Optimal Discrete Event Supervisory Control of Aircraft Gas Turbine Engines[†]

Jinbo Fu, Murat Yasar, Asok Ray
Mechanical Engineering Department
The Pennsylvania State University
University Park, PA 16802

Keywords: Optimal Control, Discrete Event Systems, Supervisory Control, Language Measure, Gas Turbine Engine

Abstract

This paper presents an application of the recently developed theory of optimal Discrete Event Supervisory (DES) control that is based on a signed real measure of regular languages. The DES control techniques are validated on an aircraft gas turbine engine simulation test bed. The test bed is implemented on a networked computer system in which two computers operate in the client-server mode. Several DES controllers have been tested for engine performance and reliability.

1 Introduction

Discrete-event dynamic behavior of physical plants is often modeled as regular languages that can be realized by finite-state automata [RW87]. The sublanguage of a controlled physical plant may be different under different supervisors such that a partially ordered set of sublanguages requires a quantitative measure for total ordering of their respective performance. To address this issue, Wang and Ray [WR04] have formulated a signed measure of regular languages followed by Surana and Ray [SR03] who have constructed a metric space of sublanguages based on the total variation measure of the language. Based on the language measure, Fu et al. have reported optimal control of regular languages in a series of papers [FRL04] [FRL03] [FLR03].

This paper presents an application of the recently developed theory of optimal Discrete Event Supervisory (DES) control [FRL04], which addresses state-based optimal supervisory control of a gas turbine engine without considering the event disabling cost. The performance index of the optimal control policy is a signed language measure of the supervised sublanguage, which is expressed in terms of a state transition cost matrix and a characteristic vector [WR04] [SR03]. In this application, a new Deterministic Finite State Automaton (DFSA) plant model is developed to represent the discrete-event dynamical behavior of a generic gas turbine engine. It is a component level engine model that represents the nonlinear dynamics of real engine operation. However, this model does not capture various aspects of the engine's abnormal operations (e.g., low oil pressure, high bearing vibration, and foreign object impact), which are of paramount interest to the pilot. For proper decision and control, this information is necessary, and was obtained by running various engine scenario simulations and compiling information regarding pilot experience. The resulting DFSA plant model was used to design an optimal DES control system for the engine operation.

The engine simulation test bed is implemented on a networked system, where two computers operate in the client-server mode. The plant (i.e., engine operation) computer is the client and executes the engine simulation. The control computer is the server and executes the tasks of the DES control and other ancillary functions such as information display. Several DES controllers, including the unsupervised plant (i.e., the engine without DES control), have been tested for comparison of engine performance and reliability. To our best knowledge, this is the first time the theory of optimal DES control has been applied to a large-scale complex system.

The paper is organized in six sections including the present one. Section 2 reviews the salient concepts of the language measure. Section 3 summarizes the DES control techniques. Section 4 discusses the implementation of DES control on the engine simulation. In section 5, the experiments carried out are discussed, and the experimental results are examined. The paper is summarized and concluded in Section 6.

2 Brief Review of the Language Measure

This section reviews the previous work on language measure [WR04] [SR03]. It provides the background information necessary to develop a performance index and an optimal control policy.

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 $|\Sigma| = m$.

[†] This work has been supported in part by the United States Army Research Office under Grant No. DAAD19-01-1-0646 and NASA Glenn Research Center under Grant Nos. NAG3-2448 and NNC04GA49G.

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: $\chi : Q \to [-1,\ 1]$ such that

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

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

$$\bar{\chi} \equiv [\chi_1\ \chi_2 \cdots \chi_{\mathrm{n}}]^T$$

Definition 4: The event cost is defined as $\tilde{\pi} : \Sigma^* \times Q \to [0,1)$ 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; $\tilde{\pi}[\varepsilon | q_j] = 1$;
- $\tilde{\pi}[\sigma_k | q_j] \equiv \tilde{\pi}_{jk} \in [0,1)$; $\sum_k \tilde{\pi}_{jk} < 1$;
- $\tilde{\pi}[\sigma_k s | q_j] = \tilde{\pi}[\sigma_k | q_j]\, \tilde{\pi}[s | \delta(q_j, \sigma_k)]$.

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 $\pi : Q \times Q \to [0,1)$ such that $\forall q_j,\ q_k \in Q$,

$$\pi(q_k | q_j) = \sum_{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k} \tilde{\pi}(\sigma | q_j) \equiv \pi_{jk} \text{ and } \pi_{jk} = 0$$

if $\{\sigma \in \Sigma : \delta(q_j, \sigma)\} = \varnothing$. 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: $\mu(\{s\}) \equiv \chi(q_j)\tilde{\pi}(s | q_i)$ $\forall s \in L(q_i, q_j) \subseteq L(G_i)$.

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

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

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

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

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

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

Wang and Ray [WR04] have 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:

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

3 Optimal Discrete Event Supervisory Control

Fu et al. [FRL04] have introduced the concept of unconstrained optimal control of regular languages based on a specified measure. This optimal control technique is designed for plants with insignificant event disabling cost. The state-based optimal control policy is obtained by selectively disabling controllable events to maximize the measure of the controlled plant language without any constraints. In each iteration, the optimal control algorithm attempts to disable all controllable events leading to "bad marked states" and enable all controllable events leading to "good marked states." It is also shown that computational complexity of the control synthesis is polynomial in the number of plant states [FRL04].

Let G be the DFSA plant model without any constraint of operational specifications. Let the state transition cost matrix of the open loop plant be: $\Pi^{plant} \in \Re^{n \times n}$ and the characteristic vector be: $\mathrm{X} \in \Re^n$. Starting with $k = 0$ and $\Pi^0 \equiv \Pi^{plant}$, the control policy is constructed by the following two-step procedure [FRL04]:

Step 1: For every state q_j for which $\mu_j^0 < 0$, disable controllable events leading to q_j. Now, $\Pi^1 = \Pi^0 - \Delta^0$, where $\Delta^0 \geq 0$ is composed of event costs corresponding to all controllable events that have been disabled at $k = 0$.

Step 2: Starting with $k = 1$, if $\mu_j^k \geq 0$, re-enable all controllable events leading to q_j, which were disabled in Step 1. The cost matrix is updated as: $\Pi^{k+1} = \Pi^k + \Delta^k$ for $k \geq 1$, where $\Delta^k \geq 0$ is composed of event costs corresponding to all currently re-enabled controllable events. The iteration is terminated if no controllable event leading to q_j remains disabled for which $\mu_j^k \geq 0$. At this stage, the optimal performance is $\mu^* \equiv [I - \Pi^*]^{-1} \mathrm{X}$.

4 Implementation of the DES Control Concept

This section presents an application of the optimal discrete event supervisory (DES) control for real-time operation of gas turbine engines. The plant under DES control in the simulation test bed is a nonlinear dynamic model of the engine together with its continuously varying multivariable controller. With the proper inputs of power lever angle (PLA) and ambient conditions, the FORTRAN program simulates both steady-state and transient operations of the gas turbine engine in the continuous setting. The objectives are to demonstrate efficacy of DES control for: (1) Structural damage reduction and life extension of aircraft engines with proper discrete command interference; and (2) Decision making and mission planning optimization.

4.1 Architecture of the DES Engine Controller

The DES control is implemented in the C++ environment around the existing engine simulation code that is written in FORTRAN. The plant model code is a stand-alone program with its own continuous-time gain-scheduled robust controller that is kept unaltered. The C++ wrapper of the simulation code takes over the major inputs and outputs of interest and makes them transparent to the end user as if the entire simulation runs in the C++ environment. The advantage of working in the C++ environment is the convenient utilization of the standard Message Application Protocol Interface (API) communication routines. In addition, all other functions (e.g., Event generator, Action Generator, and Supervisor), are implemented in C++.

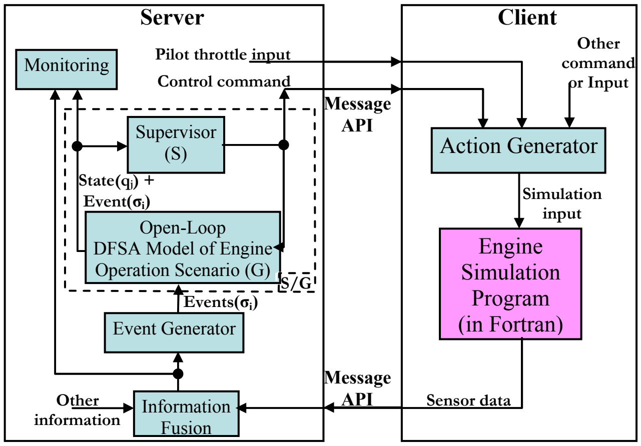


Figure 1 Architecture of the DES control system

Figure 1 shows the architecture of DES control as implemented in the simulation test bed. The DES control is implemented on a pair of networked computers that operate in the server-client mode. The plant model of the aircraft engine and the Action Generator reside in the client computer. The Event Generator and Supervisor are located in the server computer. The pilot commands are entered in the server computer. The server and client communicate with each other over the communication network via API messages as seen in Figure 1. Note that the information Fusion module at the bottom of Figure 1 requires ancillary (e.g., oil pressure and bearing vibration) information in addition to the conventional plant sensor data (e.g., gas temperature and engine shaft speed).

The Action Generator converts the discrete-event symbols of supervisor commands into continuous signals that are inputs to the plant model. The control commands (e.g., flight parameter modifications, compensating throttle inputs, mission abortion, and pilot throttle command) are passed through the Message API communication routine to the Action Generator on the client side. The Action Generator converts control commands from the supervisor into necessary simulation input. Similarly, the Event Generator converts the plant sensor signals and other pertinent information (e.g., engine operational data) into event symbols that are inputs to the supervisor. In essence, the role of Event Generator is fusion of heterogeneous information and real-time expression of the relevant part in the language of the supervisor. The sensor signals are processed with built-in information (e.g. threshold values and fault detection logic) to generate discrete events as the inputs to the supervisor. The DFSA plant model, located in the supervisor, serves as the state estimator.

4.2 DFSA Model of Gas Turbine Engine

The open-loop discrete event dynamics are modeled as a DFSA based on the postulated engine operation scenario. The model may vary for different mission scenarios. The DFSA plant model assumes that a military aircraft equipped with a single turbofan jet engine is carrying out a routine surveillance mission. Abortion of the mission is allowed at certain states when an anomaly is detected in the engine. Three major anomalies are considered: Low Oil Pressure, High Bearing Vibration and High Fatigue Crack Damage Increment.

The plant model has 42 states, of which four are marked states and the alphabet consists of 16 events, of which four are controllable events [RW87]. Table I lists the marked states and controllable events.

Table I Marked States and Controllable Events

Marked States	Controllable Events
42 - Mission aborted on ground	**a** - Start engine
14 - Mission aborted off ground	**f** -Request for abortion
16 - Unexpected engine halt	**i** - Request flight parameter change
15 - Mission successful	**p** - Maintain current condition

5 Simulation Experiments: Results and Discussion

A series of experiments were designed on the engine simulation test bed to validate the DES control concept. Upon successful implementation of the software modules on the client and server computers, the first set of experiments was conducted to verify that functions and communications added on both server and client sides do not affect the simulated engine dynamics. Completion of these experiments partially assures robustness of the DES control relative to exogenous disturbances such as communication delays. The second set of experiments was conducted to compare the engine performance and damage accumulation for the unsupervised (i.e., without DES control) plant and a supervised (i.e., under a selected DES control) plant. Figure 2 shows the given PLA input.

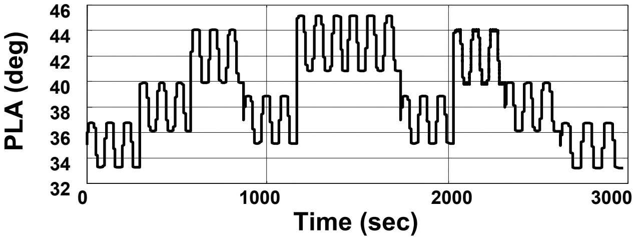
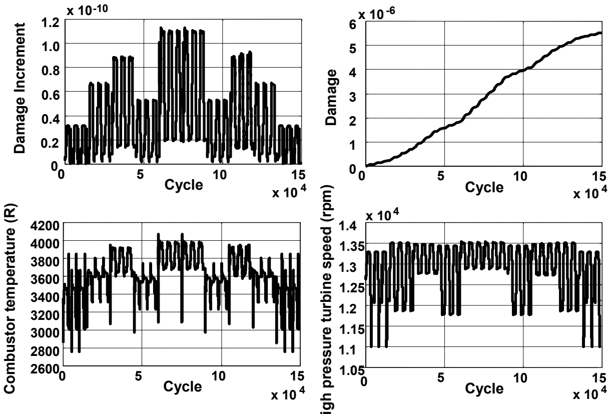
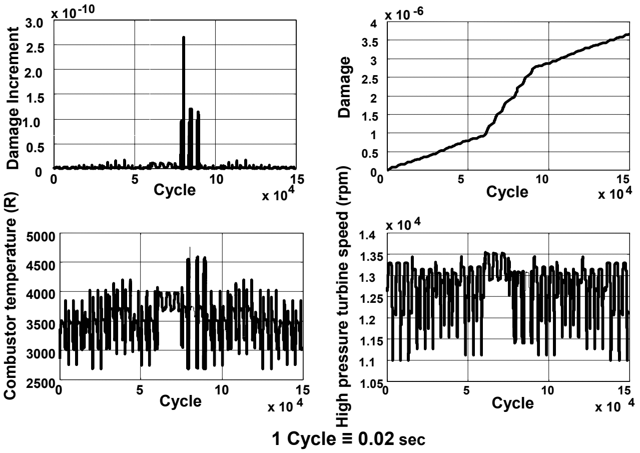


Figure 2 PLA input for the simulation



1 Cycle ≡ 0.02 sec

Figure 3 Unsupervised plant output (fixed PLA input)

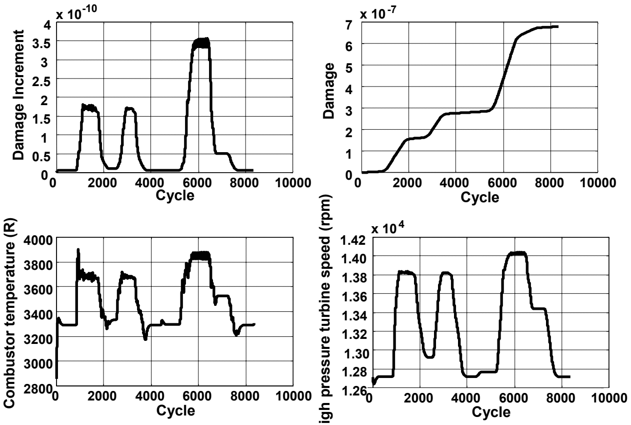


1 Cycle ≡ 0.02 sec

Figure 4 Supervised plant output (fixed PLA input)

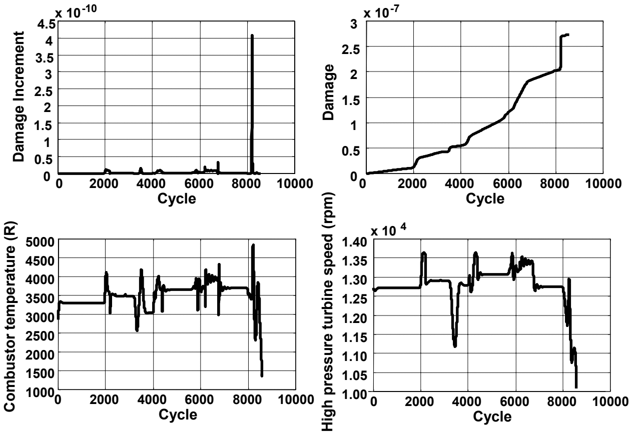
Under the same fixed PLA input, Figures 3 and 4 show the engine outputs for the unsupervised plant and supervised plant, respectively. Figure 4 shows ~35% damage reduction under DES control. Although the given throttle inputs are the same, the supervisor modifies them only if high damage increment is detected. Consequently, input to the engine simulation is adjusted to reduce the damage increment rate for the supervised case. Damage accumulation is formulated as a function of high-pressure turbine gas inlet temperature and shaft speed.

For (statistically identical) random throttle inputs, Figures 5 and 6 show the engine outputs for the unsupervised plant and supervised plant, respectively. Comparison indicates ~60% damage reduction under DES control.



1 Cycle ≡ 0.02

Figure 5 Unsupervised plant output (random PLA input)



1 Cycle ≡ 0.02 sec

Figure 6 Supervised plant output (random PLA input)

5.1 Language measure parameter identification

Analysis and synthesis of an optimal DES controller require the identification of the event cost matrix. Similar to continuously varying dynamical systems (CVDS), we use techniques of system identification to identify the language measure parameters of the DFSA plant model - the elements $\tilde{\pi}_{ij}$ of the event cost matrix $\tilde{\Pi}$ (see Definition 4 in Section 2). As the number of experiments increases, the identified event costs tend to converge in the Cauchy sense. For stationary operation of the engine, since conditional probabilities of the events can be assumed to be time-invariant, the identified event costs and their uncertainty bounds can be determined. Wang et al. [WRK03] have reported details of the identification procedure and its experimental validation on a robotic test bed. As a typical case, Figure 7 presents identification of event costs at state 6 (engine operation under normal damage increment). During 100 experiments, the states visited and the events triggered were monitored and plotted.

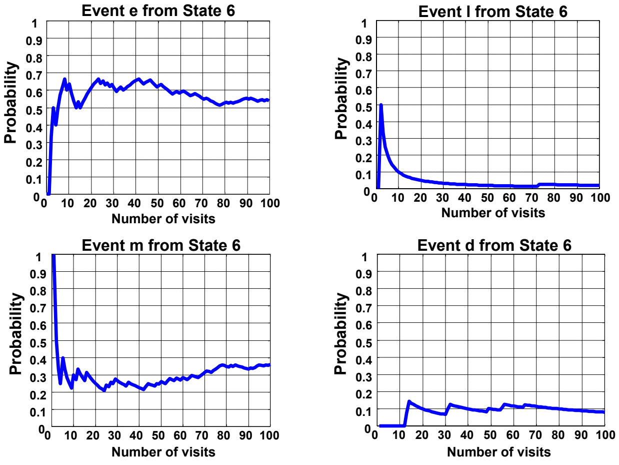


Figure 7 Convergence of event cost identification

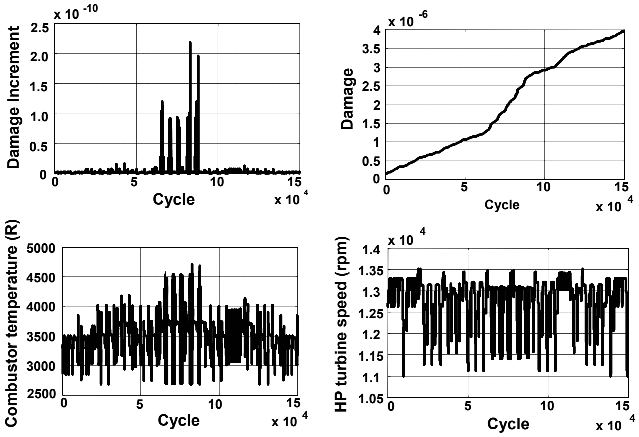
5.2 Optimal DES controller synthesis

The state transition cost matrix Π is determined from the event cost matrix $\tilde{\Pi}$, and the transition function δ of the finite state automaton. Given the state transition cost matrix Π and the state characteristic vector $\bar{\chi}$, the optimal DES controller can be synthesized. The characteristic values of the four marked states in Table I are assigned as: -0.05, -0.20, -1.00, and +0.20, respectively. These values are assigned based on the designer's perception of the importance of terminating on specific marked states. For example, the bad marked state *Unexpected engine halt* in Table I is assigned the characteristic value of -1.00 because the single engine aircraft will most likely be destroyed if the DFSA terminates on this state. On the other hand, the good marked state *Mission successful* is assigned the characteristic value of +0.20 based on its relative importance to the loss of the aircraft.

Table II lists the iterations of optimal control synthesis for the first 16 states. The performance measure of the unsupervised plant is negative at the states 3, 4, 5, 7, 8, 9, 11, 12, 13, 14, and 16 as indicated by bold script in Table II. All controllable events leading to these states are disabled and the resulting performance measure at Iteration 1 shows sign change at states 3, 4, 5, and 14 as indicated by italics in Table II. All controllable events leading to these states are now re-enabled for further increase in performance as seen in the column under Iteration 2, where sign change occurs only at state 8. Re-enabling all controllable events leads to state 8 increases the performance even further. The synthesis is complete in Iteration 3 (i.e., there is no need to go for the Iteration 4) because there is no sign change from Iteration 2 to Iteration 3. However, the Iteration 4 in Table II is shown to exhibit that there is no further improvement in the language measure.

Table II Optimal controller synthesis iterations

	Unsupervised plant	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
State 1	0.1392	0.2396	0.2654	0.2749	0.2749
State 2	0.1406	0.2420	0.2681	0.2777	0.2777
State 3	**-0.1826**	***0.0475***	0.0762	0.0819	0.0819
State 4	**-0.1011**	***0.0163***	0.0462	0.0619	0.0619
State 5	**-0.4348**	***0.0000***	0.0301	0.0346	0.0346
State 6	0.1576	0.2585	0.2833	0.2930	0.2930
State 7	**-0.3373**	**-0.0322**	**-0.0083**	**-0.0045**	**-0.0045**
State 8	**-0.1134**	**-0.0077**	***0.0126***	0.0268	0.0268
State 9	**-0.8250**	**-0.7116**	**-0.7061**	**-0.7050**	**-0.7050**
State 10	0.1249	0.2250	0.2493	0.2590	0.2590
State 11	**-0.3759**	**-0.1857**	**-0.1706**	**-0.1665**	**-0.1665**
State 12	**-0.1318**	**-0.0241**	**-0.0066**	**-0.0007**	**-0.0007**
State 13	**-0.8638**	**-0.8545**	**-0.8520**	**-0.8512**	**-0.8512**
State 14	**-0.0622**	***0.0372***	0.0628	0.0721	0.0721
State 15	0.3378	0.4372	0.4628	0.4721	0.4721
State 16	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**



1 Cycle ≡ 0.02 sec

Figure 8 Supervised plant output (fixed PLA input)

The performance of the optimal controller was compared with that of Controller 1 and Controller 2, which are designed using the conventional procedure [RW87] [WRPL03]. The optimal controller not only yields the best mission performance of all controllers and unsupervised plant, but also reduces the accumulated damage of the unsupervised plant. However, it may not necessarily yield less damage accumulation than all other controllers because damage criteria were not addressed in the formulation of the optimal control policy. Figure 8 shows the engine outputs for the input given in Figure 2 under the supervision of optimal controller whose damage accumulation is less than that of unsupervised plant, but slightly exceeds that of Controller 2. Table III compares damage accumulation for each case.

Table III Damage under Different Supervisors

Unsupervised	Controller1	Controller2	OptimalControl
5.51 x 10^{-6}	5.51 x 10^{-6}	3.65 x 10^{-6}	3.96 x 10^{-6}

Theoretical performance of the supervisors can be associated with the language measure of each supervisor. The language measure of the unsupervised plant and that of the three controllers are listed in Table IV.

Table IV Performance under Different Supervisors

$\mu_{\mathrm{Unsupervised}}$	$\mu_{\mathrm{controller1}}$	$\mu_{\mathrm{controller2}}$	$\mu_{\mathrm{OptimalControl}}$
0.1392	0.1312	0.2269	0.2749

The performance of the null and the three supervisors are compared based on observations of mission execution on the simulation test bed. For each controller, 100 missions were simulated, and the mission outcomes were recorded with respect to the characteristic values assigned to the four marked states. Assigning the characteristic values (χ): -0.2, 0.2, -1.0, -0.05 to states: *Mission Abortion off-Ground* (14), *Mission Success* (15), *Engine Halt* (16) and *Mission Abortion on-Ground* (42), respectively, simulated performance of the unsupervised plant and each of the three controllers is calculated as given below:

$$\nu_{\textbf{Unsupervised}} = 33*(-0.2)+59*0.2+4*(-0.05)+4*(-1.0)= 1.00$$

$$\nu_{\textbf{Controller1}} = 81*0.2+19*(-1.0) = -2.80$$

$$\nu_{\textbf{Controller2}} = 25*(-0.2)+68*0.2+3*(-0.05)+4*(-1.0)= 4.45$$

$$\nu_{\textbf{OptimalControl}} = 31*(-0.2)+63*0.2+6*(-0.05)+1*(-1.0)= 5.10$$

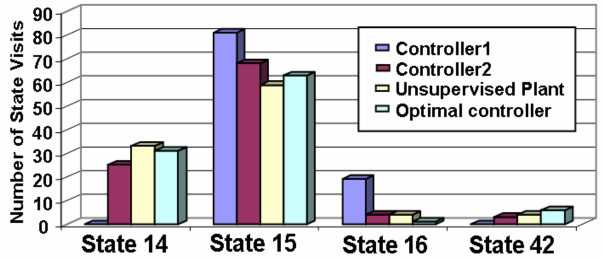


Figure 9 Simulated Performance of Controllers

The bar chart in Figure 9 shows a comparison of mission behavior for each supervisor under simulation experiments. It is seen that the theoretical performance of the supervisors, listed in Table IV, is in qualitative agreement with the experimental results.

6 Summary and Conclusions

This paper presents a quantitative approach to synthesis of an optimal discrete-event supervisory (DES) control of a complex engineering system based on the recent theoretical work in this field [WR04] [FRL04] [SR03] [WRK03].

The optimal DES control law has been validated on a gas turbine engine simulation test bed. The plant model in the simulation test bed is built upon the model of a generic turbofan 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 such as power plants or robots.

The results of simulation experiments have shown the DES supervisor is capable of simultaneously reducing structural damage and improving the mission behavior of the engine system. Real events were generated as the simulation executed the DES control policy, and control commands were issued by the supervisor based on the observed events. Simulation experiments on the test bed establish feasibility of the optimal DES control theory for applications to other large-scale engineering systems. To the best of the authors' knowledge, this is the first application of optimal DES control to a large-scale engineering system reported in open literature.

References

[FLR03] J. Fu, C. Lagoa and A. Ray, "Robust Optimal Control of Regular Languages," *42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 3209-3214.

[FRL03] J. Fu, A. Ray and C. Lagoa, "Optimal Control of Regular Languages with Event Disabling Cost," *American Control Conference*, Denver, Colorado, June 2003, pp.1691-1695.

[FRL04] J. Fu, A. Ray and C. Lagoa, "Unconstrained Optimal Control of Regular Languages," *Automatica*, Vol. 40, No.4, April 2004, pp. 639-646.

[RW87] P.J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control and Optimization,* Vol. 25, No. 1, January 1987, pp. 206-230.

[SR03] A. Surana and A. Ray, "Signed Real Measure of Regular Languages," 42nd *IEEE Conference on Decision and Control* (*CDC)*, Maui, Hawaii, December 2003, pp. 3233-3238.

[WRK03] X. Wang, A. Ray and A. Khatkhate, "On-line Identification of Language Measure Parameters for Discrete Event Supervisory Control," *IEEE Conference on Decision and Control* (*CDC)*, Maui, Hawaii, December 2003, pp.6307-6312.

[WRPL03] X. Wang, A. Ray, S. Phoha and J. Liu, "J-DES: A Graphical Interactive Package for Analysis and Synthesis of Discrete Event Systems," *American Control Conference*, Denver, Colorado, June 2003, pp. 3405-3410.

[WR04] X. Wang and A. Ray, "A Language Measure for Performance Evaluation of Discrete Event Supervisory Control Systems," *Applied Mathematical Modelling*, 2004, in press.