

Article

# State Vector Identification of Hybrid Model of a Gas Turbine by Real-Time Kalman Filter

Gustavo Delgado-Reyes <sup>1</sup>, Pedro Guevara-Lopez <sup>1</sup>, Igor Loboda <sup>1</sup>,  
Leobardo Hernandez-Gonzalez <sup>1,\*</sup> , Jazmin Ramirez-Hernandez <sup>1</sup>,  
Jorge-Salvador Valdez-Martinez <sup>2</sup>  and Asdrubal Lopez-Chau <sup>3</sup>

<sup>1</sup> Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Culhuacan, Instituto Politécnico Nacional, Av. Santa Ana No. 1000, Col. San Francisco Culhuacan, Mexico City C.P. 04430, Mexico; gdelgador@outlook.com (G.D.-R.); pguevara@ipn.mx (P.G.-L.); iloboda@ipn.mx (I.L.); jazzrh@hotmail.com (J.R.-H.)

<sup>2</sup> Universidad Tecnológica Emiliano Zapata del Estado de Morelos, Universidad Tecnológica No. 1, Morelos 62760, Mexico; jorgevaldez@utez.edu.mx

<sup>3</sup> Universidad Autónoma del Estado de México, CU UAEM Zumpango Kilómetro 3.5 Camino Viejo a Jilotzingo, Estado de Mexico 55600, Mexico; alchau@uaemex.mx

\* Correspondence: lhernandezg@ipn.mx; Tel.: +52-1-55-4488-3525

Received: 12 March 2020; Accepted: 22 April 2020; Published: 27 April 2020



**Abstract:** A model and real-time simulation of a gas turbine engine (GTE) by real-time tasks (RTT) is presented. A Kalman filter is applied to perform the state vector identification of the GTE model. The obtained algorithms are recursive and multivariable; for this reason, ANSI C libraries have been developed for (a) use of matrices and vectors, (b) dynamic memory management, (c) simulation of state-space systems, (d) approximation of systems using equations in matrix finite difference, (e) computing the mean square errors vector, and (f) state vector identification of dynamic systems through digital Kalman filter. Simulations were performed in a Single Board Computer (SBC) Raspberry Pi 2<sup>®</sup> with a real-time operating system. Execution times have been measured to justify the real-time simulation. To validate the results, multiple time plots are analyzed to verify the quality and convergence time of the mean square error obtained.

**Keywords:** gas turbine model; Kalman filter; real-time; identification; single board computer; time constraints

## 1. Introduction

A digital filter is a hardware or software used to reduce the noise in the data of a system, extract information, or predict and rebuild the behavior of a system [1]. In [1,2], two basic areas in digital filtering theory are presented: estimation and identification. Depending on the time evolution interval  $k$ , in which the digital filter works, three filter actions are defined: reconstruction, filtering and prediction.

A Real-time System (RTS) interacts with a known dynamic environment, during a certain instant of occurrence  $\mu_k$  of the physical world, and the system output depends on the input through a processing intrinsic time  $\varepsilon$  [3], and it must be completed into a sampling time of the dynamic process [4]. The interaction between RTS and dynamic processes takes place through the detection of the system variables with the use of transducers like semiconductor devices. The sensed signals may be affected by internal and external noise causing measurement errors. Filtering techniques can be added in the RTS, integrating an architecture that can be used in industrial processes interactions, according to [5,6]. One of the most widely used dynamic filters that performs estimation and identification is the Kalman filter (KF); its digital version is described as an adaptive iterative algorithm designed to identify the state vector of state space linear systems. The real-time implementation of this filter

requires an analysis of the time constraints imposed by the system. The correct characterization of these restrictions in addition with the recursive form of the algorithm enable the real-time simulation.

In the last decades, the application of gas turbines has demonstrated a significant growth [7]. These engines are principal drivers for electricity production and are widely used in oil platforms, chemical plants, gas stations, and so on. Air transport is another large area of application, where a turbofan is a main engine type. Being relatively light and compact, aircraft gas turbine engines provide high power and efficiency. To have the best performances, the engine must operate near its limits at all operating modes. To this end, an engine control must be accurate. The enhancement of an engine control system makes control laws more complex, and mathematical modeling becomes increasingly useful in the development of such a system [8]. Under contradictive demands of high engine performances, reliability, safety, and low maintenance costs, the use of developed diagnostic systems becomes an effective strategy. The development and operation of these systems are also based on different mathematical models [9,10]. Nonlinear dynamic physics-based engine models are too complex and slow to be used in on-board control and diagnostic systems. Instead, linear dynamic models and Kalman filtering operating in real time are widespread [8,9].

In this paper, a real-time implementation on a Single Board Computer (SBC) of a Kalman digital filter applied to the hybrid dynamic model of a turbofan engine is proposed. The objective is to closely examine a real-time operation of the filter. The execution time is verified at each time instance and under the conditions of a specialized computer similar to an engine control processor. This examination allows us to draw a sound conclusion whether the filter and model can be implemented in real on-board engine and aircraft systems. To our knowledge, such a strict verification is novel.

The mathematical model corresponds to a linear discrete time stochastic dynamic system expressed in a state-space form. For the experimental results, a Raspberry Pi 2 Model B<sup>®</sup> was used. This computer has a quad-core ARM Cortex-A7 CPU at 900 MHz, graphics processor Video-Core IV and 1 GB RAM. A strong contribution is a real-time implementation details of the digital Kalman filter; including the development of a library in C language for RT-Raspbian<sup>®</sup>.

The rest of this paper is organized as follows. Section 2 presents the gas turbine engine models. Section 3 shows the hybrid dynamic model for a gas turbine. In Section 4, a real-time simulation of a gas turbine engine is introduced. Section 5 presents the state vector identification using an embedded version of the Kalman filter. In Section 6, a real-time response analysis is presented. Finally, the conclusions are provided.

## 2. Gas Turbine Engine Models

### 2.1. Analyzed Engine

The engine simulated and analyzed in the present paper is a turbofan for a passenger airplane with two ducts (gas paths), as shown in Figure 1. In the first duct (internal), input air is compressed by a fan, booster, and a high-pressure compressor (HPC). The compressed air goes to a combustor where liquid fuel is burnt, producing a great volume of high temperature gas. The combustor converts chemical energy of the fuel into kinetic energy of the high-temperature gas. A high-pressure turbine (HPT) drives the HPC, while a low-pressure turbine (LPT) rotates the fan. In the second duct (external), air is compressed by the fan, mixed with the LPT exhaust gas, and then extended in a common nozzle, thus producing engine thrust. The turbofan scheme in Figure 1 also shows the points of cooling air intakes and supplies. Additionally, the scheme includes simulation stations between the components at which engine gas path variables are computed in the thermodynamic model described in the next subsection. A part of these simulated variables that are employed in the present paper is shown in Table 1.

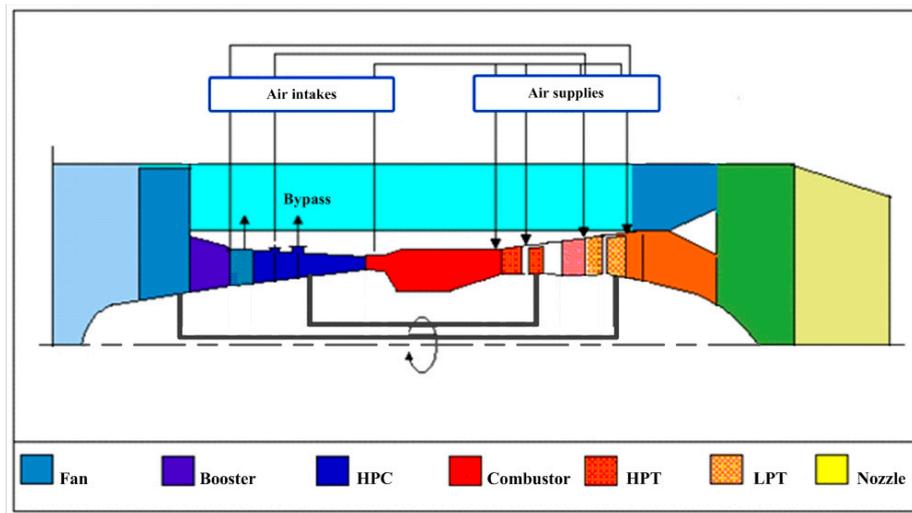


Figure 1. Turbofan scheme.

Table 1. Simulated turbofan variables.

Parameter	Meaning
<b>State Variables</b>	
$R$ [N]	Engine thrust
$\pi_c = P_{HPC}/P_{in}$	Total compressor pressure ratio
$P_{in}$ [N/m <sup>2</sup> ]	Compressor inlet pressure
$P_{HPC}$ [N/m <sup>2</sup> ]	HPC outlet pressure
<b>Control Variables U</b>	
$u_s = \pi_c$	Compressor pressure ratio
$U = G_f$ [Kg/h]	Dynamic control variable-fuel flow
<b>State space Variables X</b>	
$n_l$ [rpm]	Low pressure rotor speed
$n_h$ [rpm]	High pressure rotor speed
<b>Monitored Variables Y</b>	
$P_{HPC}$ [N/m <sup>2</sup> ]	HPC outlet pressure
$T_{CC}$ [K]	Compressor outlet temperature.
$T_{LPT}$ [K]	Low pressure turbine outlet temperature

The thermodynamic model of the analyzed turbofan engine was developed according to the engine structure in Figure 1 and the principles described below. This model consists of nonlinear static and dynamic models.

### 2.2. Nonlinear Static Model

One of the first descriptions of a gas turbine thermodynamic models can be found in [11]. They are also well described in [12]. A part of the thermodynamic model, nonlinear static model, is the principal physics-based model in the field of gas turbine control and diagnostics. This model can be used to describe engine faults [13]. Additionally, it is an integral part of diagnostic algorithms based on system identification such as the algorithm described in [14].

The developed model uses gas flow thermodynamic relations and performance maps of engine components to compute a vector  $\mathbf{Y} \in \mathbb{R}^{m \times 1}$  of gas path variables as a function of a vector  $\mathbf{U} \in \mathbb{R}^{n \times 1}$   $\mathbf{U}$  of steady operational conditions (control variables and ambient conditions) as well as a vector  $\mathbf{\Theta} \in \mathbb{R}^{r \times 1}$  of fault parameters, which can also be named health parameters or correction factors depending on the application. Usually two faults parameters are applied to describe the faults of one engine component, and a total number  $r$  of these parameters equals two multiplied by a components number. Although the model computes hundreds of engine variables, in the present study only measured variables are

considered to implement the developed algorithms in real control and diagnostic systems. A total number of measured variables equals  $m + n$  from which  $n$  measured variables including compressor pressure ratio and fuel flow are included in the vector  $\mathbf{U}$ . The other  $m$  measured variables such as high compressor pressure, turbine temperature, combustion chamber temperature and engine thrust, constitute the vector  $\mathbf{Y}$ . The variables  $\mathbf{Y}$  are sensitive to engine faults and are used to diagnose the engine, and therefore, are called monitored variables. Thus, the nonlinear static model relates the monitored variables to the operating conditions and fault parameters and has the following structure:

$$\mathbf{Y} = F(\mathbf{U}, \Theta) \tag{1}$$

Mathematically, the nonlinear static model is formed as a system of algebraic Equations reflecting mass and energy balance between engine components during steady-state operation. Expression (1) is obtained through the system solution.

This model also allows computing other engine models [15]. In particular, it simplifies the development of the nonlinear dynamic model described below.

### 2.3. Nonlinear Dynamic Model

Studies [13,16] demonstrate the growing interest in the diagnosis during gas turbine dynamic operation, making necessary the analysis of an engine dynamic model. In contrast with the static model of Equation (1), in the thermodynamic model describing engine transient operation (nonlinear dynamic model), a time variable  $t$  is added in the function arguments, and the vector  $\mathbf{U}$  is expressed as a function of time:

$$\mathbf{Y} = F[\mathbf{U}(t), \Theta, t], \tag{2}$$

Monitored variables  $\mathbf{Y}$  of Equation (2) are computed numerically as a solution of the differential equations system describing a transient response. In the dynamic nonlinear model, these differential equations differ a little from the static model equations, and the numeric procedure of the solution is practically the same. The nonlinear static and dynamic models have many common modules and constitute a united program complex of the thermodynamic model.

The developed turbofan thermodynamic model has been verified using the model of the turbofan manufacture as a pattern. It was found that the maximum difference was smaller than 1%. Table 2 helps to see better the closeness of these models by specifying their values for the variables of interest enlisted in Table 1. Comparing the values, one can see that the differences are about 0.1%, and the models are practically similar. Thus, we can state that the developed model has sufficient accuracy.

Table 2. Thermodynamic model validation data.

Compared Systems	$R$	$\pi_c$	$G_f$	$P_{HPC}$	$n_l$	$n_h$	$T_{CC}$	$T_{LPT}$
Thermodynamic model	37,790.0	15.587	1506.5	1,579,400.0	7348.1	17,943.0	1436.2	819.57
Manufacturer' model	37,766.5	15.587	1505.2	1,579,300.0	7344.9	17,950.0	1435.3	820.87

Table 3 illustrates the adequacy of the thermodynamic model with respect to a real engine. We can note that the discrepancies have risen compared to Table 2: An average difference is 0.019, and a maximum difference is 0.049 for the air consumption  $G_{air}$ . However, these differences are not too great and have a natural explanation: After the model creation and validation, the engine was further modernized, but without significant changes in engine construction. Given the above explanations, we can conclude that the thermodynamic model correctly describes a general behavior of the turbofan at steady states and transients. In this way, the results obtained in the present paper on the basis of this model will be valid for a real engine.

**Table 3.** Simulated and real engine performances.

Compared Systems	R	$\pi_c$	SFC	$G_{air}$	$m$	$T_{CC}$
Thermodynamic model	36,000.0	15.490	0.4160	132.07	4.857	1429.8
Engine	36,840.0	15.490	0.4100	125.85	4.940	1450.0

The thermodynamic model is a complex software that includes more than 70 program modules and about 2000 lines of a source text. This, as shown in Section 6, makes the model too slow for real-time application. For this reason, we will employ simplified models.

#### 2.4. Static Baseline Model

The present study is focused on the engine dynamics simulation, and the engine degradation and the faults are not considered, therefore, the function  $\mathbf{Y} = F(\mathbf{U}, \Theta)$  of Equation (1) can be written as follows:

$$\mathbf{Y} = F(\mathbf{U}) \tag{3}$$

This relation is called a baseline model. For on-line applications, a simple model is usually conformed by polynomials or neural networks [12]. For one monitored variable as a function of four operating conditions, the used polynomial is written as:

$$\begin{aligned} Y(\mathbf{U}) = a_0 + a_1u_1 + a_2u_2 + a_3u_3 + a_4u_4 + a_5u_1u_2 + a_6u_1u_3 + a_7u_1u_4 + a_8u_2u_3 \\ + a_9u_2u_4 + a_{10}u_3u_4 + a_{11}u_1^2 + a_{12}u_2^2 + a_{13}u_3^2 + a_{14}u_4^2 \end{aligned} \tag{4}$$

To determine the coefficients  $a_i$ , the least squares method is applied, and the necessary input data  $\mathbf{Y}$  and  $\mathbf{U}$  were obtained through the thermodynamic model. To have accurate estimates  $a_i$ , a large volume of these data uniformly distributed in the space  $\mathbf{U}$  was generated.

#### 2.5. Linear Dynamic Model

Linear state space dynamic models are widely used in real-time applications. They can be given by the two following Equations:

$$\dot{\mathbf{X}}(t) = \mathbf{A}\Delta\mathbf{X}(t) + \mathbf{B}\Delta\mathbf{U}(t), \tag{5}$$

$$\Delta\mathbf{Y}(t) = \mathbf{C}\Delta\mathbf{X}(t) + \mathbf{D}\Delta\mathbf{U}(t), \tag{6}$$

where  $\mathbf{X}$  is a vector of state space variables, and  $\Delta$  means a difference between dynamic and static values. For the analyzed gas turbine, the vector  $\mathbf{X}$  includes rotation speeds of low and high-pressure spools; unknown matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are determined using the thermodynamic model of Equation (1).

### 3. Hybrid Dynamic Model

To obtain dynamic values of gas path variables in a simplified hybrid dynamic model, the dynamic corrections of Equations (5) and (6) are added to the static model values:

$$\mathbf{U} = \mathbf{U}_{st} + \Delta\mathbf{U}, \tag{7}$$

$$\mathbf{X} = \mathbf{X}_{st} + \Delta\mathbf{X}, \tag{8}$$

$$\mathbf{Y} = \mathbf{Y}_{st} + \Delta\mathbf{Y}, \tag{9}$$

Table 1 specifies the control, state space, and monitored variables. The simulation was performed at standard ambient conditions, therefore, the static and the linear dynamic models have only one operating condition that is the control variable (power set). A pressure ratio is chosen as a static control variable  $u_s$ , and a fuel flow is a dynamic control variable  $u$ .

The static performance depends on the power level; a piecewise approximation is used to determine the unknown values in Equations (3), (5), and (6). According to the control variable (power set),  $u_s = \pi_c$ , the interval of engine operation is divided into five zones  $i_z = 1, 2 \dots 5$ , where the last zone corresponds to the highest power. This division is related to the action of bypass valves, which causes the shifts of engine performances. These zones are also the result of the trade-off between approximation accuracy and simplicity. The static quantities  $u_{st}$ ,  $\mathbf{X}_{st}$ , and  $\mathbf{Y}_{st}$  are presented as polynomial functions of the variable  $\pi_c$ . The matrixes  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are computed using the static nonlinear model, see Equation (1). In total, the approximation can be given by the system:

$$i_z = f(u_s), \tag{10}$$

$$u_{st}, \mathbf{X}_{st}, \mathbf{Y}_{st} = f(i_z, u_s), \tag{11}$$

$$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} = f(i_z), \tag{12}$$

Variables are computed for times  $t = t_1, t_2, t_3, \dots t_k, t_{k+1}, \dots t_n$ , determined with a constant time interval  $\Delta t = t_{k+1} - t_k$ . The transient is set according to a control variable time profile  $u = f(t)$ . The simulation is defined in 10 steps:

Step 1. Current time  $t_k$  is given by:

$$t_k = t_{k-1} + \Delta t, \tag{13}$$

Step 2. Using a time profile  $u = f(t)$ , the dynamic control variable is determined by:

$$u(k) = f(t_k), \tag{14}$$

Step 3. The static control variable  $u_s(k)$  is known from the previous time point  $k-1$ .

Step 4. According to Equation (10), the zone number is obtained:

$$i_z(k) = f[u_s(k)], \tag{15}$$

Step 5. Equation (11) is given by the following static values:

$$u_{st}(k), \mathbf{X}_{st}(k), \mathbf{Y}_{st}(k) = f[i_z(k), u_s(k)], \tag{16}$$

Step 6. The differences are calculated according to:

$$\Delta u(k) = u(k) - u_{st}(k), \Delta \mathbf{X}(k) = \mathbf{X}(k) - \mathbf{X}_{st}(k). \tag{17}$$

The dynamic and static values are calculated. Note that the state space variable dynamic values  $\mathbf{X}(k)$  are known from the previous instant  $k - 1$ .

Step 7. According to Equation (12), the following matrixes are chosen:

$$\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k), \mathbf{D}(k) = f[i_z(k)], \tag{18}$$

Step 8. Placing the matrixes of Equation (18) in (5), the state space derivatives are determined by:

$$\dot{\mathbf{X}}(k) = \mathbf{A}(k)\Delta \mathbf{X}(k) + \mathbf{B}(k)\Delta u(k), \tag{19}$$

and the changes of monitored variables are defined according to:

$$\mathbf{Y}(k) = \mathbf{C}(k)\Delta \mathbf{X}(k) + \mathbf{D}(k)\Delta u(k), \tag{20}$$

Step 9. The knowledge of the derivatives allows us to compute dynamic values of state space variables for the next point  $k + 1$ .

$$\mathbf{X}(k + 1) = \mathbf{X}(k) + \dot{\mathbf{X}}(k)\Delta t, \tag{21}$$

Step 10. Dynamic values of monitored variables are calculated as:

$$\mathbf{Y}(k) = \mathbf{Y}_{st}(k) + \Delta\mathbf{Y}(k). \tag{22}$$

The first monitored variable is the pressure  $P_{HPC}$ , and its value  $Y_1(k)$  is used for the next time to calculate  $\pi_c$  as a static control variable  $u_s(k + 1)$ . Therefore, for each iteration  $k$  and time  $t_k$ , the 10 steps described above are executed to determine the dynamic control variable  $u(k)$  and the dynamic values  $\mathbf{Y}(k)$ . The iteration  $k$  also provides the values  $\mathbf{X}(k + 1)$  and  $u_s(k + 1)$  used as input data in the next iteration. In this way, continuous dynamic gas turbine simulation in the interval  $(t_1 - t_n)$  is achieved. The scheme in Figure 2 illustrates this simulation.

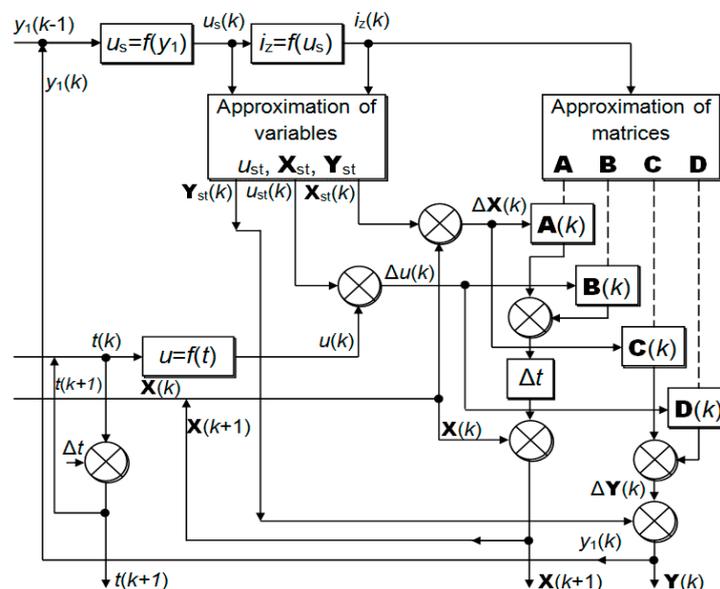


Figure 2. Block diagram of the hybrid dynamic model operation.

#### 4. Real-Time Simulation of GTE

The hybrid model was programmed in the SBC Raspberry pi 2 model B<sup>®</sup> with RT-Raspbian and C language, [17]; some libraries have been developed to provide necessary tools for (a) the use of matrices and vectors, (b) dynamic memory management, (c) simulation of state-space systems, and (d) approximation of systems by equations in matrix finite differences. The gas turbine engine operation was simulated during 10 s with a time step  $\Delta t = 50$  ms, resulting in 200 real-time instances. The fuel consumption was linearly increased from an idle regime value at  $t = 1$  s to a take-off regime value at  $t = 2$  s. The resulting time plots of static and dynamic values of the variables and dynamic changes can be seen in Figures 3–5.

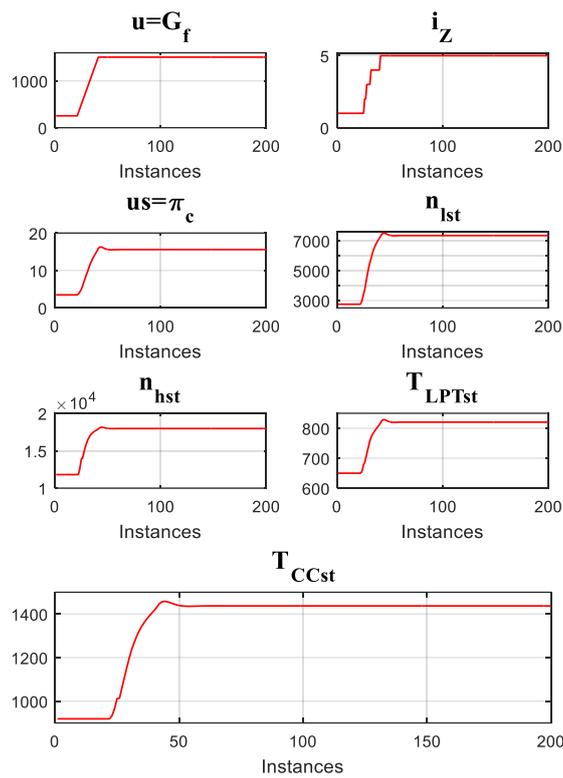


Figure 3. Simulation of static values (static model within the hybrid model).

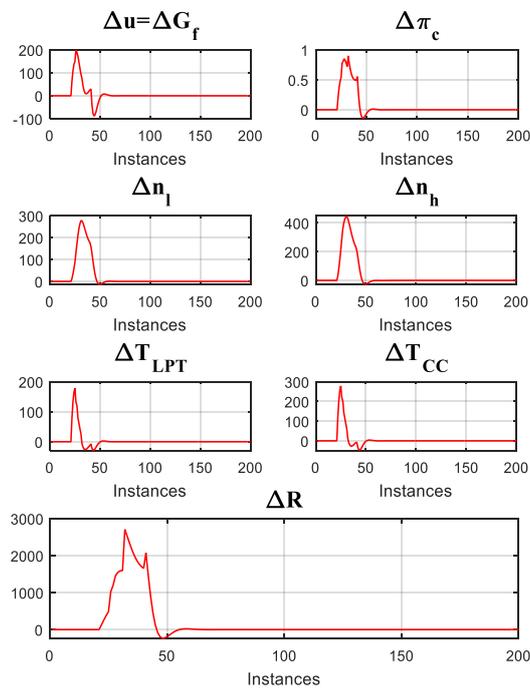


Figure 4. Simulation of dynamic deviations. (linear dynamic model within the hybrid model).

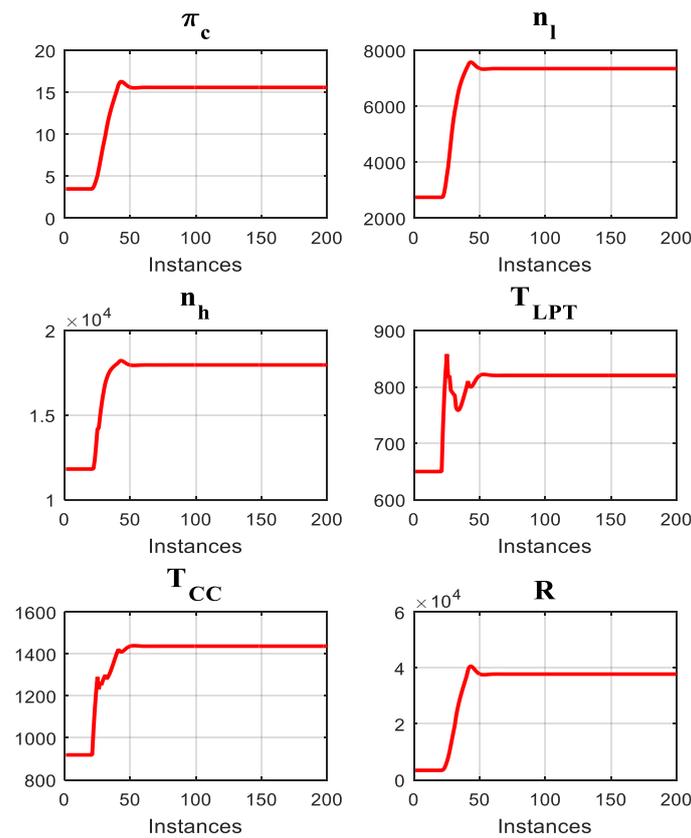


Figure 5. Simulation of dynamic values (hybrid dynamic model).

In Figure 3, the plot  $u = G_f(k)$  shows the lineal behavior of the control variable. The fuel consumption increases, causing the increment in the zone number, as is shown in plot  $i_z(k)$ .

The control variable  $u_s$  is set as a dynamic value of  $\pi_c$  from the previous instance, therefore,  $u_s$  changes dynamically after the instance 40 generating a transient response. In Equation (10), the static variables  $X_{st}$  and  $Y_{st}$  depend on  $\pi_c$ , therefore, their behavior should be similar to  $\pi_c(k)$ . This is shown in Figure 3 with the plots of  $n_{lst}$ ,  $n_{hst}$ ,  $P_{HPCst}$ ,  $T_{LPTst}$ , and  $T_{CCst}$ .

To verify the dynamic behavior of the simulated engine, the plots of the dynamic deviations (changes) are presented in Figure 4. The overshoots that appear at the point  $t = 1$  s (instance 20) are greater than the ones after  $t = 2$  s (instance 40), then stabilize approximately at the point  $t = 3$  s (instance 60).

Figure 5 integrates the static values and dynamic deviations resulting in dynamic values of gas turbine variables. These dynamic values were compared with the outputs of the dynamic nonlinear model, Equation (2), and its accuracy was confirmed. Thus, Figures 3–5 validate the hybrid model.

### 5. State Vector Identification by Kalman Filter

The hybrid model described in Section 3 and the variables simulated with this model in Section 4 do not consider internal or external noise. Since in a real system (gas turbine), the measurements contain noise, the computed variables will have errors. To reduce this noise and minimize the random estimation errors, a Kalman filter is used. The Digital Kalman filter is an iterative algorithm based on a linear state space model of a non-deterministic system in which internal noise  $\mathbf{V}(k)$  and external noise  $\mathbf{W}(k)$  have been added in Equations (5) and (6) respectively [18]. The noise interacts with the system and modifies it, resulting in a stochastic process. The filter is applied to the systems to identify its state vector and then reconstruct its outputs with minimum errors. In other cases, the Kalman filter is applied to estimate the non-measurable state variables through the engine measured variables. In [19] it was used as a strategy to control position with vibration suppression of Flexible Link Manipulator. In [20],

an Extended Kalman Filter (EKF) is used to estimate position and speed, without any mechanical sensor in an uncertain Permanent Magnet Synchronous Motor (PMSM).

The Equations (23)–(27) define the filter operation at instant  $k$  and are used to identify its state vector that is also illustrated in Figure 6.

$$\hat{\mathbf{G}}(k) = \mathbf{A}\hat{\Delta\mathbf{X}}(k-1) + \mathbf{B}\Delta\mathbf{U}(k-1), \tag{23}$$

$$\mathbf{J}(k) = \{\mathbf{A}\mathbf{P}(k-1)\mathbf{A}^T + \mathbf{Q}\}, \tag{24}$$

$$\mathbf{K}(k) = \mathbf{J}(k)\mathbf{C}^T[\mathbf{C}\mathbf{J}(k)\mathbf{C}^T + \mathbf{R}]^{-1}, \tag{25}$$

$$\hat{\Delta\mathbf{X}}(k) = \hat{\mathbf{G}}(k) + \mathbf{K}(k)\left[\Delta\mathbf{Y}(k) - \mathbf{C}\hat{\mathbf{G}}(k) - \mathbf{D}\Delta\mathbf{U}(k)\right], \tag{26}$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{C}]\mathbf{J}(k), \tag{27}$$

where  $\mathbf{P}(k) = E\{\mathbf{E}_i(k)\mathbf{E}_i^T(k)\}$  is a covariance matrix of a state identification error  $\mathbf{E}_i(k) = \Delta\mathbf{X}(k) - \hat{\Delta\mathbf{X}}(k)$ . The matrix  $\mathbf{Q}(k) = E\{\mathbf{V}(k)\mathbf{V}^T(k)\}$  is an internal noise covariance matrix;  $\mathbf{R}(k) = E\{\mathbf{W}(k)\mathbf{W}^T(k)\}$  represents the external noise covariance matrix;  $\hat{\mathbf{G}}(k)$  and  $\hat{\Delta\mathbf{X}}(k)$  denote a state vector estimation. The quantities  $\hat{\Delta\mathbf{X}}(k)$  and  $\mathbf{P}(k)$  found at the iteration  $k$  are input data to the next iteration  $k + 1$ . The Kalman filter considers no correlation between error vectors, that is:

$$E\{\mathbf{V}(k)\mathbf{V}(j)\} = 0 \quad \forall k \neq j \tag{28}$$

$$\{\mathbf{W}(k)\mathbf{W}(j)\} = 0 \quad \forall k \neq j, \tag{29}$$

$$E\{\mathbf{V}(k)\mathbf{W}(j)\} = 0 \tag{30}$$

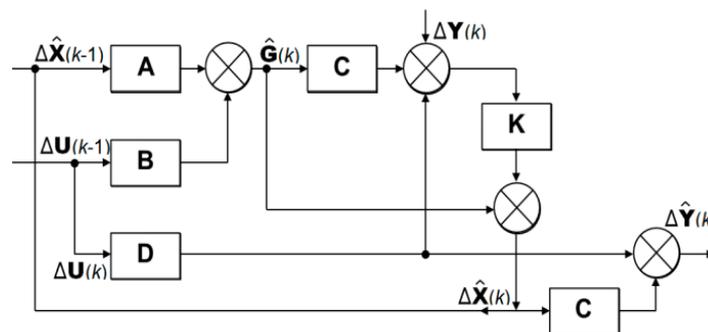
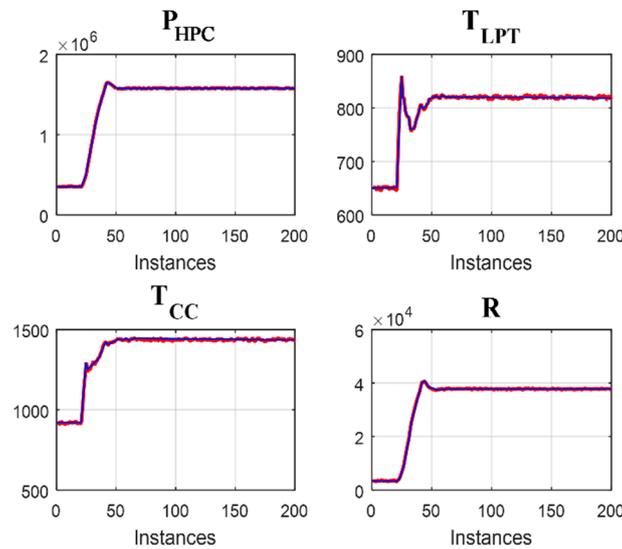


Figure 6. Filtered (reconstructed) GTE variables.

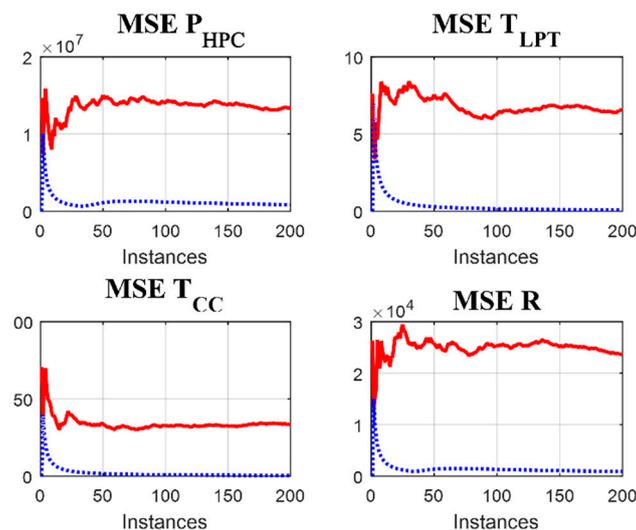
To mitigate the noise effects, Kalman filter Equations (23)–(27) were integrated in the hybrid dynamic model described in Section 3. The original model state  $\mathbf{X}(k)$  was calculated with monitored variables  $\mathbf{Y}(k)$ , while the modified model uses the known measurements  $\mathbf{Y}(k)$  as input data. To verify the new model with the Kalman filter, the data similar to measurements were firstly obtained by adding a Gaussian zero mean noise component to the hybrid model values. Then the new model with the filter was applied to the noisy data. The use of the same engine description for data generation and data filtering allows evaluating the filtering effect. Otherwise, if we use, for example, the nonlinear dynamic model of Equation (2) to generate measurements, the differences between this model and the hybrid model will alter the filtering effect. Other comparisons of linear and non-linear gas turbine performance can be seen in [21]. Figure 7 shows the initial noisy values and filtered values of the same four GTE variables; this figure allows the Kalman filter action verification. It can be seen by comparing

the curves that general behavior of the variables is the same, but the noise in red curves (initial noisy GTE variables) disappears in blue curves (filtered GTE variables). In other words, Figure 7 confirms a correct operation of the filter.



**Figure 7.** Initial noisy (observed) GTE variables in red color and Filtered (reconstructed) GTE variables in blue color.

To quantify the filtering effect, two variations of the mean square error (MSE) were computed for each variable: one between the initial noisy values and the exact values and the second between the filtered values and the exact values. Figure 8 shows these two variations of the MSE against time instances for the same four variables  $P_{HPC}$ ,  $T_{LPT}$ ,  $T_{CC}$ , and  $R$  analyzed before. Comparing the MSE of the initial and filtered values of each variable, we can confirm that the errors after the filtration decrease rapidly and are by far smaller than the initial errors.



**Figure 8.** Mean squared errors (MSE) for measured variables  $P_{HPC}$ ,  $T_{LPT}$ ,  $T_{CC}$ ,  $R$  (red line) and their estimations  $\hat{P}_{HPC}$ ,  $\hat{T}_{LPT}$ ,  $\hat{T}_{CC}$ ,  $\hat{R}$  (blue dotted line).

In real on-board implementation, the significant noise suppression effect illustrated above will result in more precise variables in control algorithms and, in general, more accurate engine control. The use of the Kalman filter in an on-board diagnostic subsystem will provide more precise estimates of the fault parameters, and, consequently, more reliable engine diagnosis.

## 6. Real-Time Response Analysis

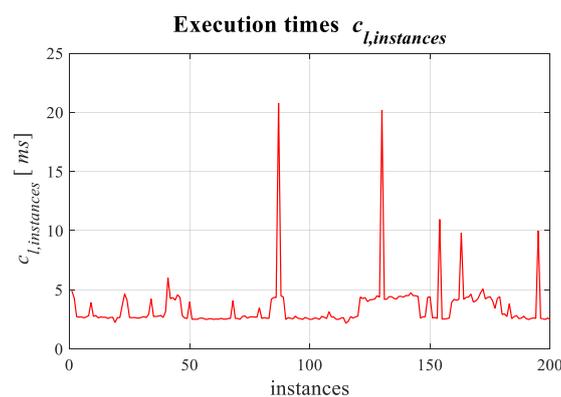
A real-time system (RTS) implemented in a digital computer interacts with the physical world through the conditioning variables (sensors, actuators, and two converters: Analog/Digital and Digital/Analog) and processes them by a task in real-time,  $J_i$ . As shown in [3], a real-time task  $J_i$  is described as an executable job entity characterized by arrival and restriction times associated with an instance  $j_{i,k}$ , where  $i$  is the task index, and  $k$  is the instance index, see [4]. The real-time task has many internal tasks, each one with its execution time  $C_{i,k}$  limited by a maximum deadline  $D_{i,k}$ . These temporal variables can be defined as follows:

- *Execution time* is the time in which the instance with  $k$  index of a real-time task carries out its operations.
- *Deadline* is the maximum time in which a local response from the RTS must be obtained, without modification of the dynamic process.

As shown in [3], real-time systems can be classified as critical and non-critical according to the time limits, synchrony, and quality of response of the environment that interacts with the RTS. For critical RTS, every instance must satisfy  $C_{i,k} < D_{i,k}$ ; for non-critical RTS, the condition  $E\{C_{i,k}\} < D_{i,k}$  is sufficient. For a RTS that is programmed on a digital computer and operates in parallel with a physical system, the analysis of execution times and the fulfillment of the time restrictions ( $D_{i,k}$ ) justify that this RTS works in real-time.

In our case, the gas turbine measurement filtering is performed with the frequency of 20 Hz that corresponds to the interval 5–20 Hz of engine controllers [9]. The noise of real engine sensors is also considered. Given the frequency chosen, a 10 s time period is presented by 200 instances with an interval of 50 ms, and the deadline for all instances is  $D_{1,i} = \Delta t = 50$  ms. This time resolution is higher than the resolution observed in the figures of simulated dynamic processes in published studies [22–25].

Figure 9 shows the results of this real-time simulation in the form of the execution time plotted for each instance. It can be denoted that the execution time is by far smaller than the deadline; this confirms that all instances achieve the requirements of the real-time operation for a critical real time system. A total execution time of the hybrid dynamic model with the Kalman filter is about 0.8 s. In contrast, the nonlinear dynamic model with no filtering runs 15 s the same transient and, therefore, cannot be used for real-time simulation.



**Figure 9.** Execution times over the time instances.

## 7. Conclusions

In this paper, a turbofan engine for a passenger airplane was analyzed. Since the turbofan presents a complex and expensive mechanical system, to avoid real engine operation, the measurements were previously generated by the hybrid dynamic model and a Kalman filter was integrated to filter measurement errors. The RTS was programmed and operates on the SBC.

The development and validation of this new technique for filtering gas turbine measurements in real-time were performed. To verify the possibility of real-time operation, the algorithm was implemented on the SBC Raspberry Pi 2<sup>®</sup> with the Emlid RT real-time operating system. First, the chosen gas turbine engine was simulated by the hybrid dynamic model and its accuracy was validated. Then, based on the MSE analysis, the operation of the filtering algorithm was verified; it was found that the algorithm performs the filtering correctly and suppresses measurement noise. Finally, execution time in every time instance was determined, and it was shown that the proposed filtering algorithm has the capacity to work in real-time, which is a strong requirement for its use in real gas turbine control and diagnosis systems.

**Author Contributions:** Conceptualization, G.D.-R. and P.G.-L.; data curation, P.G.-L. and I.L.; formal analysis, J.-S.V.-M., A.L.-C. and I.L.; investigation, L.H.-G. and J.R.-H.; methodology, L.H.-G. and J.R.-H.; software, G.D.-R. and P.G.-L.; supervision and validation, I.L. and A.L.-C.; writing, original draft, G.D.-R., P.G.-L. and I.L.; writing, review and editing, L.H.-G. and J.R.-H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Instituto Politécnico Nacional.

**Acknowledgments:** Authors are grateful to the Instituto Politécnico Nacional (IPN) for their encouragement and kind economic support.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Nomenclature

### Abbreviations

GTE	gas turbine engine
HPC	high-pressure compressor
HPT	high-pressure turbine
KF	Kalman Filter
LPT	low-pressure turbine
MSE	mean square error
RTS	real-time System
SBC	single board computer

### Variables

a	polynomial coefficient
<b>A,B,C,D</b>	state space model matrixes
C	execution time
D	deadline time
G	mass flow
<b>G</b>	state vector estimation; mass consumption
k	iteration number
<b>K</b>	Kalman filter gain matrix
m	number of monitored variables; bypass ratio
n	rotation speed, number of operating conditions
P	pressure
<b>P</b>	covariance matrix of state variables
<b>Q</b>	internal noise covariance matrix
<b>R</b>	external noise covariance matrix
r	number of fault parameters
R	engine thrust
SFC	specific fuel consumption
T	temperature
u	control variable
<b>U</b>	vector of operating conditions
<b>X</b>	vector of state variables
<b>Y</b>	vector of monitored variables
$\Delta$	difference

$\pi$	pressure ratio
$\Theta$	vector of fault parameters

## Indexes

air	air
c	compressor
cc	combustion chamber
f	fuel
h	high pressure rotor
l	low pressure rotor
s, st	static
z	zone

## References

- Haykin, S. *Adaptive Filter Theory*, 3rd ed.; Prentice Hall: Upper Saddle River, NY, USA, 1991; pp. 32–40.
- Medel, J. Análisis de dos métodos de estimación para sistemas lineales estacionarios e invariantes en el tiempo con perturbaciones correlacionadas con el estado observable del tipo: Una entrada una salida. *Comput. Sist.* **2002**, *5*, 215–222.
- Medel, J.; Guevara, P.; Cruz, D. *Temas Selectos de Sistemas en Tiempo Real*; Editorial Politécnico: Mexico City, México, 2007; pp. 156–187.
- Nyquist, H. Certain Topics in Telegraph Transmission Theory. *Trans. Am. Inst. Electr. Eng.* **1928**, *47*, 617–644. [[CrossRef](#)]
- Buttazzo, G.C. *Hard Real-Time Computing Systems*; Springer: Boston, MA, USA, 1997; pp. 23–43.
- Chui, C.K.; Chen, G. *Kalman Filtering with Real-Time Applications*; Springer Science and Business Media LLC: Berlin, Germany, 1987; Volume 17, pp. 20–29.
- Boyce, M.P. *Gas Turbine Engineering Handbook*, 4th ed.; Elsevier Inc.: Oxford, UK, 2012; p. 956.
- Kulikov, G.G.; Thompson, H.A. *Dynamic Modelling of Gas Turbines: Identification, Simulation, Condition Monitoring, and Optimal Control*; Springer: London, UK, 2004; p. 309.
- Volponi, A. Gas Turbine Engine Health Management: Past, Present, and Future Trends. *J. Eng. Gas Turbines Power* **2014**, *136*, 051201. [[CrossRef](#)]
- Yunusov, S.; Labendik, V.; Guseynov, S. *Monitoring and Diagnostics of Aircraft Gas Turbine Engines*; LAMBERT Academic Publishing GmbH&Co: Saarbrücken, Germany, 2014; p. 196.
- Saravanamuttoo, H.I.H.; MacIsaac, B.D. Thermodynamic Models for Pipeline Gas Turbine Diagnostics. *J. Eng. Power* **1983**, *105*, 875–884. [[CrossRef](#)]
- Ekinci, S.; GURSOY, G.; YAVRUCUK, I.; UZOL, O. An approach to sensor failure recovery using a neural network based adaptive observer. In Proceedings of the IGTI/ASME Turbo Expo 2016, Seoul, Korea, 13–17 June 2016; ASME Paper GT2016-57811. pp. 1–10.
- Loboda, I.; Feldshteyn, Y.; Yepifanov, S. Gas Turbine Diagnostics Under Variable Operating Conditions. *Int. J. Turbo Jet-Engines* **2007**, *24*, 231–244. [[CrossRef](#)]
- Pinelli, M.; Spina, P. Gas Turbine Field Performance Determination: Sources of Uncertainties. *J. Eng. Gas Turbines Power* **2000**, *124*, 155–160. [[CrossRef](#)]
- Sampath, S.; Singh, R. An Integrated Fault Diagnostics Model Using Genetic Algorithm and Neural Networks. *J. Eng. Gas Turbines Power* **2004**, *128*, 49–56. [[CrossRef](#)]
- Ogaji, S.O.T.; Li, Y.G.; Sampath, S.; Singh, R. Gas Path Fault Diagnosis of a Turbofan Engine from Transient Data Using Artificial Neural Networks. In Proceedings of the Turbo Expo 2003, Atlanta, GA, USA, 16–19 June 2003; Volume 4, pp. 405–414.
- Carvalho, A.; Machado, C.; Moraes, F. Raspberry Pi Performance Analysis in Real-Time Applications with the RT-Preempt Patch. In Proceedings of the 2019 Latin American Robotics Symposium, Rio Grande, Brazil, 23–25 October 2019; pp. 162–167.
- Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
- Reddy, M.P.P.; Jacob, J. Vibration Control of Flexible Link Manipulator Using SDRE Controller and Kalman Filtering. *Stud. Inform. Control.* **2017**, *26*, 143–150. [[CrossRef](#)]

20. Dchich, K.; Zaafour, A.; Chaari, A. Combined Riccati-Genetic Algorithms Proposed for Non-Convex Optimization Problem Resolution—A Robust Control Model for PMSM. *Stud. Inform. Control* **2015**, *24*, 317–328. [[CrossRef](#)]
21. Kamboukos, P.; Mathioudakis, K. Comparison of Linear and Non-Linear Gas Turbine Performance Diagnostics. *J. Eng. Gas Turbines Power*. **2003**, *127*, 451–459. [[CrossRef](#)]
22. Panov, V. Auto-tuning of real-time dynamic gas turbine models. In Proceedings of the IGTI/ASME Turbo Expo 2014, Düsseldorf, Germany, 16–20 June 2014; pp. 1–10, ASME Paper GT2014–25606.
23. Dewallef, P.; Borguet, S. A Methodology to Improve the Robustness of Gas Turbine Engine Performance Monitoring Against Sensor Faults. *J. Eng. Gas Turbines Power* **2013**, *135*, 051601. [[CrossRef](#)]
24. Loboda, I.; Feldshcheyn, Y. Polynomials and Neural Networks for Gas Turbine Monitoring: A Comparative Study. *Int. J. Turbo Jet-Engines* **2011**, *28*, 227–236. [[CrossRef](#)]
25. Simon, D.L.; Armstrong, J.B.; Garg, S. Application of an Optimal Tuner Selection Approach for On-Board Self-Tuning Engine Models. *J. Eng. Gas Turbines Power* **2012**, *134*, 041601. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).