Morteza Montazeri-Gh
Soroush Abyaneh

https://doi.org/10.21278/TOF.41405
ISSN 1333-1124
eISSN 1849-1391

ERROR EVALUATION OF HARDWARE-IN-THE-LOOP SIMULATION OF A GAS TURBINE ENGINE FUEL CONTROLLER

Summary

Correct implementation of a fuel control algorithm for a gas turbine engine (GTE) in an electronic control unit (ECU) is one of the basic challenges in the development of a GTE fuel control system. A common measure in such implementation is the error of the hardware-in-the-loop (HIL) simulation test. In this paper, evaluation and diminution of the hardware-in-the-loop test error for a gas turbine engine fuel controller is presented. For this purpose, a fuel controller has been designed for a power generating gas turbine engine. The designed controller was then implemented in the PC104 hardware and tested in an HIL simulation. The test results were then evaluated in order to study the controller functionality. In this study, a procedure is proposed for evaluating the implementation of the fuel control algorithm in PC104 and diminishing the HIL simulation errors. Finally, it is shown that the proposed approach to decreasing the HIL simulation error is effective.

Key words: *control strategy implementation, error evaluation, fuel control system, gas turbine engine, hardware-in-the-loop simulation*

1. Introduction

An effective approach to an accurate control strategy design and implementation of an electronic control unit in a gas turbine engine is a remaining challenge in the area of industrial applications. Technological developments have enabled researchers to establish simulation platforms for observing ECU characteristics and real time testing, such as hardware-in-the-loop simulation. In an HIL simulation, there is no need for all systems to be real but a combination of precise numerical models and hardware components is required instead. HIL testbeds have been proven to be successful in many applications, such as in wind energy systems [1], automotive industry [2], structural dynamics [3] and electrical systems [4-5].

HIL simulations have been widely used in the ECU design process of many plants including gas turbines [6-7]. This is an electronic hardware, programmed to control the engine's inlet fuel flow as exactly as required. Therefore, its acceptable performance is unquestionably vital before the engine is manufactured. The ECU design and test process has three steps. First, the engine and its controlling algorithm are designed. Then, they are modelled using software so they can be simulated numerically in real time. This step is called the Software-in-the-Loop (SIL) simulation. Finally, the control strategy is implemented in the hardware. In order to verify this implementation, the HIL simulation is a suitable approach in the initial verification of the control system, as reported in many studies. For example, in [8] the design of a fuzzy controller for a micro-jet engine is discussed. In [9], a start controller for a mini turbojet engine is investigated. Multi-variable controllers for turbojets are tested in [10-11] and those for turbofan engines are tested in [12-13]. Every HIL simulation, however, is subject to error (when compared with the corresponding results obtained from SIL simulations performed in the same conditions) due to implementation procedures, hardware connections and wirings, unaccounted system inputs, etc. So, a systematic approach to the HIL error evaluation and consequently the error diminution is necessary. Consequently, some solutions have been introduced for real time applications in [14-15-16-17]. In spite of this extensive use of HIL simulations, to the best knowledge of the authors, no report has been published on an HIL simulation of a two-shaft power generating gas turbine engine including a simulation error evaluation. In this study, an HIL simulation of a fuel controller of a power generating gas turbine has been conducted and a new approach to the error evaluation and diminution of the HIL test is proposed. Such innovative contribution to HIL simulation and error evaluation for any kind of power generating gas turbine engines has not yet been reported (to the best knowledge of the authors). The engine is a numerically-simulated component of the test setup where its ECU (controller) is an embedded system, deployed on the PC104 hardware. Therefore, any deviations between the SIL and the HIL simulation results originate from the deployment procedure of the ECU control algorithm on the hardware which is implemented via C++ coding. Any parameter in this code that could contribute to error generation is to be identified and optimized in a way to reduce the HIL simulation error to a minimum.

It should be mentioned that there are some differences between the simulated power plant and a real engine. For example, in the simulated plant, two parameters are considered to affect the fuel control algorithm. In a real engine, there are more such parameters. Also, no uncertainties have been considered in modelling the engine, while in a real gas turbine, uncertainties exist. Another difference is in the number of inputs to the engine model (two inputs in this paper) and a real engine (numerous inputs).

2. Control system description

The present study is about a closed loop control system for a two-shaft turbo-shaft engine. The purpose of this system is to control the amount of fuel, required for the engine to perform desirably with respect to all physical and operational limitations of the gas turbine engine. A Matlab/Simulink model of this system has been developed as depicted in Fig. 1. This system is made of two blocks: the turbo-shaft engine and the ECU. The fuel controller (ECU) is designed to calculate the required amount of fuel mass flow (FMF) in the engine. In the engine block, two other parameters are computed, namely, the angular velocity of the power turbine axis (NPT) and the angular velocity of the gas generator axis (NGG). This control system has two inputs. The first is a torque load, exerted on the engine as a function, set by the designer, and the second is a reference rotational velocity (RRV) of the power turbine axis, which is to be a constant value and is applied to the ECU. The signal connections between the ECU, the engine and the input blocks are demonstrated in Fig. 1.

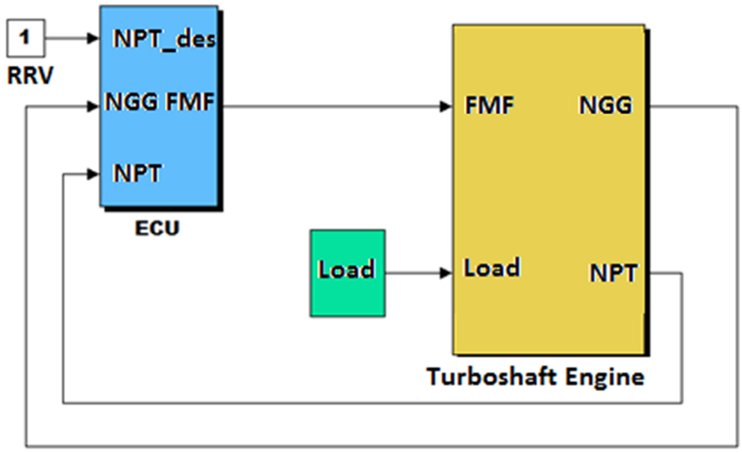

Fig. 1 Block diagram of the control system

2.1 Engine

A Wiener block structure has been used for modelling the gas turbine engine [18]. As shown in Fig. 2, the Wiener structure is an interconnection of a linear dynamic block and a non-linear static block between the fuel input and each output of the engine, representing the dynamic and static characteristics of the gas turbine, respectively.

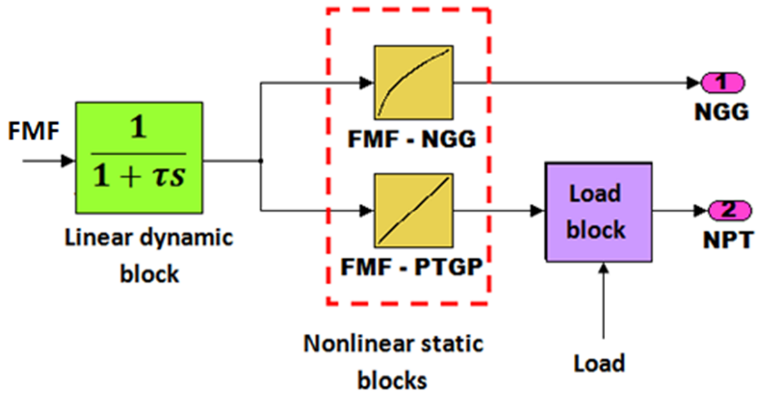

Fig. 2 The Wiener structure of the gas turbine engine model

In this study, the dynamic block is considered to be a first order lag transfer function H(s) as in Eq. 1 for both the NPT and the NGG. A single lookup table has been assigned to correlate between the FMF and the NGG. Another lookup table correlates between the FMF and the power turbine generated power (PTGP). The NPT is calculated in the load block that receives PTGP and load as input signals. For more information regarding Wiener models of a two-shaft gas turbine engine please refer to [19-20].

$$H(s) = \frac{1}{1+\tau s} \tag{1}$$

where τ is the time lag constant and s is the transfer function Laplace variable.

2.2 Controller

The control algorithm in the ECU model is based on the min-max control strategy. The min-max control algorithm is one of the most widely used control strategies for electronic control units of gas turbines.

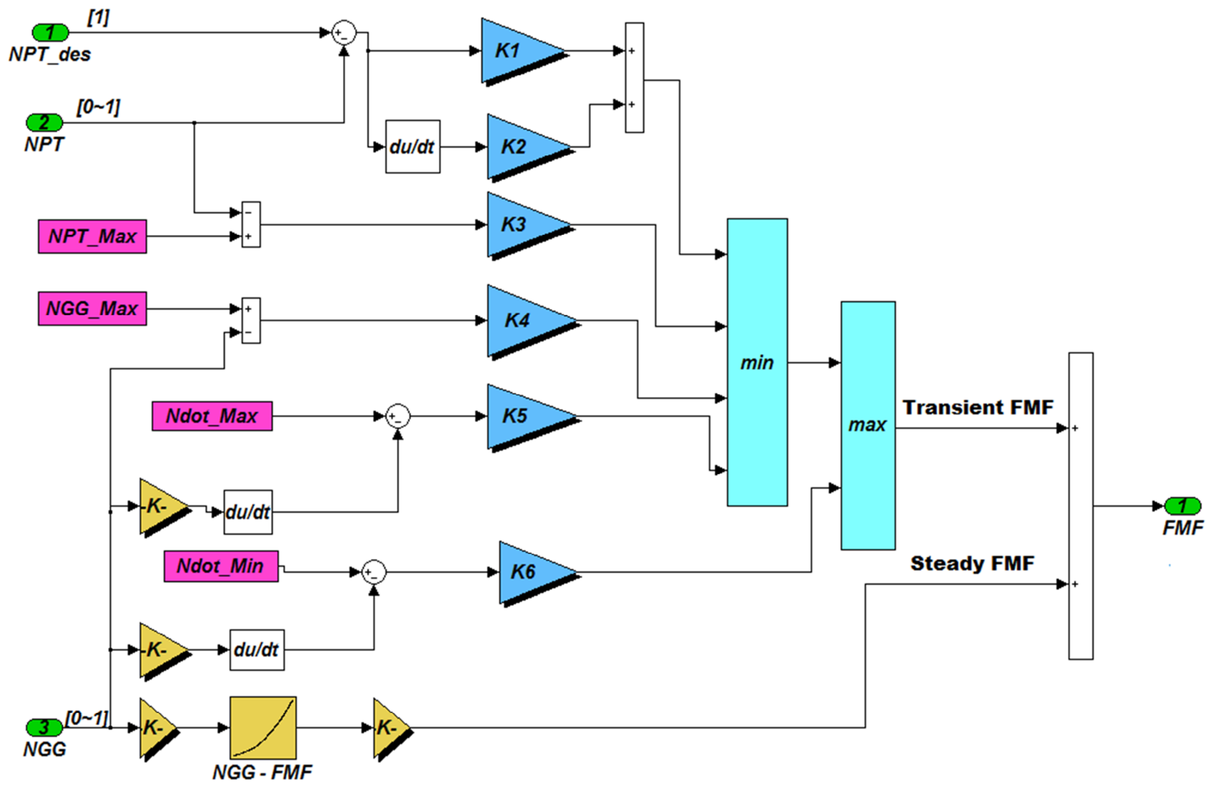

Fig. 3 ECU block diagram

The min-max controller consists of several transient control loops, as well as a single steady-state control loop. Every loop is to satisfy a single control requirement that has been imposed on the engine in order to guarantee its safe and optimized functionality. These requirements include:

- maintaining the NPT at its desired value
- bounding the NGG, the NPT and their changing rate values within their allowable limits
- providing enough fuel to avoid flame-out in the combustion chamber

In this strategy, the most conservative approach is taken to choose the required fuel for the engine in order to observe all control requirements through some minimum/maximum-based selections. The block diagram of such a controller is presented in Fig. 3.

3. HIL test setup

When this entirely software-based system is simulated and run in real time, it is called the software-in-the-loop or SIL simulation. With this numerical simulation, an overall understanding of the control system behaviour is provided. Nevertheless, if the same control algorithm, implemented in a real controller, is to be used for a gas turbine engine, the applicability of the control strategy implementation in that hardware and its satisfactory performance should be examined through impeccable tests before any further approach is taken. However, testing real systems at first is unacceptable due to unexpected risks and costs involved. The hardware-in-the-loop simulation is an intermediate approach in which some parts of the system remain as a software model, interacting with some other physical parts in real time. In this case, the gas turbine engine remains as a software-based model, while the control algorithm is embedded in VDX6354 PC104 (manufactured by ICOP Technology Inc) as the ECU via a C++ code. Therefore, in this HIL setup, the controlled plant model (gas turbine engine) is simulated on a generic PC target connected to the controller PC104 which is a modular PC in which the control strategy is implemented. A PCI data acquisition card (PCI-1711 manufactured by Advantech Company) is used in the PC to connect the signals of both systems. Fig. 4 shows a schematic of the signal transmission between the hardware and software elements of the testing setup via a data acquisition card.

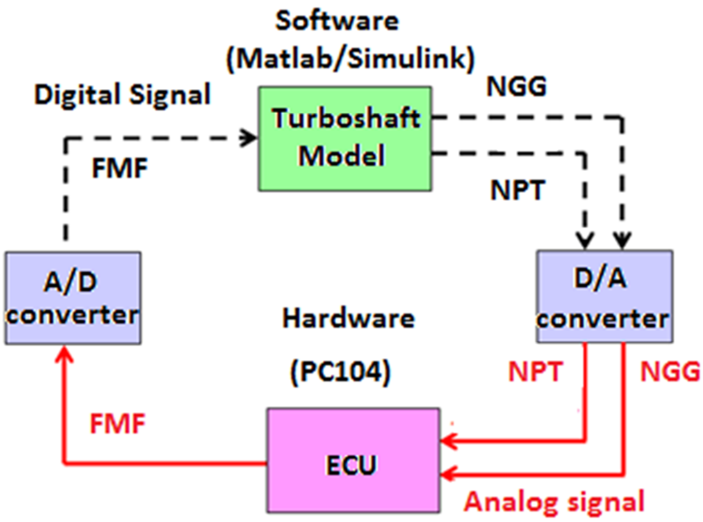

Fig. 4 Schematic of signal transmission in the ECU HIL simulation

The VDX-6354 PC104 family of the embedded controller is designed to provide a migration path for projects facing end-of-life challenges with their existing x86 based PC104 controller. The VDX-6354 family of the controller is designed as a plug in replacement, with backward compatibility to support the legacy software to help extend the existing product life cycle without heavy re-engineering [21]. The HIL simulation framework for testing the ECU is presented in Fig. 5.

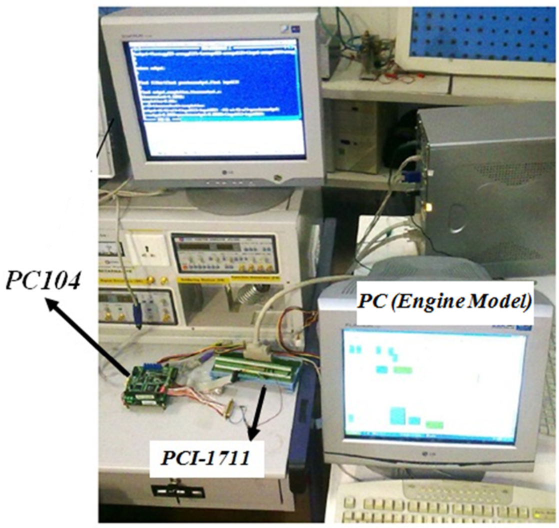

Fig. 5 HIL testbed

4. HIL test results

For the verification, the results should be compared with those of the SIL simulation. Therefore, the same torque load and RRV that were used in the SIL simulation are again chosen for the HIL test and the HIL setup is run. The n ormalized NGG, NPT and FMF from both HIL and SIL simulations are demonstrated in Fig. 6-8, respectively.

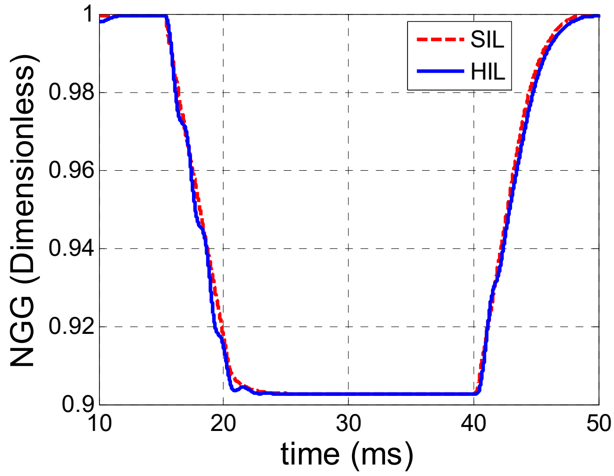

Fig. 6 NGG in HIL and SIL simulations

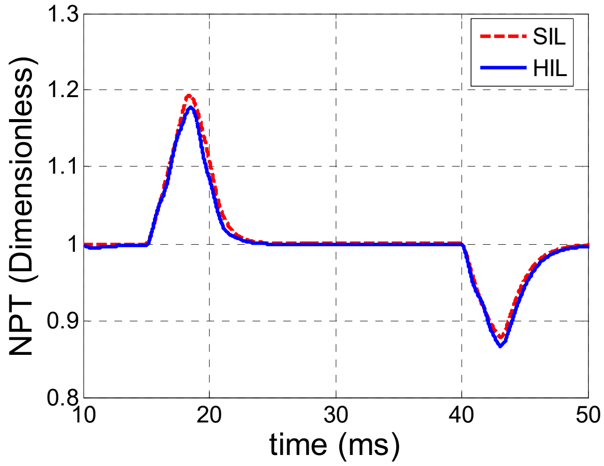

Fig. 7 NPT in HIL and SIL simulations

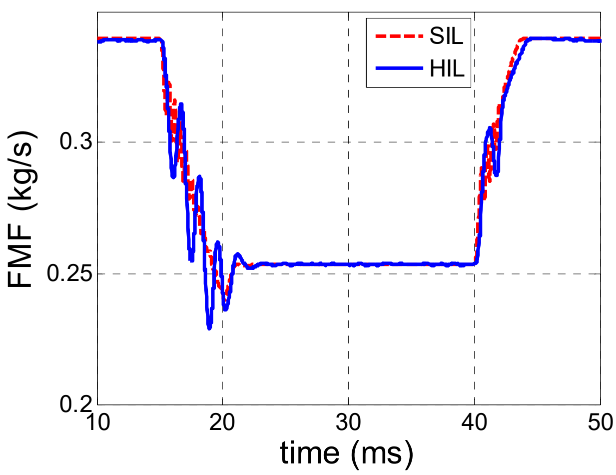

Fig. 8 Fuel flow rate (FMF) in HIL and SIL simulations

5. Error evaluation

Though slightly, the results obtained from the HIL simulation differ from those of the SIL simulation. This deviation results from the use of hardware in the simulation. Therefore, the issues related to the lack of precision in the simulation and the methods for reducing them are of great importance. Ideally, if A/D and D/A conversions as well as the implementation procedure are error-free, then the HIL and SIL simulation results must coincide exactly. However, this is not the fact. This difference is attributed to the technique of the ECU implementation in the PC104. In other words, PC104 is a digital version of the continuous ECU block that was used previously in the SIL simulation. In fact, to investigate the causes of HIL simulation errors further, the ECU's discrete model parameters and other important factors ought to be carefully accounted for and inspected. These parameters and factors contribute to A/D conversions and include: sampling time, derivative time step constant and transfer function discretizing method. In the following section, the effect of each of these parameters on the test is discussed in order to find the best option for minimizing the overall error. Also it should be noted that the D/A conversion of the digitized signal is not error-free. This error is caused by the noise introduced by the converter which leads to the rounding error between the analog output voltage and the input digitized value of the converter. The effect of noise on the errors is discussed in this paper.

5.1 Sampling time

Sampling time is kept constant during the simulation using a defined function in the PC104 coding. To this end, at the beginning of the compilation, a timer function is activated and the time elapsed for the code to be executed for each sampled data is measured using the PCM-5115 counter and then subtracted from the constant sampling time. Finally, a delay equal to this difference time period is applied to the compiler. Therefore, the sampling time is consolidated throughout the simulation.

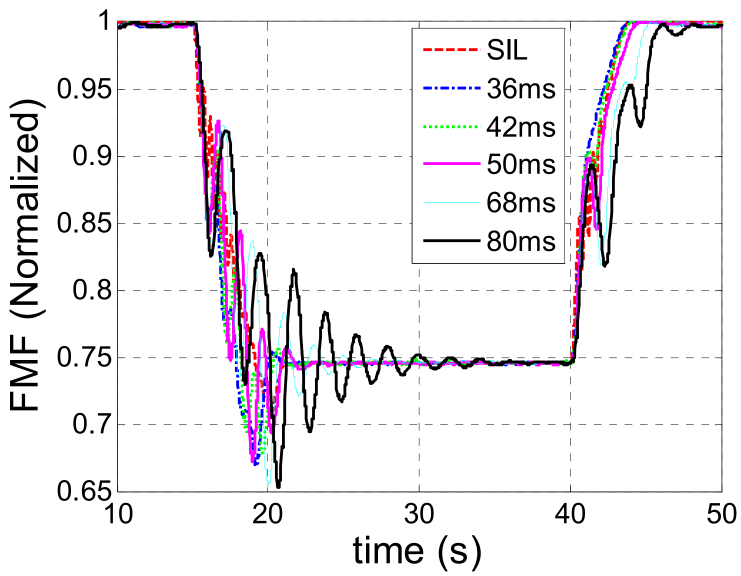

Fig. 9 Fuel flow rate in HIL simulations with five different sampling times

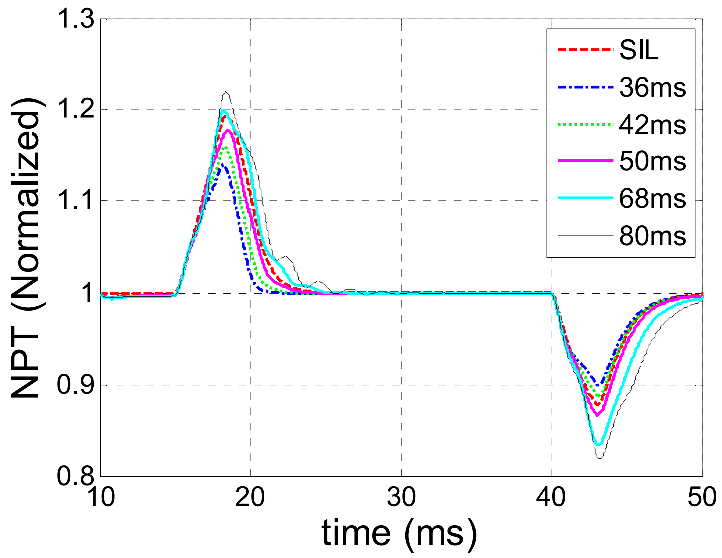

Fig. 10 NPT in HIL simulations with different sampling times

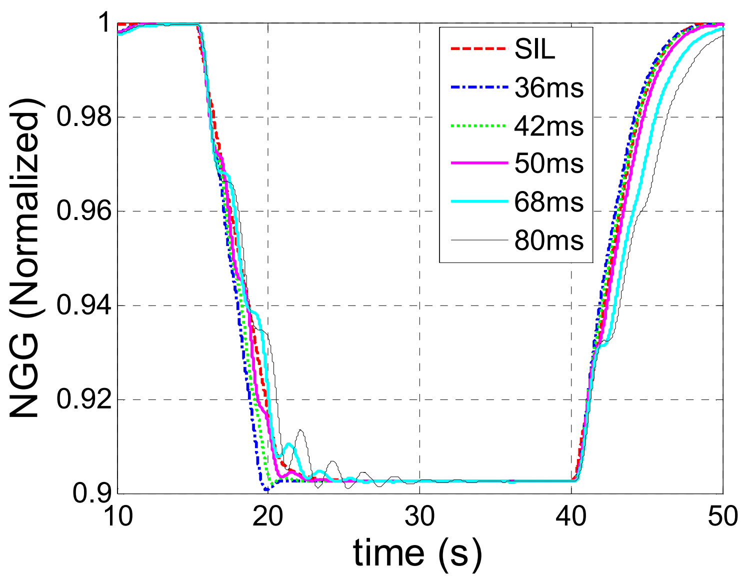

Fig. 11 NGG in HIL simulations with different sampling times

Table 1 Errors in HIL simulations with five different sampling times

Sampling time (ms)	Error in FMF signal (%)		Error in NGG signal (%)		Error in NPT signal (%)	
	Max. error	Mean error	Max. error	Mean error	Max. error	Mean error
36	7.5	0.8	2.3	0.23	8.1	0.7
42	4.5	0.6	1.5	0.11	5.3	0.4
50	5.3	0.7	0.7	0.09	2.4	0.3
68	8.3	1.1	1.4	0.21	5.2	0.6
80	8.9	1.8	2.4	0.35	7.3	1.1

As could be deduced from these results, too low or too high sampling time would lead to a higher error. For example, when the sampling time is 80 ms, the ECU exhibits an over-fluctuating response. This is because of the too much delay imposed by the code for each epoch. So this sampling time is not suitable for the test. On the other hand, when the sampling time is 36 ms or 42 ms, fluctuations mainly disappear but the error is considerable. As a matter of fact, in the case of small sampling times, the time needed for the implemented code to be run will be longer than the sampling time; therefore, before the code execution is completed, another sampled data is fed into the ECU. This will increase the simulation mean error. After the inspection and analysis of these numbers according to the average error for each signal (such as curve fitting and finding its minimum value), it could be inferred that the time constant of approximately 46 milliseconds is the best choice for this parameter.

5.2 Derivative time step constant

In the continuous version of the ECU in Simulink, various functions are used. In implementing these functions in C++, some uncalled-for restrictions are experienced. However, the implementation procedure of these functions is another factor that could potentially generate error. One of the implemented functions in the code structure is the derivative function.

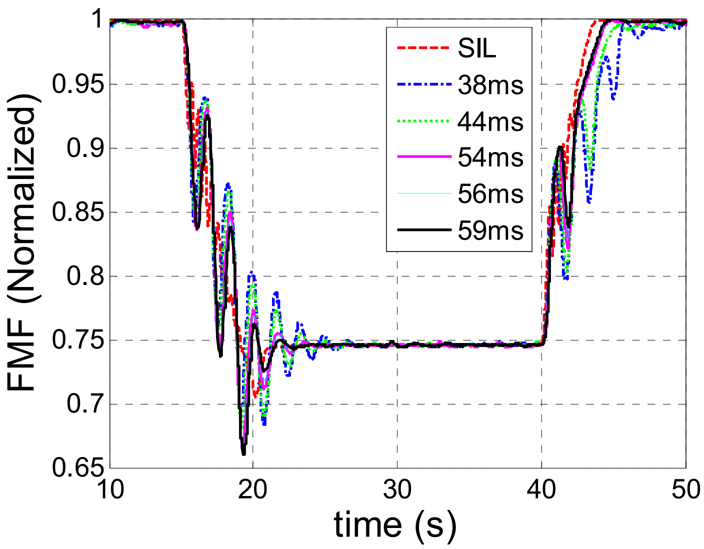

Fig. 12 Fuel flow rate in HIL simulations with five different derivative time step constants

This function has a time constant which has a significant impact on the accuracy of the results. This constant is actually the time step required for calculating a derivative numerically at any point and is assumed to be a definite value in the code by the programmer. Regulating this constant is one of the most important factors for minimizing the HIL simulation error. For five different derivative time step constants of 38, 44, 54, 56 and 59 ms, an HIL simulation is performed with the rest of the parameters remaining the same (the best sampling time of 46 ms obtained in the previous subsection has been chosen for all simulations in this subsection). The results generated from the FMF, NGG and NPT signals are presented in Fig. **12**-14, respectively.

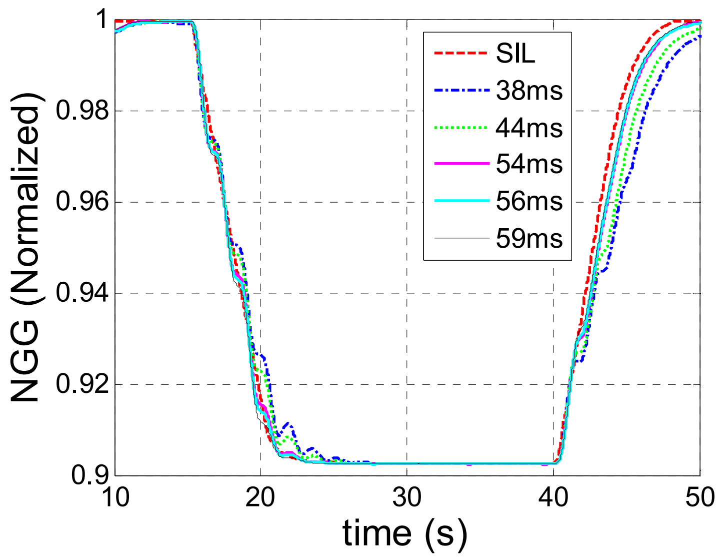

Fig. 13 NGG in HIL simulations with different derivative time step constants

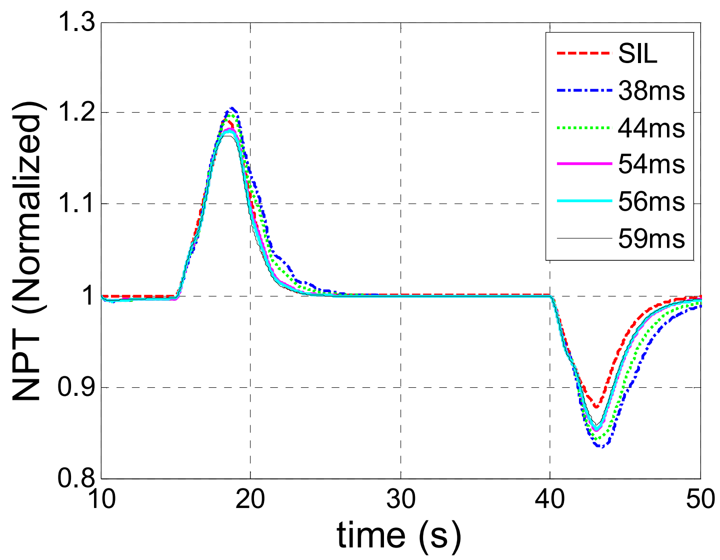

Fig. 14 NPT in HIL simulations with different derivative time step constants

The mean and maximum errors for all signals in each test are summarized in Table 2.

Table 2 Errors in HIL simulations with five different derivative time step constants

Sampling time (ms)	Error in FMF signal (%)		Error in NGG signal (%)		Error in NPT signal (%)	
	Max. error	Mean error	Max. error	Mean error	Max. error	Mean error
38	13.2	1.5	2.4	0.3	7.7	0.1
44	11.4	1.2	1.9	0.2	6.2	0.6
54	10.3	0.9	0.13	0.1	3.0	0.3
56	9.3	0.8	0.8	0.11	2.1	0.27
59	11.3	0.9	2.9	0.8	2.5	0.3

The results show that, for too low derivative time step constants, the controller response is not satisfactory. For too high constants, such as 59 ms, the error also increases due to inappropriate approximation. A curve is then fitted through the points whose x-coordinates are derivative time step constants and y-coordinates are the corresponding errors of the FMF.

The best choice for the constant is found to be the minimum of the curve which is approximately 58 ms.

It should be noted that the effect of the sampling time and the derivative time step in the error evaluation have been considered independently. These two parameters have no interface in the implemented code and therefore, there is no need for taking their interaction into account. As will be discussed in the following section, the final result made with this assumption has the highest accuracy with a satisfactory error reduction compared with earlier simulations.

5.3 Transfer function discretizing method

The other cause of error in the HIL simulation is dependent on the accuracy of the discrete version of the only transfer function present in the ECU continuous model. All methods of continuous to digital conversion are somehow erroneous. This error will vary depending on what kind of approximation is used in the discretization procedure. To determine this, the transfer function used in the control structure will be assigned a step input. Then, the transfer function is converted to a digital form using three different transformation methods, including Tustin's method (T), the zero-order hold (ZOH) and the matched pole-zero (MPZ) method.

The transfer function that has been used in the ECU is a first-order function with the form of $G(s)=\frac{\mathrm{a}}{\mathrm{s}+\mathrm{a}}$, where a is a constant and s is the transfer function Laplace variable. The Z-transformation of this transfer function with each of these methods is described in Equations 2-4 [22-23]:

$$\hat{G}_T(z)=\frac{a}{\frac{2(z-1)}{T(z+1)}+a} \tag{2}$$

$$\hat{G}_{MPZ}(z)=\frac{1-\mathrm{e}^{-aT}}{2}\left(\frac{z+1}{z-\mathrm{e}^{-aT}}\right) \tag{3}$$

$$\hat{G}_{ZOH}(z)=(1-z^{-1})Z\left\{L^{-1}\left[\frac{G(s)}{s}\right]\right\} \tag{4}$$

In Eqs. 2-3, T is the sampling time, while in Eq. 4, L and Z represent the Laplace and Z-transformation operators, respectively. Also, $\hat{G}_T$, $\hat{G}_{MPZ}$ and $\hat{G}_{ZOH}$ are the Z-transformation of the transfer function $G(\mathrm{s})$ with Tustin's, matched pole-zero and the zero-order hold methods with z as their variable in the Z domain.

The unit step responses of these digital functions and the continuous transfer function are compared in Fig. 15. According to the results, it could be inferred that with the Tustin transform, less error is generated than when other methods are applied.

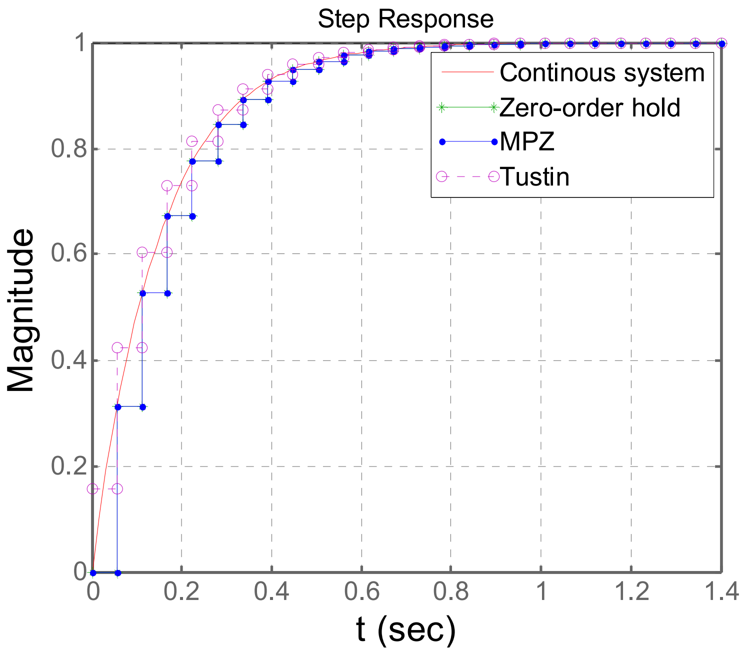

Fig. 15 Unit step response of continuous and digital transfer functions

The HIL simulation is repeated with the obtained ECU discrete model parameters. So, the Tustin method is used for discretising the transfer functions, and the sampling time and the derivative time step constants are set to be 46 and 58 milliseconds, respectively. The new result for the FMF signal (with a different load input for the HIL and SIL simulations) is shown in Fig. 16. As could be deduced from this result, the difference between HIL and SIL has been reduced significantly with the new set of parameters. The mean error for the FMF signal has been decreased to about 0.2%.

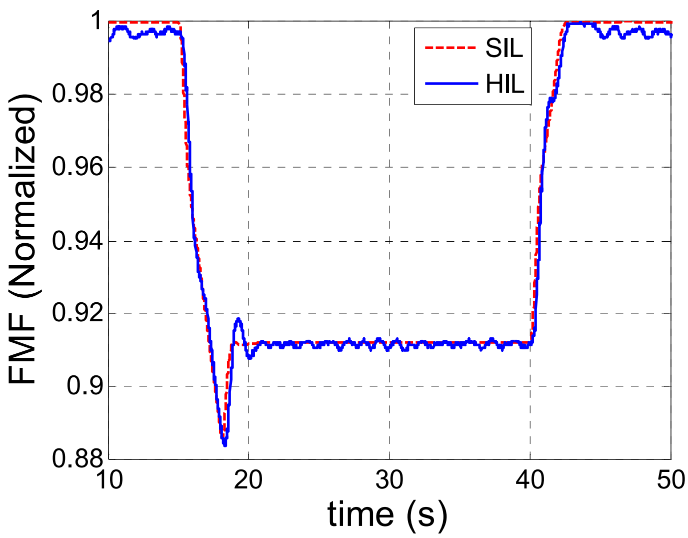

Fig. 16 Fuel flow rate in HIL simulation with ECU's discrete model optimum parameters

6. Conclusion

In this paper, a procedure for evaluation and diminution of an HIL test error for a two-shaft gas turbine engine fuel controller is presented. For this purpose, the studied power generating gas turbine engine is first introduced and the structure of its electronic control unit based on the min-max control strategy is explained. The min-max controller consists of various control loops, each of which is responsible for satisfying one of the engine control requirements. In order to ensure acceptable functionality of the controller, its algorithm is implemented in an actual hardware and is tested in an HIL simulation in order to verify its correct performance. Although the convergence of the HIL simulation results to those of the SIL simulation is admissible, these results do not coincide exactly. This means that there is an error in the test, though in small quantities. Therefore, the entire implementation procedure is inspected to search for any parameter that could contribute to the error. Upon the inspection, three parameters have been identified: sampling time, derivative time step constant and transfer function discretizing method. The error evaluation of the HIL simulation demonstrates that the best choices for these parameters in order to reduce the test error are: 46ms, 58ms and Tustin's method. By repeating the HIL simulation with the optimized parameters and a different load input, the result shows that the HIL error diminution is substantial and therefore, the proposed approach is effective.

REFERENCES

[1] Li H, Steurer M, Shi K, Woodruff S and Zhang D (2006). Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation. IEEE Trans. Ind. Electron 53(4): 1144–1151. https://doi.org/10.1109/TIE.2006.878319

[2] Schlager M, Elmenreich W and Wenzel I (2006). Interface design for hardware-in-the-loop simulation. Proceeding of IEEE International Symposium on Industrial Informatics, Montreal, Canada 2: 1554–1559. https://doi.org/10.1109/ISIE.2006.295703

[3] Chen C and Ricles J M (2009). Improving the inverse compensation method for real-time hybrid simulation through a dual compensation scheme. Earthquake Engineering and Structural Dynamics 38(10): 1237-1255. https://doi.org/10.1002/eqe.904

[4] Jun Z, Jianbo G, Sybille, G and Yiying Z (2013). Implementation, verification and application of power hardware in-the-loop simulation for HVDC links using TLM method. International Transactions on Electrical Energy Systems 23: 1139–1155. https://doi.org/10.1002/etep.1645

[5] Velvelidis V, Hollinger R and Wittwer C (2014). Hardware-in-the-loop testing of control strategies for distributed generation in the smart grid. Energy Technology 2: 100–106. https://doi.org/10.1002/ente.201300079

[6] M Montazeri-Gh, M Nasiri and S Jafari (2011). Real-time multi-rate HIL simulation platform for evaluation of a jet engine fuel controller. Simulation Modeling Practice and Theory 19 (3): 996–1006. https://doi.org/10.1016/j.simpat.2010.12.011

[7] Jun L, Ying-Qing G and Hai-Quan W (2008). Rapid prototyping real-time simulation platform for digital electronic engine control. Second International Symposium on Systems and Control in Aerospace and Astronautics, Art. No. 4776230: 1-5.

[8] Watanabe A, Ölçmen S M, Leland R P, Whitaker K W, Trevino L C and Nott C (2006). Soft computing applications on a SR-30 turbojet engine. Fuzzy sets and systems 157(22): 3007-3024. https://doi.org/10.1016/j.fss.2006.05.011

[9] Cheng T (2004). Hardware in the loop simulation of mini type turbojet engine digital control regulator. Journal of Aerospace Power 19(3): 383-386.

[10] Wang H, Guo Y and Lu J (2007). Design and validation of aeroengine control system with nonfully recovering LQG/LTR method. Second International Conference on Innovative Computing, Information and Control: 469.

[11] Yang Y W (2004). Digital electronic control system design and test for a certain small turbojet engine. Journal of Propulsion technology 25(6): 526-529.

[12] Duan C, Xie S S and Cai K L (2005). Hardware-in-the-loop simulation of a turbofan aero engine control system. Journal of Propulsion Technology 5: 434-438.

[13] Bao W, Sui Y F, Liu Z M and Liu J F (2006). Design and realization of hardware-in-the loop simulation for turbofan-engine. Journal of System Simulation 10: 603-615.

[14] Hassana M A and Abido M A (2014). Real time implementation and optimal design of autonomous microgrids. Electric Power Systems Research 109:118–127. https://doi.org/10.1016/j.epsr.2013.12.001

[15] Guo-Ping L, Jian S and Zhao Y B (2013). Design, Analysis and Real-time Implementation of Networked Predictive Control Systems. Acta Automatica Sinica 39(11).

[16] Mamdoohi G et al (2012). Implementation of genetical algorithm in an embedded microcontroller-based polarization control system. Engineering Applications of Artificial Intelligence 25: 869-873. https://doi.org/10.1016/j.engappai.2012.01.018

[17] Wenxin L, Liu L and Cartes D A (2014). Efforts on real-time Implementation of PSO based PMSM parameter identification, Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century: 1-7.

[18] Schetzen M (1981). Nonlinear system modeling based on the wiener theory. Proceeding of IEEE 69(12): 1557–1573. https://doi.org/10.1109/PROC.1981.12201

[19] Mohammadi E and Montazeri-Gh M (2015). A new approach to the gray-box identification of wiener models with the application of gas turbine engine modeling. Journal of Engineering for Gas Turbines and Power 137(7): 071202. https://doi.org/10.1115/1.4029170

[20] Mohammadi E, Montazeri-Gh and M Khalaf P (2013). Metaheuristic design and optimization of fuzzy-based gas turbine engine fuel controller using hybrid invasive weed optimization/particle swarm optimization algorithm. ASME Journal of Engineering Gas Turbines Power 136(3): 031601. https://doi.org/10.1115/1.4025884

[21] VDX-6354 / VDX-6354-PLUS user's manual

[22] Chen T and Francis B (1995). Optimal sampled-data control systems. 1st edition, London, UK, Springer-Verlag. https://doi.org/10.1007/978-1-4471-3037-6

[23] Yang W Y (2009). Signals and Systems with MATLAB. Springer. https://doi.org/10.1007/978-3-540-92954-3

Submitted: 06.4.2016

Accepted: 12.6.2017

Morteza Montazeri-Gh
Soroush Abyaneh
Systems Simulation and Control Laboratory,
Department of Mechanical Engineering, Iran
University of Science and Technology,
Tehran, Iran