that the valve exit stream is piped directly to a relatively large bypass stream and that changes in temperature and pressure of the bypass stream are negligible.

B. **T-MATS Solving method**

The T-MATS solvers are based around an iterative solver which operates with information defined by a Jacobian calculator. The iterative solver makes use of the Newton-Raphson (NR) method to step a plant toward a solution, a process described mathematically in Eq. (15), where *k* is the step iteration number.

$$x(k+1) = x(k) - \frac{f(x(k))}{J} \tag{15}$$

The Jacobian (*J*) is a linear map between the inputs (*x*) and outputs (*f(x)*) of a plant and is defined by perturbing each *x* from its initial conditions (*x*(0)) to find the effect on *f(x)*. A mathematical description of the Jacobian is given in Eq. (16). For non-linear systems, the Jacobian is only valid in a neighborhood local to the linearization point.

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{16}$$

These two equations, Eqs. (15) and (16), represent the two main steps of operation in a T-MATS solver. Initially, the Jacobian calculator component creates a linear representation of the plant at the current operating point. This requires perturbing slightly, in turn, each input from its initial condition, recording the results, and numerically determining the partial derivative of each output with respect to each input. The Jacobian is built from these calculations, inverted, and provided to a NR solver. In the second step, the NR solver steps toward a solution using the Jacobian developed in the first step. Because this solver method makes the assumption that the plant is locally linear, there is a chance the solver will not converge to a solution. To reduce the chances of non-convergence, it is important to have solver initial conditions and perturbation sizes set to values appropriate for the system.

In order to increase computational speed, the Jacobian calculator is not run every time step. A Jacobian is calculated at the beginning of the simulation and is not recalculated unless the NR solver has difficulty converging. The NR solver is configurable, giving users the capability of specifying the bounds of convergence, as well as the maximum number of iterations to try in a single time step.

In T-MATS, the method described above is used as the basis for a steady-state and a dynamic solver. When running dynamic simulations, the dynamic solver block can be utilized, in conjunction with a "while loop" as shown in Figure 1, to ensure convergence of a simulation at each time step. During testing, or when determining initial conditions for the system, it is sometimes advantageous to remove the time iteration and simply solve the system in steady-state. The steady-state solver does not require the "while loop" to operate, as it treats each time step as the solver iteration, an invalid model architecture if the simulation contains dynamic effects.

C. Effectors and Controls components

The effectors and controls components are intended for use in creating the "outer" loop hardware and controller algorithm models, providing a starting point for control system simulation. Included are the following blocks: a proportional integral (PI) controller, a fuel actuator model, and a sensor model. The sensor and actuator models consist of a first order lag with the ability to add Gaussian noise, faults, and other nonlinearities. The default PI controller is a simplified set-point regulator that is designed as a template that encourages controller expansion.

D. Model Development Tools

In addition to the block sets, T-MATS includes a set of tools, accessible from the Simulink menu bar, that make model creation easier and faster. Discussing these tools in detail is beyond the scope of this paper, however a list of the each tool and what it does has been included in Appendix B.

IV. Turbojet Example

Using maps, map scale factors, and other modeling constants taken from a public source NPSS model,[6] a turbojet plant model was created using T-MATS in order to validate the component models, and to serve as an example that can be distributed with the software. For this example, the entire simulation was created first in steady-state, to verify a model match with an NPSS "truth" model, and then turned into a dynamic model for time-dependent simulation. The plant model contains a compressor, burner, turbine, duct, and nozzle as well as an ambient component connected as shown in Figure 2, where the station numbers are indicated along the bottom. The compressor and turbine are connected via a mechanical shaft that drives the compressor using power from the turbine.

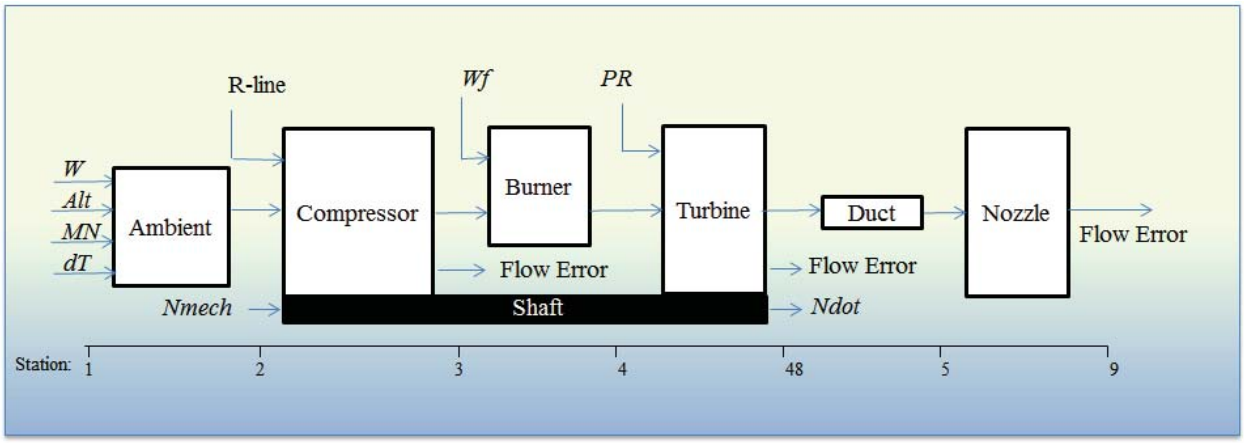

Figure 2. Turbojet example problem block diagram.

The inputs and outputs of the turbojet plant connect with either the iterative solver (in the "inner" loop) or with integrators (in the "outer" loop). Solver independent and dependent variables as well as plant input variables, including those connected to an "outer" loop integrator, are summarized in Table 3. In the steady-state simulation, input fuel flow (*Wf*) is set to the design point dictated by the NPSS data. The fuel flow is allowed to vary with time in the dynamic simulation. For more information on model implementation with T-MATS, see Ref. 6.

Table 3. Plant input and output connections.

Simulation type	**Independent variables**			
Steady-State	*W*	R-line	*PR*	*Ndot*
Dynamic				None
	Dependent variables			
Steady-State	Compressor Flow Error	Turbine Flow Error	Nozzle Flow Error	*Nmech*
Dynamic				None
	Integration variable			
Steady-State	None			
Dynamic Only	*Nmech* from *Ndot*			
	Other inputs			
Steady-State	Wf	*Alt*	*MN*	*dT*
Dynamic				

A. Steady-State Model

Accuracy of the turbojet plant model was investigated by comparing the steady-state model data with the equivalent NPSS data. The percent difference between T-MATS and the NPSS "truth" model can be viewed in Table 4. Steady-state data were gathered for the following parameters at the stations immediately after each component, as well as at the compressor input: flow (*W*), total temperature (*Tt*), total pressure (*Pt*), ambient pressure (*Pa*), shaft torque (*Trq*), shaft speed (*N*), fuel to air ratio (FAR), and gross thrust (*Fg*). If a parameter was not calculated at a particular station, e.g., shaft torque by the burner, it was neglected. In all cases, the difference between the NPSS model and the T-MATS model was less than 0.5%. The largest deviations were in station 4 temperature and station 5 pressure calculations. The difference in thrust was 0.15% and engine flow differences were all negligible. Steady-state differences' of these magnitudes demonstrate component-level and system-level modeling in T-MATS that is on par with the NPSS turbojet example model and should be acceptable in similar gas turbine applications.

Table 4. Turbojet performance comparison.

Station	Parameter	T-MATS	NPSS	% Difference
Station 2 Input	*W* (pps)	99.97	100.0	0.03%
	Tt (degR)	518.67	518.67	0.0%
	Pt (psia)	14.40	14.40	0.0%
	Pa (psia)	14.70	14.70	0.0%
Station 3 Compressor	*W* (pps)	99.97	100.0	0.03%
	Tt (degR)	1317.0	1315.80	0.0912%
	Pt (psia)	288.60	288.0	0.2083%
	Trq (lbf*ft)	14660.0	14650.0	0.0683%
Station 4 Burner	*W* (pps)	103.0	103.0	0.0%
	Tt (degR)	3079.0	3065.0	0.4568%
	Pt (psia)	274.10	273.64	0.1681%
	FAR	0.03	0.03	0.0%
Station 48 Turbine	*W* (pps)	103.0	103.0	0.0%
	Tt (degR)	2454.0	2449.0	0.2042%
	Pt (psia)	89.94	89.60	0.3795%
	FAR	0.30	0.30	0.0%
	Trq (lbf*ft)	14660.0	14650.0	0.0683%
Station 5 Duct	*W* (pps)	103.0	103.0	0.0%
	Pt (psia)	89.04	88.70	0.3833%
	FAR	0.03	0.03	0.0%
Station 9 Nozzle	*W* (pps)	103.0	103.0	0.0%
	Fg (lbf)	10670.0	10654.0	0.1502%
Shaft	*N* (rpm)	9999.79	10000.0	0.0021%

B. Dynamic model

Once the plant model accuracy had been verified, the simulation was made dynamic by placing the engine plant and a T-MATS dynamic solver inside a "while loop." The "outer" loop was created by adding an integrator, to calculate the shaft speed, and a shaft speed control system, to produce a fuel demand. This setup can be represented by the block diagram in Fig. 1, where the outer loop plant contains the shaft speed control system as well as the shaft speed integrator and the inner loop plant contains the engine plant model and solver.

The plots in Figure 3 through Figure 5 were created from simulation of the dynamic turbojet model subjected to a thrust "chop", a fast reduction in shaft speed demand (*Ndmd*), at 15 s into the simulation, decreasing *Ndmd* from 10,000 rpm to 8,000 rpm in 0.001 s. A PI controller is used to determine the fuel command from the error between *Ndmd* and a sensed shaft speed. Results in Figure 3 show that, as *Ndmd* is dropped, fuel flow is reduced and the shaft speed tracks the decrease in *Ndmd*, resulting in lower gross thrust. While the final drop in shaft speed was only 20% both the fuel and gross thrust were reduced by about 80%, which is typical for gas turbine engines. Gross thrust fall time can be observed to be around 1 s with an undershoot of about 1% the initial values and a settling time of about 2 s. For a small engine, such as the example turbofan, these types of responses are typical.

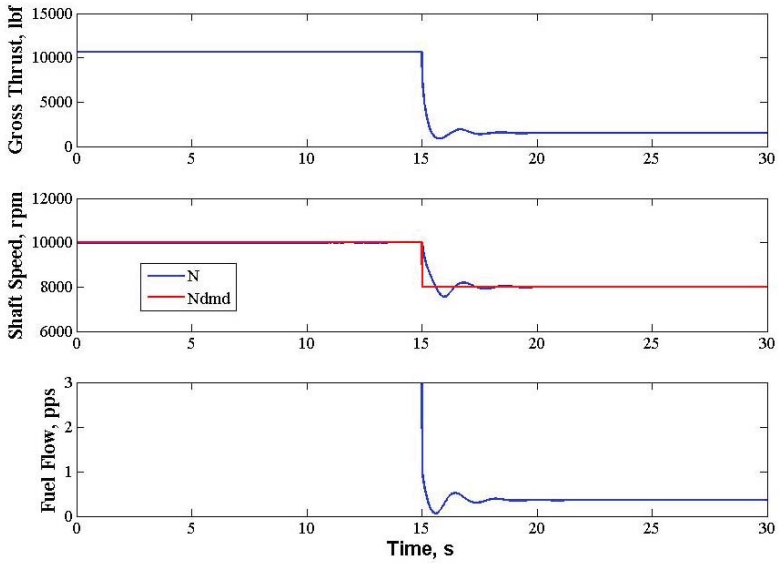

Figure 3. Dynamic turbojet transient simulation output.

Results in Figure 4 show temperatures and pressures at key stations during the transient. Pressures after the compressor and after the turbine, *Pt3* and *Pt48* respectively, drop about 70% from their initial values, with transient fall times, and settling times, comparable to those observed in the fuel flow transient (1 s and 2 s, respectively). Pressure undershoot was observed to be slightly larger after the compressor, 8%, rather than after the turbine, 5%. Temperatures, *Tt3* and *Tt48*, show drops of 70% and 50%, respectively, and fall times that are consistent with the pressure transients (about 1 s). Undershoot and settling for *Tt3* (8% and 2 s) are significantly less than for *Tt48* (30% and 4 s). This difference in temperature transient response can be attributed to a delay in the energy balance along the shaft.

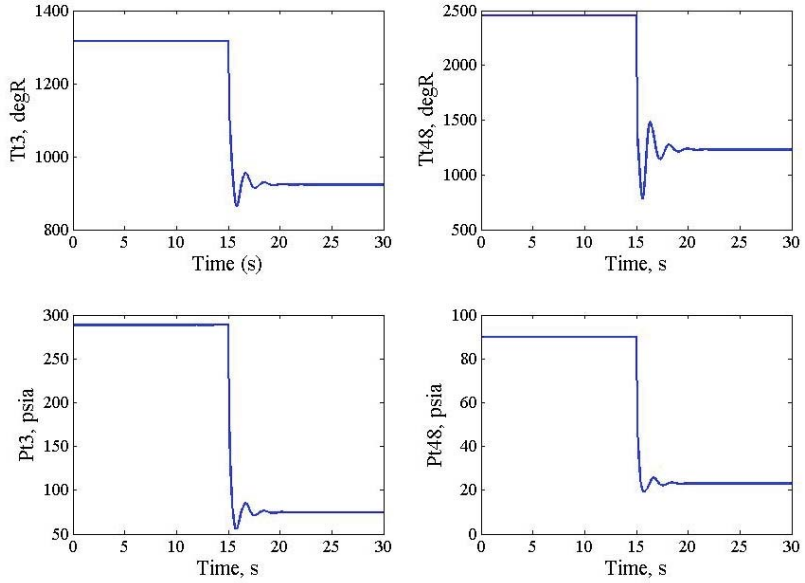

Figure 4. Dynamic turbojet thermodynamic dimension outputs.

The plot in Figure 5 shows the transient compressor operating points superimposed over the compressor map. The pressure ratio is initially nearly 20, and then quickly drops to 15 following the set point change, progressing to around 5 as the system reaches steady-state, which is a typical trend for a thrust chop event. Initially stall margin is low, and then grows quickly as fuel flow is cut off sharply. As fuel stabilizes, the stall margin moves back up to about the same value it was before the transient began. It can be seen that in all cases the stall line remains above the

operating points, suggesting stall free operation. Transient performance, in this example, is dependent on PI tuning and is meant only to demonstrate the dynamic capability of the model.

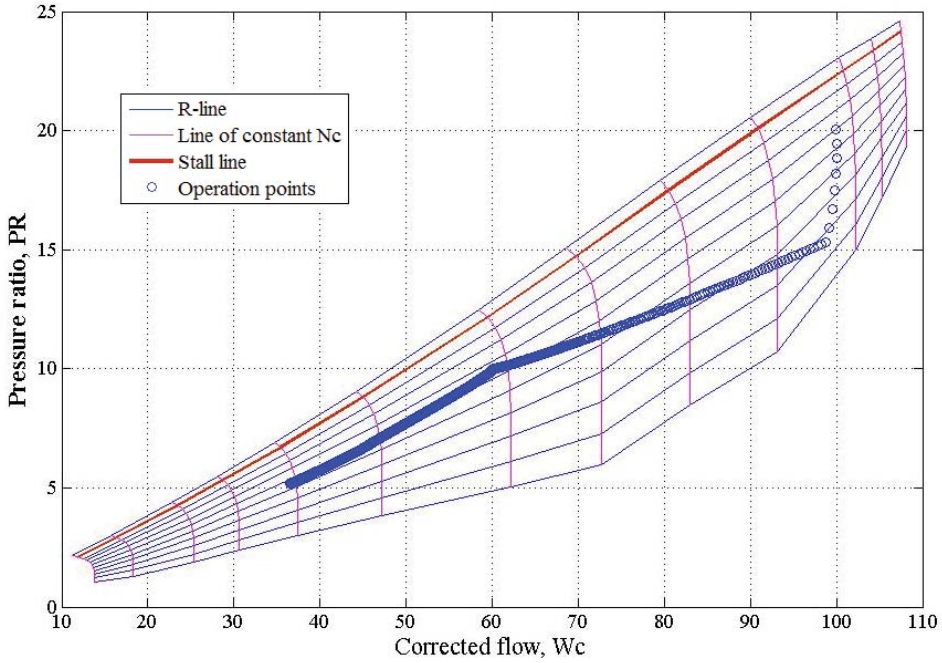

Figure 5. Compressor transient operating line for dynamic turbojet simulation.

V. Summary

The Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) software package is an open source MATLAB/Simulink toolbox (plug-in) for the development of custom thermodynamic and controls models that allows for the building of an entire system simulation in a single computational environment. This paper presents an introduction to the T-MATS software package, describing possible propulsion simulation architectures, detailing the theory behind the Simulink block sets included in the package, and showing an example of a steady-state and a dynamic turbojet engine simulation created with T-MATS. A model matching was performed by comparing the steady-state T-MATS simulation against a well-documented NPSS steady-state engine model. Results from this model matching show comparable performance between the two simulations and that T-MATS is a simulation environment that is capable of modeling turbo-machinery. Dynamic simulation results show transients that are plausible and realistic for a small turbofan engine. Additional information on the capabilities and usage of T-MATS can be found in locations described in Appendix B and Ref. 6.

Acknowledgments

The authors would like to thank the NASA Aviation Safety Program's Vehicle Systems Safety Technologies Project for funding this work. The assistance of the OpenMDAO team was critical to enabling the authors to release this tool under an open-source license.

Appendix

A. T-MATS Tool Description

The T-MATS Model Development Tools, accessible from the Simulink menu bar, include the ability to toggle between Simulink GoTo and From blocks, turning the T-MATS iDesign feature on or off for all blocks in a model, and automatically creating GoTo and From blocks for each of the inputs and outputs of a subsystem, as shown in Table 5.

Table 5. Description of T-MATS tools.

T-MATS Tool	Description
GF_Convert	Toggles between GoTo and From blocks, without changing the block name. It also works to toggle between Inport and Outport blocks.
iDesign_On	Turns iDesign on for every block within the model.
iDesign_Off	Turns iDesign off for every block within the model.
Block Link Setup	Creates Goto or From blocks for every input and output of a subsystem. For turbo-machinery blocks the tool creates linkages appropriate for T-MATS connections, which allow blocks to be connected quickly and easily.

B. T-MATS Download information

The T-MATS software can be downloaded from the Github® repository at https://github.com/nasa/T-MATS/releases, where lists of bugs and feature requests are also maintained. A user group has been created at https://groups.google.com/forum/#!forum/t-mats-user-group to enable collaboration between users and developers. It should be noted that although T-MATS was initially developed by NASA, T-MATS is open source and, as such, collaboration with the user community in finding bugs, adding features, and generally working toward a better product is strongly encouraged.

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] Nichols, L.D. and Chamis, C.C., "Numerical Propulsion System Simulation: An Interdisciplinary Approach," AIAA-91-3554, September 1991.

[3] Gasturb, Software package, Ver. 12, GasTurb GmbH, Aachen, Germany, 2014.

[4] Thermolib, Software package, Ver. 5.2, EUTech, Aachen, Germany, 2014.

[5] 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," AIAA-2010-6630, 46th AIAA Joint Propulsion Conference & Exhibit, Nashville, TN, July, 2010.

[6] Chapman, J.W., Lavelle, T.M., May, R.D., Litt, J.S., Guo, T.H., "Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) User's Guide," NASA/TM-2014-216638, January 2014.

[7] Jones, S.M., "Steady-State Modeling of Gas Turbine Engines Using The Numerical Propulsion System Simulation Code," GT2010-22350, June 2010.