



Hierarchical Discrete Event Supervisory Control of Aircraft Propulsion Systems

Murat Yasar, Devendra Tolani, and Asok Ray
The Pennsylvania State University, University Park, Pennsylvania

Neerav Shah
Glenn Research Center, Cleveland, Ohio

Jonathan S. Litt
U.S. Army Research Laboratory, Glenn Research Center, Cleveland, Ohio

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at 301-621-0134
- Telephone the NASA Access Help Desk at 301-621-0390
- Write to:
NASA Access Help Desk
NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076



Hierarchical Discrete Event Supervisory Control of Aircraft Propulsion Systems

Murat Yasar, Devendra Tolani, and Asok Ray
The Pennsylvania State University, University Park, Pennsylvania

Neerav Shah
Glenn Research Center, Cleveland, Ohio

Jonathan S. Litt
U.S. Army Research Laboratory, Glenn Research Center, Cleveland, Ohio

Prepared for the
First Intelligent Systems Technical Conference
sponsored by the American Institute of Aeronautics and Astronautics
Chicago, Illinois, September 20–22, 2004

National Aeronautics and
Space Administration

Glenn Research Center

Acknowledgments

This work has been supported in part by NASA Glenn Research Center under Grant No. NNC04GA49G, and the Army Research Office (ARO) under Grant No. DAAD19-01-1-0646.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at <http://gltrs.grc.nasa.gov>

Hierarchical Discrete Event Supervisory Control of Aircraft Propulsion Systems

Murat Yasar, Devendra Tolani, and Asok Ray
The Pennsylvania State University
University Park, Pennsylvania 16802

Neerav Shah
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Jonathan S. Litt
U.S. Army Research Laboratory
Glenn Research Center
Cleveland, Ohio 44135

This paper presents a hierarchical application of Discrete Event Supervisory (DES) control theory for intelligent decision and control of a twin-engine aircraft propulsion system. A dual layer hierarchical DES controller is designed to supervise and coordinate the operation of two engines of the propulsion system. The two engines are individually controlled to achieve enhanced performance and reliability, necessary for fulfilling the mission objectives. Each engine is operated under a continuously varying control system that maintains the specified performance and a local discrete-event supervisor for condition monitoring and life extending control. A global upper level DES controller is designed for load balancing and overall health management of the propulsion system.

I. Introduction

Discrete-event dynamic behavior of physical plants is often modeled as regular languages^{1,2} that can be realized by finite-state automata.³ This paper focuses on the development of intelligent decision and control algorithms based on the theory of Discrete Event Supervisory (DES) control on twin-engine aircraft propulsion systems. This work extends the research in the DES field and adds validity to its theory.

The DES control system is designed to be hierarchically structured. Each engine's continuously varying controller interacts with its own local DES controller for detailed health monitoring and intelligent control. The operational information is abstracted and reported to the coordinator for propulsion or mission level DES control of the two engines. Furthermore, the propulsion level DES controller allows for interaction with external inputs such as those from a pilot, aircraft components (e.g. flight control, etc.), and other vehicle management systems, resulting in flexibility for making on-line modifications in the mission objectives. The uniqueness of this DES control approach is that the control policy can be adaptively updated on-line at both the engine and propulsion coordinator levels, and the system is tolerant of small components faults.

Although DES control has been developed for quite some time, there are only a few application examples.^{4,5} It is primarily because no quantitative analytical tool has been established to help design and evaluate the DES controllers. This research is the first time hierarchical DES control is used on a complex non-linear dynamical system, such as an aircraft.

The objective of the paper is to apply hierarchical Discrete Event Supervisory (DES) control theory for intelligent decision and control of a twin-engine aircraft propulsion system. In this paper, the following has been demonstrated:

- (1) DES control can be used for intelligent decision and control of twin-engine aircraft propulsion systems e.g. the problem of load balancing between various engines.
- (2) Implementation of DES control can reduce the accumulated engine damage and thus extend the life of the aircraft engine.
- (3) DES control is helpful in improving the overall mission success and operational behavior.

In this paper the terms controller and supervisor are used interchangeably, also the phrases “Upper Level” and “Propulsion Level” are synonymous and similarly the phrases “Lower Level” and “Engine Level” are synonymous. The paper is organized in six sections including the present one. Section II describes the simulation setup. Section III summarizes the DES control techniques. Section IV discusses the design of the engine level and propulsion level DES controller. The simulation results are examined in section V, and the paper is concluded in section VI.

II. Simulation Setup

In this section the implementation of Discrete Event Supervisory (DES) control on a commercial turbofan engine model simulation is discussed. The objective of such an application is to validate the hierarchical DES control technique with a real world complex non-linear dynamical system. The overall system structure will be explained first, followed by a description of the two hierarchical levels of the system including their internal components.

A. Overall Simulation Testbed Architecture

A hierarchical DES controlled propulsion system has been designed and tested on a simulation test bed that consists of three networked computers using the client/server concept, refer to Figure 1. Computer 1 is the upper level propulsion system coordinator responsible for load balancing, intelligent decision making and health monitoring of the engines. Each of the other two computers, Computer 2 and Computer 3, runs its own copy of a gas turbine engine simulation (differing from each other only through such performance-related parameters as component efficiencies, etc.), including its continuously varying control system and a local discrete-event supervisor. Each of the engine simulators integrates the event-driven discrete dynamics modeled by a finite-state automaton as well as time-driven continuous dynamics modeled by ordinary differential equations through continuous-discrete and discrete-continuous interfaces.⁶ The test bed is capable of simulating different dynamics for each individual engine based on performance-related parameters and operating conditions.

As can be seen in Figure 1, the simulation architecture can be expanded to include more engines that would run in parallel with Computers 2 and 3. It could also be expanded to include higher levels of control. For example, there could be a level above Computer 1 that controls propulsion in parallel with flight control. In this sense, hierarchical DES could be applied to any number of complex systems and missions.

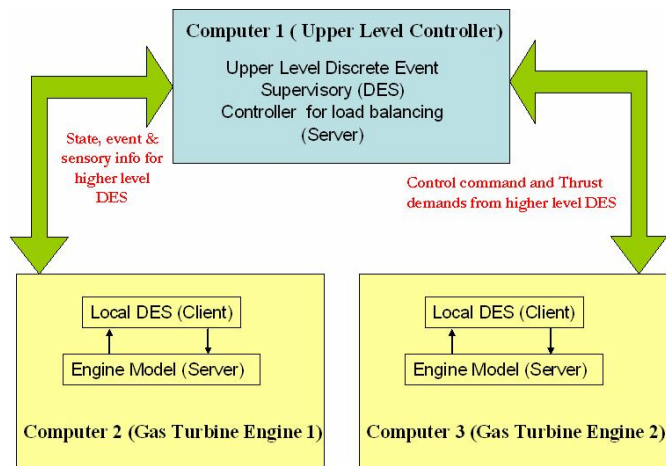


Figure 1.—Overall Testbed architecture.

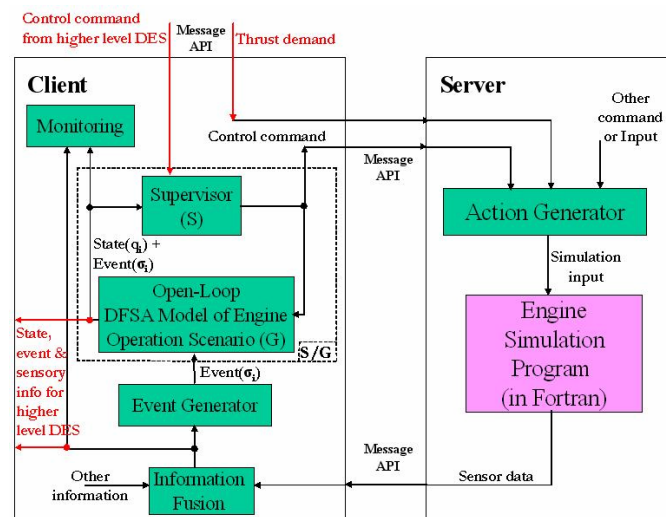


Figure 2.—Engine Level Plant/DES controller.

B. Engine Level DES Control

Figure 2 shows the architecture of the engine level plant and DES controller implementation, which is replicated on Computer 2 and Computer 3. This section will briefly address the main points. For more detailed explanations, refer to reference 6.

In order to implement the DES control in tandem with the gas turbine engine simulation program (written in FORTRAN), a C++ interface was written. The C++ wrapper interfaces the major inputs/outputs of the engine simulation and makes it function just like a simulation running in the C++ environment.

The turbofan engine simulation was designed for full flight envelope operation. With the proper inputs such as power lever angle (PLA or pilot throttle input), and ambient condition (altitude and speed or temperature and pressure), the FORTRAN simulation program simulates the complex operation of the engine from transient to some steady state running condition with large order differential and difference equations. This simulation is a stand-alone program with its own continuous time gain scheduled controller. From the engine simulation, important sensor data such as combustion chamber temperature and high-pressure turbine speed, together with other important information, such as the simulation step size and the simulation cycle number are collected by the C++ wrapper program exchanged with the DES controller through the Message API (Application Programming Interface) communication routine.

The engine level DES controller design has two important components, refer to Figure 2. One is the Event Generator and the other is the Action Generator (which resides on the Server). The Event Generator is the module which receives continuous time sensor data from the plant. This data along with other information, such as estimated state and external inputs are used by this module to generate events, which in turn are inputs to an open-loop DES model of engine operation. The open-loop DES model is constructed based on the operating scenario. The details of the modeling are discussed in section IV. The state-based Deterministic Finite State Automata (DFSA) model serves as state estimator and provides important state and event (both controllable and uncontrollable) information for the discrete-event supervisor to take proper action. Event behavior in the state-based DFSA model only depends on the state where the event is defined and does not depend on the history of how the state was reached.

The DES controller represents the control policy applied to the DFSA model of engine operation, and it could be an ad hoc DES controller based on experience or a discrete-event supervisor designed by the control techniques discussed in section III and IV. The DES controller takes the estimated state as input and generates control commands (controllable event disabling or enabling) as outputs. The control commands are sent through a Message API communication routine to the Action Generator. The primary task of Action Generator is to convert control commands from the supervisor into necessary simulation input for the continuously varying plant (altitude and mach).

C. Propulsion Level DES Control

Figure 3 shows the structure of the propulsion level DES controller together with its own Event Generator. This structure is implemented on a separate computer (Computer 1) which uses the messaging interface to communicate with the other two computers. The Higher Level Supervisor is a DES controller. If expanded graphically, it would be similar to the Client on the Engine Level design.

The software architecture of the simulation test bed is flexible to adapt arbitrary DFSA models and controller designs which make it portable to other complex dynamic systems.

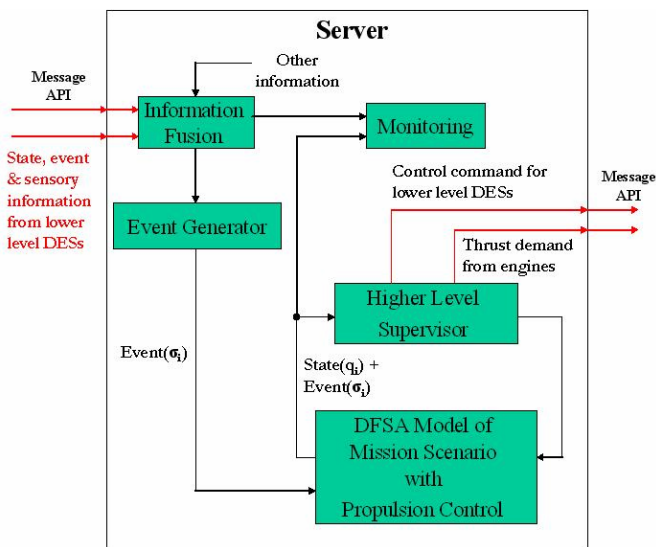


Figure 3.—Propulsion Level DES controller.

III. Review of Language Measure Concepts

This section reviews the previous work on language measure.^{7,8} It provides the background information necessary to develop a performance index.

Let the dynamical behavior of a physical plant be modeled as a deterministic finite state automaton (DFSA) $G_i \equiv (Q, \Sigma, \delta, q_i, Q_m)$ with $|Q| = n$ and $|\Sigma| = m$. Here $Q = \{q_0, q_1, \dots, q_n\}$ is a finite set of states with q_0 the initial state; Σ is a finite alphabet of events; δ is a function from $Q \times \Sigma$ to Q (the transition function); and $Q_m \subseteq Q$ is the set of marked states. Additionally, Σ^* is the set of all strings over Σ .

Definition 1: A DFSA G_i , initialized at $q_i \in Q$, generates the language $L(G_i) \equiv \{s \in \Sigma^* : \delta^*(q_i, s) \in Q\}$ and its marked sublanguage $L_m(G_i) \equiv \{s \in \Sigma^* : \delta^*(q_i, s) \in Q_m\}$.

Definition 2: The language of all strings that, starting at $q_i \in Q$, and terminating at $q_j \in Q$, is denoted as $L(q_i, q_j)$.

Definition 3: The characteristic function that assigns a signed real weight to state-partitioned sublanguages $L(q_i, q_j)$ is defined as: $\chi : Q \rightarrow [-1, 1]$ such that

$$\chi_j \equiv \chi(q_j) \in \begin{cases} [-1, 0) & \text{if } q_j \in Q_m^- \\ \{0\} & \text{if } q_j \notin Q_m \text{ independent of } q_i \\ (0, 1] & \text{if } q_j \in Q_m^+ \end{cases}$$

The $(n \times 1)$ characteristic vector is denoted as:

$$\bar{\chi} \equiv [\chi_1 \ \chi_2 \ \cdots \ \chi_n]^T.$$

Definition 4: The event cost is defined as $\tilde{\pi} : \Sigma^* \times Q \rightarrow [0, 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma, \forall s \in \Sigma^*$,

- $\tilde{\pi}[\sigma_k | q_j] = 0$ if $\delta(q_j, \sigma_k)$ is undefined; $\tilde{\pi}[\varepsilon | q_j] = 1$, where ε is the empty (zero length) string;
- $\tilde{\pi}[\sigma_k | q_j] \equiv \tilde{\pi}_{jk} \in [0, 1]$; $\sum_k \tilde{\pi}_{jk} < 1$;
- $\tilde{\pi}[\sigma_k s | q_j] = \tilde{\pi}[\sigma_k | q_j] \tilde{\pi}[s | \delta(q_j, \sigma_k)]$.

The $(n \times m)$ event cost matrix is denoted as: $\tilde{\Pi} \equiv [\tilde{\pi}_{ij}]$.

Definition 5: The state transition cost of the DFSA is defined as a function $\pi : Q \times Q \rightarrow [0, 1]$ such that $\forall q_j, q_k \in Q, \pi(q_k | q_j) = \sum_{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k} \tilde{\pi}(\sigma | q_j) \equiv \pi_{jk}$ and $\pi_{jk} = 0$ if $\{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k\} = \emptyset$.

The $n \times n$ state transition cost matrix, denoted as Π , is defined as:

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2n} \\ \vdots & & \ddots & \vdots \\ \pi_{n1} & \pi_{n2} & \cdots & \pi_{nn} \end{bmatrix}$$

Definition 6: The signed real measure μ of a singleton string set $\{s\}$ is defined as:

$$\mu(\{s\}) \equiv \chi(q_j) \tilde{\pi}(s | q_i) \quad \forall s \in L(q_i, q_j) \subseteq L(G_i).$$

The signed real measure of $L(q_i, q_j)$ is defined as:

$$\mu(L(q_i, q_j)) \equiv \left(\sum_{s \in L(q_i, q_j)} \mu(\{s\}) \right)$$

The signed real measure of a DFSA G_i , initialized at the state $q_i \in Q$, is defined as

$$\mu_i \equiv \mu(L(G_i)) = \sum_j \mu(L(q_i, q_j)).$$

The $n \times 1$ real signed measure vector is denoted as:

$$\bar{\mu} \equiv [\mu_1 \ \mu_2 \ \cdots \ \mu_n]^T$$

In reference 5, it has been shown that the measure of the language $L(G_i)$, where $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ can be expressed as: $\mu_i = \sum_j \pi_{ij} \mu_j + \chi_i$. Equivalently, in vector notation: $\bar{\mu} = \Pi \bar{\mu} + \bar{\chi}$. Since Π is a contraction operator, the measure vector $\bar{\mu}$ is uniquely determined as:

$$\bar{\mu} = [I - \Pi]^{-1} \bar{\chi} \tag{1}$$

IV. DES controllers

One of the major tasks of this paper is fusion of the (possibly) redundant, conflicting and incomplete information to make timely decisions. Such information can be derived from different types of sensor data as well as operational history and the knowledge base generated from pilots' personal experience. Computer-based advanced analytical techniques are necessary for fusion of the time series data available from multiple sensors and the relevant non-sensor-based information (e.g. weather data, analytical damage model data) to make specific inferences that could not be achieved by the use of single sensors alone. However, improved performance may not result simply from an increased volume of sensor data and engine information unless the ensemble of information is systematically processed in the context of the engine operational conditions and mission objectives. In essence, fusion of the heterogeneous information is necessary to:

- (1) Guarantee improvement of resolution and reduction of ambiguity in decision and control.
- (2) Make advantageous trade-offs between probability of false alarms and missed detections.

The open-loop discrete event dynamics are modeled as a Discrete Finite State Automaton (DFSA) based on the postulated engine operating scenario. The model may vary for different mission scenarios. The DFSA plant model assumes that an aircraft equipped with twin turbofan engines is carrying out a routine surveillance mission. Abortion of the mission is allowed at certain states when an anomaly is detected in the engine. Each engine of the aircraft is equipped with a continuous time controller which is supervised by a local DES controller. The primary objective of the local Engine level DES controller is to strike the right balance between the conflicting demands of higher performance from a higher level DES and limiting the damage (stress-related damage modes) to the engine. The upper Propulsion level DES controller redistributes the load depending on the health of the engine and thrust demand of the pilot.

A. Engine Level DES control

Figure 4 presents the DFSA model of the engine operation discrete-event supervisory control for an individual engine, whereas Figure 5 shows the associated DES controller, which is derived from the engine model. The state and the event list are presented as tables A-1 through A-4 in the appendix of the paper (C denotes controllable events and UC denotes uncontrollable events). The engine can operate in two regimes, one is a high performance regime (state q_3) where the damage rate is high, and the other is a low performance regime (state q_5) where the damage rate is low. In the high performance regime the engine has a tendency of going to state q_4 , where state variables like combustor temperature have been observed to have oscillatory behavior. This is extremely harmful in terms of engine health and must be avoided. The engine level controller chooses the regime of operation (q_3 or q_5) depending on two factors: (1) Thrust demand from upper level DES, and (2) Health of the engine (as formulated below).

In the context of this paper, "health" specifically refers to structural health, i.e. the remaining life of the engine or its components. Life is consumed by thermal and/or structural loads that result in engine "damage." The health of the engine is determined by the damage accumulated over the period of operation.

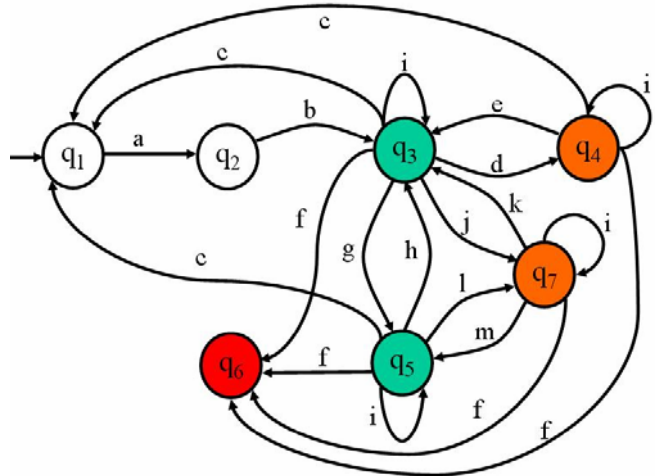


Figure 4.—Engine Level DES plant model.

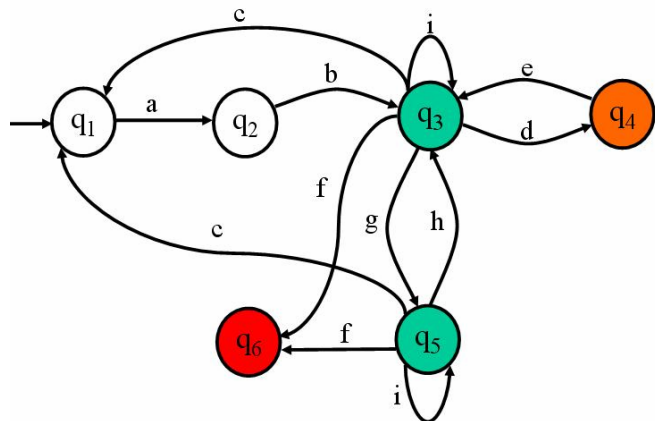


Figure 5.—Engine Level DES controller model.

Damage accumulation has two components. It is formulated as a function of high-pressure turbine gas inlet temperature and shaft speed. In addition, at random time intervals, damage spikes (sudden jumps) are introduced to simulate sudden events of unspecified cause resulting in damage accumulation in the engine.

B. Propulsion level DES Control

The main task of the upper level DES controller is to redistribute the load between two engines depending on the current health of each engine and the thrust demand of the pilot. The states and the event list are presented in table A-3 and A-4 of the appendix. A graphical representation for the propulsion level DES controller, such as Figure 5 for the engine level controller, is provided in the appendix as Figure A-1. Due to the complexity of the propulsion level system, the figure is not fully labeled.

V. Simulation Experiments: Results and Discussion

The experimentation setup for the DES simulator is designed on the engine simulation test bed to validate the DES control concept. Upon successful implementation of the software modules (discussed in section II) on the client and server computers, two sets of experiments were performed. The first set of experiments was performed on the engine level DES controller to validate that implementation of DES control reduces the engine damage and thus extends engine life. The second set of experiments was performed on the Propulsion Level DES to validate that it judiciously balances engine load for the two engines. Afterwards, the DES controller performance is analyzed.

A. Engine Level DES Test

Both unsupervised and supervised plants are excited by the same predetermined fixed input shown in Figure 6. To make a comparison between the supervised and the unsupervised cases several outputs (Combustor temperature, high pressure turbine speed, net thrust of the engine and fuel flow through the main burner) of the engine simulation were observed over a period of 12 minutes. Figures 7 and 8 show the simulation outputs for the unsupervised and supervised cases respectively.

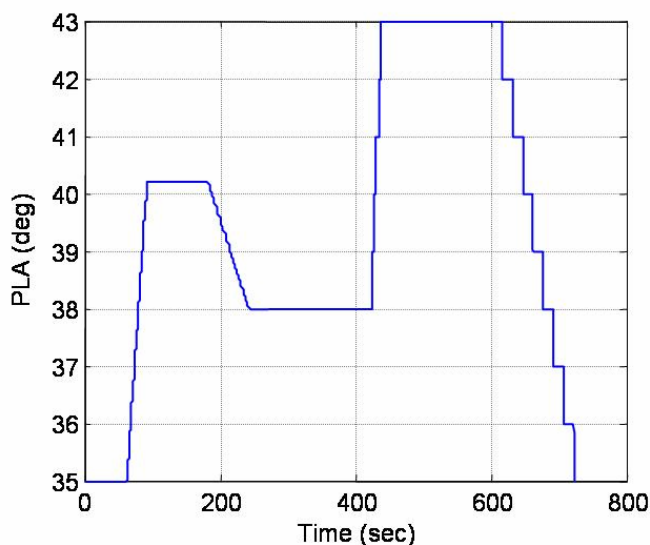


Figure 6.—Power Lever Angle input.

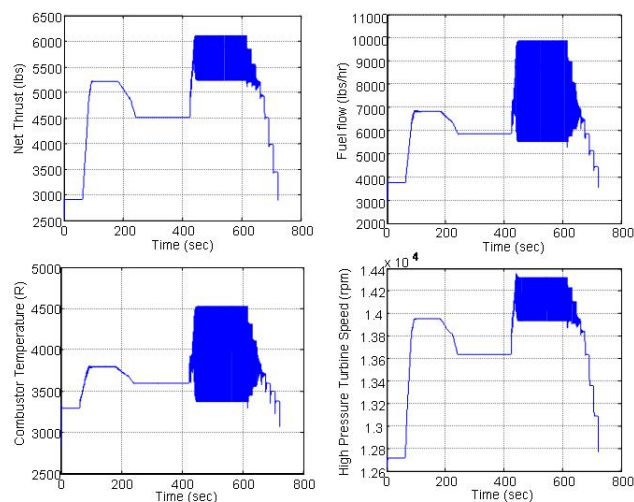


Figure 7.—Simulation output for the unsupervised case.

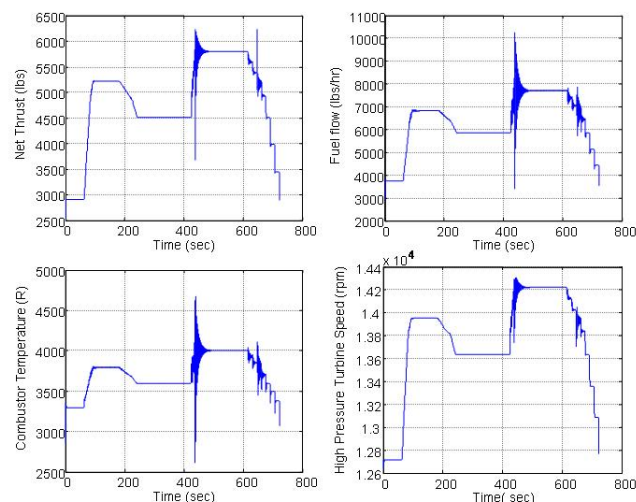


Figure 8.—Simulation output for the supervised case.

The comparison of two figures indicates that the DES controller applied in the engine level eliminates the high frequency oscillations which are observed in the unsupervised case. High frequency oscillations are modeled as the primary source of damage to the turbine blades and adversely affect the overall health of the engine.

B. Propulsion Level DES Test

The propulsion level DES controller has two main tasks. One is the intelligent decision making and control of twin-engine aircraft propulsion systems and the second is to improve the overall mission and operational behavior so that engine health is maintained. These can be demonstrated through intelligent load balancing. The issue of load balancing becomes especially important when the health conditions of two engines are significantly different (one can be called in “bad” condition and in “good” condition). For this scenario, the aim of the DES controller is judicious redistribution of the load between two engines such that the “bad” engine carries lower load than the “good” one, subject to the condition that the total thrust output of the engines remains same. Figures 9 and 10 show the simulation inputs/outputs of each of the two engines. Initially in region (1), both engines are in “good” condition. As the mission progresses, Engine 2 suffers a sudden damage-inducing event, i.e. the condition of Engine 2 becomes “bad” (as indicated by the discrete damage increment spike at approx 325s; this unexplained event facilitates the demonstration of the Propulsion level DES). At this point (entering region (2)), the propulsion level DES redistributes the load such that Engine 1 carries a higher load than the damaged Engine 2, (as can be seen by comparing Figures 9 and 10). At approx 625s, Engine 1 goes “bad” (as indicated by the discrete damage increment spike on Figure 9). Entering region (3), both Engine 1 and Engine 2 are considered to be in “bad” condition, thus the propulsion level DES controller balances the load approximately equally between the two.

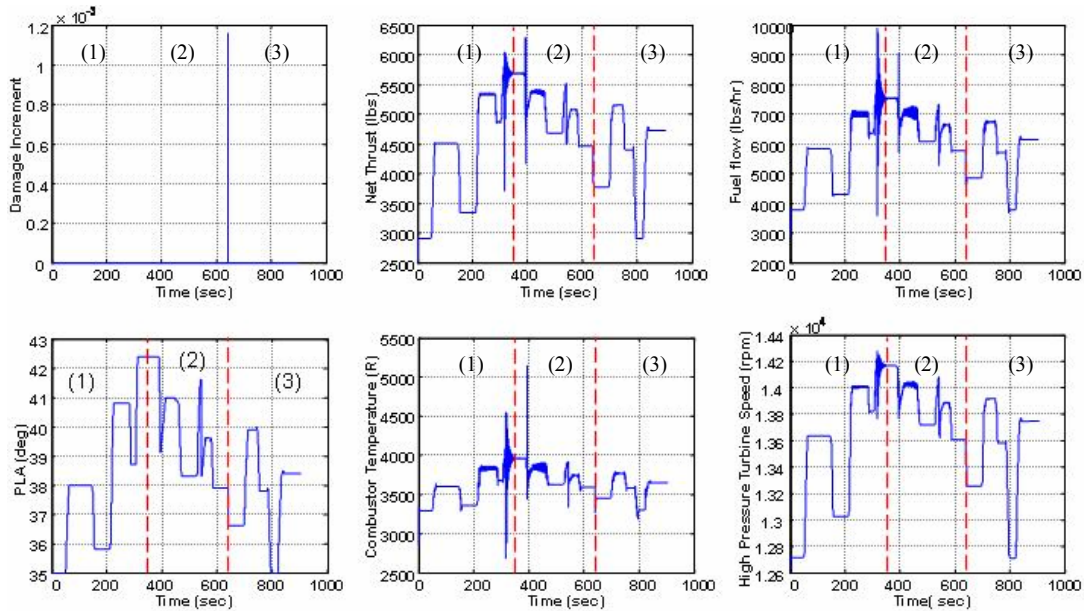


Figure 9.—Effect of propulsion level DES controller on Engine 1.

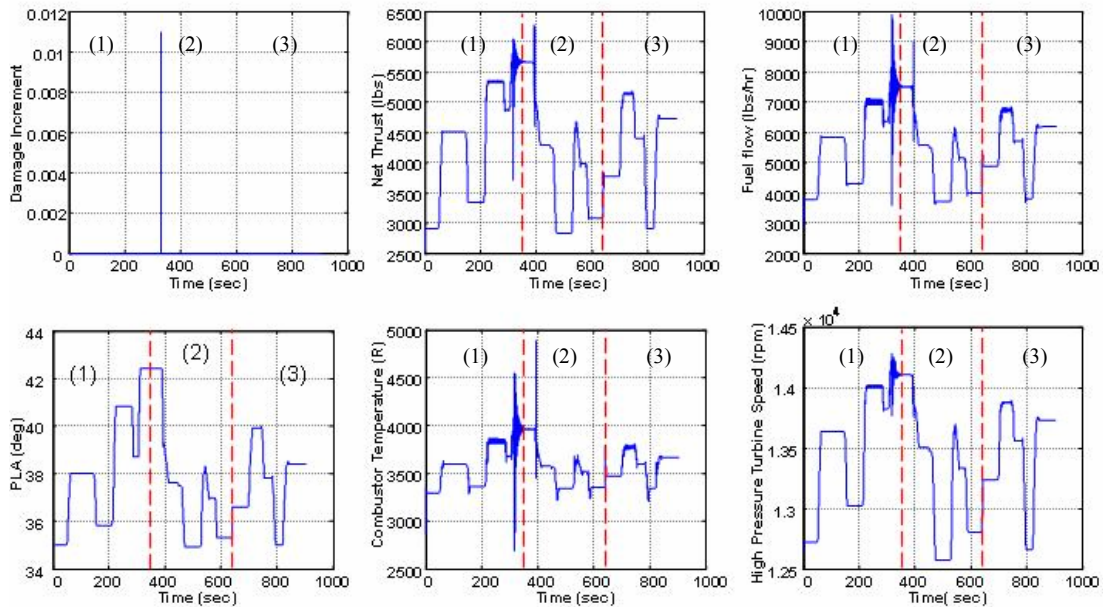


Figure 10.—Effect of propulsion level DES controller on Engine 2.

C. DES Controller Performance Measure

To observe the impact of the propulsion level DES controller on the overall mission, the concept of “language measure,” described in section III is used. Language measure gives a theoretical performance measure (equation (1) from section III) of the controllers, given the state weights (χ) and the state transition probabilities (Π). To form the state transition probability matrix Π , 50 experimental runs were conducted for both supervised and unsupervised cases. The Π matrices for the engine level and propulsion level DES controllers are presented in the appendix as Table A-5 and A-6.

Given the vector for the state weights of the engine level as $\chi = [0 \ 0 \ 0.2 \ -0.1 \ 0.2 \ -0.5]$ (designed by the control engineer), the theoretical language measure for the unsupervised plant and supervised plant are computed from equation (1) to be:

$$\mu_{\text{unsupervised}} = 4.603 \text{ and}$$

$$\mu_{\text{supervised}} = 6.3678.$$

Improved performance is a positive gain in the language measure. Therefore, the engine level DES controller improves the performance of the open loop plant.

For the propulsion level DES controller the weights of the states are selected according to their importance to the mission management as $\chi = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ -0.2 \ 0.2 \ 0 \ 0]$. Table 1 shows the

experimental results of the number of visits to the states which have weights other than 0. These are State 8: Plane destroyed, State 9: Mission abort, State 10: Mission successful. They have relative weights of -1, -0.2 and 0.2 respectively. The language measures, i.e. the theoretical performance of the controllers are calculated as:

$$\mu_{\text{unsupervised}} = -2.7999 \text{ and}$$

$$\mu_{\text{supervised}} = -1.6003,$$

which indicate that the propulsion level DES controller improves mission performance.

Experimental outcomes, shown in Table 1, can also be used to evaluate the DES controller performance directly by multiplying each state visit with the relative weight of the state. The experimental performances of the propulsion level controllers are:

$$\text{Unsupervised} : 1 \times (-1) + 15 \times (-0.2) + 34 \times (0.2) = 2.8$$

$$\text{Supervised} : 1 \times (-1) + 8 \times (-0.2) + 41 \times (0.2) = 5.6.$$

Once again, a positive gain is observed, which confirms that the supervised plant performs better than the unsupervised plant, indicating an overall improvement of the mission performance. A similar experimental calculation for the engine level controllers is not shown because there is no mission behavior element in the engine level.

VI. Conclusions

This paper presents a quantitative approach to synthesis of a hierarchical discrete-event supervisory (DES) control of an aircraft propulsion system. The DES control law has been validated on a networked simulation test bed. The plant dynamics in the simulation test bed are built upon the model of a generic gas turbine engine. The software architecture of the simulation test bed is flexible to adapt arbitrary DFSA models and controller designs and to fit other complex systems. The supervisory control laws are quantitatively analyzed using a language measure.^{7,8}

The long-term objectives of the proposed approach are to achieve: (i) intelligent decision and control of distributed propulsion management systems, where each of the engines has its own local DES control; (ii) structural damage reduction and life extension of aircraft engines without any significant loss of the system performance; and (iii) decision making and mission planning modifications through a high-level DES coordinator; (iv) extension of this work to other complex dynamic systems such as Rotorcraft, power plants, and spacecraft simulation test-beds. Future work includes incorporation of optimal control laws.⁵

Table 1.—Simulated Performance of Controllers.

STATE NUMBER	NUMBER OF STATE VISITS	
	Supervised	Unsupervised
8	1	1
9	8	15
10	41	34

Appendix

Table A-1.—State List for the engine level DES supervisor.

Name	Description	Status
q₁	Engine Start	
q₂	Engine Warm up	
q₃	High Performance/ High damage rate	Marked (good)
q₄	Oscillations	Marked (bad)
q₅	Low Performance/ low damage rate	Marked (good)
q₆	Engine inoperable	Marked (worst)
q₇	Low Performance/ high damage rate	Marked (bad)

Table A-2.—Event List for the engine level DES supervisor.

Name	Description	Status
a	Start	C
b	Warm up complete	UC
c	Shut down the engine	UC
d	Detection of oscillations	UC
e	Nozzle area reduction	C
f	Engine fails	UC
g	Reduce performance /reduce damage	C
h	Increase performance/ increase damage	C
i	Remain in the state	C
j	Reduce performance	C
k	Increase performance	C
l	Increase damage rate	C
m	Decrease damage rate	C

Table A-3.—State List for the propulsion level DES supervisor.

State	Description	Status
q₁	Engines on ground	
q₂	Engines warming up	
q₃	Both engines in High Performance operation	
q₄	One engine in High one engine in Low Performance	
q₅	Both engines in Low Performance operation	
q₆	One engine stopped one engine in High Performance	
q₇	One engine stopped one engine in Low Performance	
q₈	Both engines failed	bad
q₉	Decision for abort mission	bad
q₁₀	Mission successful	good
q₁₁	High damage detected for one engine	
q₁₂	High damage detected for both engines	

Table A-4.—Event List for the propulsion level DES supervisor.

Event	Description	Status
A	Start engine	C
B	Warm up complete	UC
C	One engine deteriorates	UC
D	Redistribute the load	C
E	Both engines deteriorate	UC
F	One good engine fails	UC
G	Both engines fail	UC
H	Increase performance	C
I	Reduce performance	C
J	Request to abort mission	C
K	Request accepted	UC
L	Request rejected	UC
M	Mission accomplished	UC
N	Turn off engines	UC
O	Redistribute load	C
P	Redistribute load	C
Q	Request rejected	UC
R	One bad engine fails	UC

Table A-5.—Engine Level II matrix.

	a	b	c	d	e	f	g	h
q ₁	1	0	0	0	0	0	0	0
q ₂	0	1	0	0	0	0	0	0
q ₃	0	0	0	0.1938	0	0.015504	0.7907	0
q ₄	0	0	0	0	0.96	0.04	0	0
q ₅	0	0	0.16667	0	0	0	0	0.83333
q ₆	0	0	0	0	0	0	0	0

Table A-6.—Propulsion Level II matrix.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
q ₁	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q ₂	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q ₃	0	0	0.091	0	0	0.045	0	0	0.864	0	0	0	0	0	0	0	0	0
q ₄	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q ₅	0	0	0.045	0	0	0	0	0.773	0	0	0	0	0.182	0	0	0	0	0
q ₆	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
q ₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q ₈	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q ₉	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
q ₁₀	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
q ₁₁	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0.3	0	0	0
q ₁₂	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

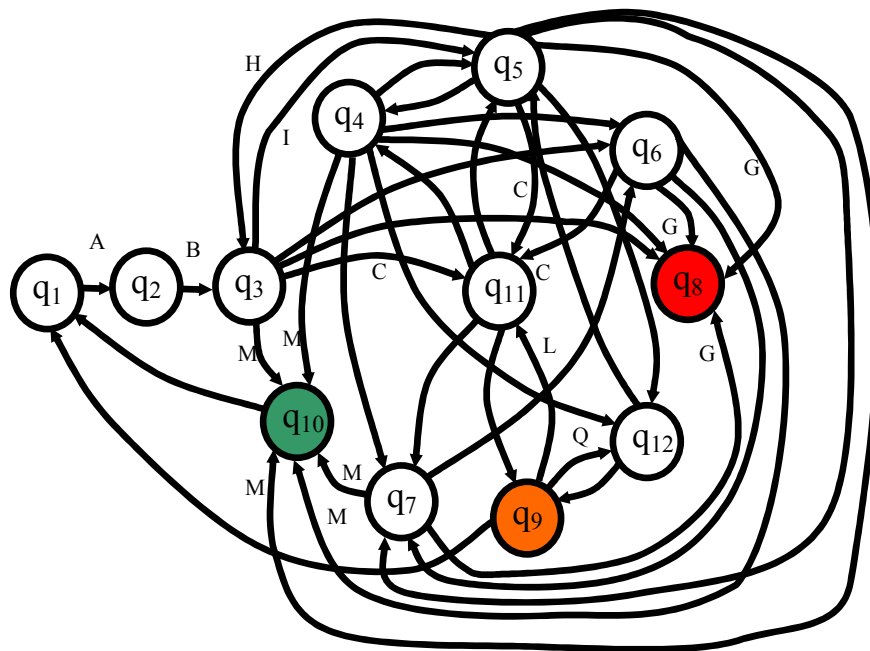


Figure A-1.—Propulsion Level DES controller
(NOTE: not all state transitions are labeled due to complexity).

References

- ¹Hopcraft, J.E., Ullman, J.D., *Introduction to Automata Theory, languages, and Computation.*, Addison-Wesley, 1979.
- ²Martin, J. C., *Introduction to the Languages and the Theory of Computation.*, McGraw Hill, 2003.
- ³Ramadge, P.J., Wonham, W. M., "Supervisory control of a class of discrete event processes," *SIAM J. Control and Optimization*, Vol. 25, No. 1, January 1987, pp. 206-230.
- ⁴Zhang, H., Ray, A., and Phoha, S., "Hybrid Life Extending Control of Mechanical Systems: Experimental Validation of the Concept," *Automatica*, Vol. 36, No. 1, January 2000, pp. 23-36.
- ⁵Fu, J., Ray, A., Lagoa, C., "Unconstrained Optimal Control of Regular Languages," *Automatica*, Vol. 40, No. 4, April 2004, pp. 639-646.
- ⁶Fu, J., Yasar, M., and Ray, A., "Optimal Discrete Event Supervisory Control of Aircraft Gas Turbine Engines," *American Control Conference*, Boston, Massachusetts, June 30th to July 2nd 2004.
- ⁷Surana, A., and Ray, A., "Signed Real Measure of Regular Languages," *Demonstratio Mathematica*, Vol. 37, No. 2, June 2004, pp. 485-503.
- ⁸Wang, X., and Ray, A., "A Language Measure for Performance Evaluation of Discrete Event Supervisory Control Systems," *Applied Mathematical Modelling*, 2004, in press.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 2004		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE Hierarchical Discrete Event Supervisory Control of Aircraft Propulsion Systems			5. FUNDING NUMBERS WBS-22-303-30-72 1L161102AF20	
6. AUTHOR(S) Murat Yasar, Devendra Tolani, Asok Ray, Neerav Shah, and Jonathan S. Litt				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-14856	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001 and U.S. Army Research Laboratory Adelphi, Maryland 20783-1145			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-2004-213376 ARL-TR-3311 AIAA-2004-6469	
11. SUPPLEMENTARY NOTES Prepared for the First Intelligent Systems Technical Conference sponsored by the American Institute of Aeronautics and Astronautics, Chicago, Illinois, September 20-22, 2004. Murat Yasar, Devendra Tolani, and Asok Ray, Pennsylvania State University, University Park, Pennsylvania 16802; Neerav Shah, NASA Glenn Research Center; Jonathan S. Litt, U.S. Army Research Laboratory, NASA Glenn Research Center, Cleveland, Ohio. Responsible person, Jonathan S. Litt, organization code 5530, 216-433-3748.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category: 63 Available electronically at http://gltrs.grc.nasa.gov This publication is available from the NASA Center for AeroSpace Information, 301-621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper presents a hierarchical application of Discrete Event Supervisory (DES) control theory for intelligent decision and control of a twin-engine aircraft propulsion system. A dual layer hierarchical DES controller is designed to supervise and coordinate the operation of two engines of the propulsion system. The two engines are individually controlled to achieve enhanced performance and reliability, necessary for fulfilling the mission objectives. Each engine is operated under a continuously varying control system that maintains the specified performance and a local discrete-event supervisor for condition monitoring and life extending control. A global upper level DES controller is designed for load balancing and overall health management of the propulsion system.				
14. SUBJECT TERMS Discrete event supervisory control; DES; Life extending control; Intelligent control; Aircraft propulsion system; Aircraft engine control; Discrete systems; Supervisor; Mission planning; Finite-state automation			15. NUMBER OF PAGES 21	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

