FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 7, July, 2025, pp 294 – 303
DOI: https://doi.org/10.33003/fjs-2025-0907-3618

EXPERT SYSTEM FOR DETECTING AND DIAGNOSING CAR ENGINE OVERHEATING FAULT USING DYNAMIC CONTROL SYSTEM (DCS)

***[1]David Ibitayo Lanlege, [2]Adeola Isaac Adeniyi, [3]Abdullahi A. Wachin and †[1]Olojede Abayomi Opeyemi**

[1]Department of Mathematics, Faculty of Science, Federal University Lokoja.
[2]Department of Statistics, Faculty of Science, Federal University Lokoja.
[3]Department of Mathematics, Faculty of Science, Airforce Institute of Technology Kaduna.

*Corresponding authors' email: david.lanlege@fulokoja.edu.ng Phone: +2348030528667
ORCID iD: *https://orcid.org/0009-0002-4135-4537 †https://orcid.org/0009-0009-3112-3613

ABSTRACT

Dynamic Control Systems (DCS) can be applied in detecting and diagnosing car engine overheating and can be continuously implemented to serve fields such as Applied Mathematics, among others. Car engine overheating is a sequence of diagnostic processes that deploys expertise. In addition, Advanced System is one of the leading Artificial Intelligence techniques that have been adopted to handle such tasks. This paper presents the imperatives for an advanced system in developing Dynamic Control Systems for detecting and diagnosing car engine overheating through input and output requirements of constructing successful differential equations. Thus, DCS provides input and output equations in the form of Matrix/Vector State Space Representation (MSSR) via mathematical Differential Equations in the form of the DCS approach.

Keywords: Car Engine Overheating, Generating Function, Input and Output equations, Differential Equations, Matrix/Vector State Space Representation (MSSR), Dynamic Control Systems (DCS), State Equation

INTRODUCTION

The meaning of an ES as proposed by (Al-Taani, 2007) is: "an intelligent computer program that uses information and induction systems to take care of issues that are sufficiently troublesome to require huge human ability for their answer". The ES is an information-based framework comprising two primary modules: the information base and the induction motor. It, as a rule, has an information procurement module and a clarification module as additional parts. Frameworks that use the information-based methodology are clearer than the ordinary methodology. Information is addressed expressly in the information base, so it tends to be modified a bit. This representation frequently appears as rules. The inference engine uses the information base items to tackle a specific issue as indicated by the client reactions through a connection point (for example, the symptoms of the car fault).

The inference module takes advantage of the knowledge to apply it to the given issue (Al-Taani, 2007). Expert systems (ES) provide a powerful and flexible means of solving various problems that often cannot be addressed by more traditional and conventional methods. The terms *expert system* and *knowledge-based system* (KBS) are often used interchangeably. The four fundamental components of a KBS are: a knowledge base, an inference engine, a knowledge acquisition tool, and a dedicated user interface. Some of the key applications of KBS include clinical diagnosis, engineering failure analysis, decision support, knowledge representation, weather forecasting, decision-making, learning, and chemical process control (Al-Taani 2007).

An advanced system is a group of component parts put together to accomplish a certain task. It is also said to be an arrangement or collection of things connected or related in such a manner as to form an entire whole. Simply, a system is an arrangement of physical components connected or related in such a manner as to form and or act as an entire unit. Whereas the concept of Control is analogous to either direct, regulate or command. Thus, a Control System is an arrangement of physical components connected or related in such a manner as to command, direct and regulate itself or another system. It is therefore important to note that a control system is made up of three components, namely: input, process and output (Ibitayo et al., 2015)

(Widodo et al., 2020). Exploring the effect of knowledge management and social intelligence on professional performance of an advanced system may completely fulfill a function that normally requires human expertise, or it may play the role of assistance to a human decision maker. The decision maker may be an expert in his own right, in which case the program may justify its existence by improving his productivity Knowledge acquisition is the transfer and transformation of potential problem-solving expertise from some knowledge source to a program. Knowledge representation is a substantial subfield which shares many concerns with both formal philosophy and cognitive psychology. It is concerned with how information might be stored and associated in the human brain, usually from a logical, rather than a biological, perspective. An advanced system can be distinguished from a more conventional application program. It stimulates human reasoning about a problem domain, rather than simulating the domain itself. It performs reasoning over representations of human knowledge, in addition to doing numerical calculations or data retrieval.

The knowledge base is structured as a set of objects, with rules relating them to corresponding attributes. An object is the conclusion that is defined by its associated rules, while an attribute is a specific quality that, with its rule, helps define the object. The rule that is applied to an attribute states that an object either has or does not have that attribute. These rules occur in sequences and are expressed in the form (Widodo et al., 2020): If < conditions >, then < actions >. If the conditions are true, then the actions are executed. When rules are examined by the inference engine, actions are executed if the information supplied by the user satisfies the conditions in the rules. Conditions are expressions involving attributes and logical connectives. The rule-based advanced systems have a wide range of applications for diagnostic tasks where expertise and experience are available, but a deep understanding of the physical properties of the system is either unavailable or too costly to obtain. In the rule-based systems, knowledge is represented in the form of production rules.

The Inference Engine

To execute a rule-based advanced system using the method of forward chaining, we merely need to fire (or execute) actions whenever they appear on the action list of a rule whose conditions are true (Durkin, 1990b). This involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied. A general algorithm for this might be:

While values for attributes remain to be input
Read the value and assign it to the attribute
Evaluate conditions
Fire rules whose conditions are satisfied.

Several points about this require consideration. First, a conflict resolution strategy needs to be employed to decide which rules are fired first. Our method is to fire the rule that the system designer defined first. Also, we wish to cut down on computational time. To do this, we must not do anything which does not absolutely need to be done. This means that conditions are only evaluated at the time they might change, and that rules are checked (to see if all of their conditions are satisfied) only when they might be ready to be fired, not before. We shall do this as attributes are assigned values and shall only consider rules and conditions affected by the new attribute assignment (Durkin, 1990a).

Working Memory

Specific information on a current problem is represented as case facts and entered into the advanced system's working memory. The working memory contains both the facts entered by the user from questions asked by the advanced system, and facts inferred by the system. The working memory could also acquire information from databases, spreadsheets, or sensors, and be used by the advance system to conclude additional information about the problem by using the general knowledge contained in the knowledge base (Durkin, 1990b).

Explanation of Facility

Other than giving eventual results or conclusions, human specialists and advanced systems can explain how they arrived at their results. This capacity is frequently important because the types of issues to which advanced systems are applied require support for the result to be given to the user. For instance, a specialist framework which suggests antibiotic treatment for a patient would have to clarify for the doctor how this proposal was figured out (Durkin, 1990). Advanced systems likewise can make sense of why a given inquiry is being posed. At the point when an individual talks with a human master, the discussion is exceptionally intuitive, and once in a while, the individual might inquire as to why a specific logic is being sought after. The clarification given can cause the client to feel more OK with the line of addressing and furthermore help to explain what the master accepts are significant for the issue (Durkin, 1990).

The User Interface

The user interface is the means of communication between a user and the advanced systems' problem-solving processes. A good advanced system should have an efficient interface. The user interface in this system will be able to accept the instructions in a form that the user enters and translate them into working instructions for the rest of the system. The user will be asked several questions, which he will have to answer; subsequently, a solution will be displayed. Careful attention should be given to the screen design to make the advanced system appear friendly to the user (Neeta et al., 2010).

Advanced System for Diagnosing the Failure of Different Machines

One of the earlier published references on diagnostic ES for technical fault diagnosis was developed at MIT in the early 1970s, as stated by Scherer and White in 1989 (Salama et al., 2012). Some research that utilises the advanced system to diagnose the failure of different machines is reviewed in the following paragraphs below: Kadarsah proposed and designed a decision model for car fault diagnosis in which an ES was utilised to help experienced mechanics and drivers (Salama et al., 2012). The model consists of an inference engine, a knowledge base, a database, user interaction and an adaptive mechanism. The Inference engine uses backwards chaining as a result of a small number of outputs with many possible inputs. In addition, the adaptive mechanism was utilised in the user interaction section to receive feedback about the system's diagnosis result. The feedback results were stored in a database.

The adaptive system then processes the stored data and extracts additional rules to improve the knowledge base. In this system, car faults are divided into three states: Start-up state, Run-stable state and Movement-state. A rule-based advanced system (CLIPS) with a forward chaining inference engine is used in the implementation. CLIPS stores the knowledge in rule form, which has a logic-based representation as well as production rules. The system interacts with the user through an interface and gives the diagnosis result with an illustration. The rule-based advanced system contains 150 rules for car failure causes. However, improvement in the domain knowledge and applying adaptive techniques for knowledge creation are required in such a system (Salama et al., 2012).

In the work of Peter Nabende and Tom Wanyama, Heavy diesel Engines (HDDEs) were proposed. HDDEs maintenance requires high technical skills and extensive experienced mechanics, which are scarce. As a result, employing an advanced system in such a domain can be highly useful. The HDDE faults diagnosis ES was able to successfully detect malfunctions in the engines and give recommendations for corrective actions. System development leads to a collection of valuable information related to HDDE fault diagnosis and training. However, updating the knowledge base affects the reasoning process performance, especially in the continuous run (Salama et al., 2012). Research was done by Jindal, Jain, Aggarwal and Verma to assist in the design of an ES for car failure diagnosis and repair (Salama et al., 2012). Many factors were considered in this research, such as the required time, the place and the human expertise level.

In addition, the ES development was accompanied by reviewing the technologies used in designing such systems to achieve the best means to be followed. However, the proposed prototype was not promoted to be used as a complete application due to time and resource limitations. Thus, adopting new rules to be performed was an example of further enhancements that the system needed. A survey was done by Milanović, Misita, Tadić, and Milanović for developing a motor cultivator fault diagnosis model. This model is based on the hybridisation of the Advanced system (ES) and the Decision Support System (DSS), in which ES outcome represents the input to the DSS. The supplier selection for faulty component replacement is made by DSS based on the ES outcome. In practice, the designed hybrid system was applied in a small motor cultivator importer and distributor company for servicing purposes. It has proved to be a very useful tool for equipment servicing needs with a low development cost. It increases the efficiency of labour and workers ‘satisfaction (Salama et al., 2012).

Conventional Programs versus Advanced Systems
According to Durkin (1990a), it is important to understand and appreciate the differences between conventional computer programs and knowledge processing or advanced systems. Knowledge processing represents an evolution, rather than a revolution, in the way individuals and computers interact to solve problems. The most basic difference between the two is that conventional programs process data, while advanced systems process knowledge. This basic difference influences both the nature of the processing technique used and the results obtained. Conventional programs process data, which is usually in numeric form, while an advanced system works with symbolic information. Data are isolated bits of information about a problem; whereas symbolic information represents statements or facts concerning the problem, which can be used with general knowledge to infer new information. Conventional programs process data using algorithms, whereas an advanced system will use heuristic reasoning techniques. An algorithm represents a finite set of well-defined steps to be performed. Heuristic reasoning works with the available information to draw conclusions about the problem, but does not follow a prescribed sequence of steps. A conventional program requires complete and precise information. An advanced system can work with the available information, whether it is incomplete or uncertain. In this sense, an advanced system can provide some results even under the constraints of limited or uncertain information. A conventional program would be severely limited under such constraints.
The interface of an advanced system permits questions to be asked and answers given using a natural language style. This interface is more readily accepted by end-users than the command interface found with most conventional programs. Interaction with an advanced system also follows more closely the conversation between one human obtaining advice from another human. During the conversation, explanations are provided by the expert to queries as to "why" a question is being asked, and "how" a given conclusion was reached. This point makes an advanced system considerably unlike a conventional program, which simply provides a final answer. Conventional programs provide a final solution, usually in the form of a result from a computation. The final solution offered by conventional programs is typically the computation's result. It doesn't matter if the computation required a complicated sequence of steps; the user will only see the outcome.
An explanation in the form of a line graph is provided by advanced systems along with a recommendation as the outcome. When given accurate data, traditional software will precisely solve an issue. "All or nothing" applies to this circumstance. Advanced systems can make mistakes, just as a human expert might. This point appears to give the conventional program an advantage over the advanced system. However, this appearance is only an illusion (Rasmussen, 1990). Advanced systems work on types of problems which are less structured than conventional programs, and the information available may not be sufficient to obtain an exact solution. However, the advanced system will still be able to reach some reasonable conclusion, even if it is not optimal, whereas a conventional program will fail if not provided with all of the information it needs.
This ability of an advanced system to be able to make decisions in the absence of complete or certain information is the result of developments in the area of inexact reasoning (Durkin, 1990a).

Why use an advanced system?
Like any project venture, developing an advanced system must have some justification (Durkin, 1990a). Insight for justifying an advanced system can be gained when one compares an advanced system with a human expert. One can formulate several general reasons for employing an advanced system, such as:

i. Replacement of a human expert.
ii. Assistant to a human expert.

Transfer of expertise to novice (Durkin, 1990a). Using an advanced system to replace a human expert is done primarily to use the system when the expert is not available. For example, through time constraints, the human expert may not be available, while an advanced system designed to control some manufacturing process would be available 24 hours a day. Another advanced system, containing the expertise of a unique expert within a company, could be made available to company sites located in other geographic areas. If the expert should leave or retire from the company, the expertise captured in the advanced system could serve as a replacement for the expert.
Human experts may be scarce, hence expensive. Advanced systems, by contrast, may be inexpensive. Developing an advanced system can be a costly venture, but the finished product will have low operating costs. The finished system can also be duplicated at low cost and distributed widely (Durkin,1990b). In the area of science, justifying an advanced system for replacing a human can be found in such applications as space exploration or providing the expertise of a geophysicist to some remote oil exploration site. Another example would be to replace the human operator of a control process. Assisting a human expert is one of the most commonly found applications of advanced systems.
In this application, the advanced system attempts to aid the human expert in a routine or mundane task. For example, a physician may have general knowledge of most diseases, but could use some additional support in diagnosing a given problem with a patient. In another example, a bank manager may be responsible for processing numerous loan applications, but could use help with some of the routine decisions made. In both applications, the human expert is fully capable of performing the task, but obtains additional support from the advanced system. In this type of advanced system application, the objective is to improve the overall productivity of the current practice. One specialised application of an advanced system which can be used to assist the expert is the ability of the advanced system to learn about a specific problem. The most common learning method used in advanced systems today is a technique known as induction (Durkin, 1990b).
The induction technique works with information contained in a set of examples to induce a set of rules which capture the knowledge about the problem. This approach has particular value for those problems where the expert lacks the knowledge to form decisions, but has a history of data on the problem. The induction technique can uncover classifications in the data, which can be used to guide the decision process. The expertise held by a human expert is a valuable resource. Knowledge is gained by the expert through years of experience from working on the problem. In many organisations, it is important that this expertise not be lost but

transferred to others through training. An advanced system can be developed to accomplish this training task.

Method of Solution

Dynamics Control System (DCS)

In applied mathematics and engineering, the central theory deals with the behaviour of dynamical systems over time. The dynamic behaviour of a system may therefore be understood by studying its mathematical description. For instance, the flight path of an airplane subject to certain engine thrust, rudder elevation angles and particular wind condition or the current flowing in an electrical circuit consisting of interconnections of resistors, inductors, capacitors, transistors, diodes, voltage or current source etc can be predicted using mathematical description of the pertinent behaviour. Mathematical equations in the form of Differential or difference equations are used to describe the behaviour of the process, usually referred to as governing equations, whose solutions give the required response of the particular system under consideration (Pantazi et al., 2008)

A system is a group of component parts put together to accomplish a specific task. It is also referred to as an arrangement or collection of things connected or related in such a way that they form a whole. The system is a simple system of physical components connected or related in such a way that they form and act as a whole unit (Verma, et al., 2010). The concept of control is analogous to direct, regulate or command. Thus, a control system is a set of physical components connected or related in such a way as to control, direct and regulate itself or another system. It should therefore be noted that the control system consists of three components: input, process and output.

Mathematical Classification of Systems

In this paper, we will not focus on a comprehensive classification of systems, as this may not provide the desired understanding of the concept. Therefore, the enumeration of the most common classes of systems most often encountered in the field of engineering and science is of great importance (Pantazi, 2009). Each particular set of equations describing a given system generally depends on the effect to be captured. Some of these systems may include lumped parameters or finite-dimensional systems, distributed parameters or infinite-dimensional systems, continuous-time and discrete-time systems, deterministic and stochastic systems, and an appropriate combination of any of the above-mentioned systems is known as a hybrid systems. It should be noted, however, that the appropriate mathematical setting for a finite-dimensional system is finite-dimensional vector spaces, and for an infinite-dimensional system, an infinite-dimensional linear space is defined.

Continuous-time finite-dimensional Systems are described by ordinary differential equations or integral equations, whereas discrete-time finite-dimensional systems are governed by ordinary difference equations or discrete-time counterparts to these Integral equations. The governing differential equations for infinite-dimensional Systems include partial differential equations, Volterra integro-differential equations, and functional equations.

Finite-Dimensional System

This is mainly concerned with continuous-time and Discrete-time finite-dimensional systems.

The continuous–time finite-dimensional dynamic system for our consideration will be those described by the following set of governing differential equations:

$\dot{x}_i = f_i(t, x_1, x_2, \ldots x_n, u_1, u_2, \ldots u_m), i = 1,2,3, \ldots n$

$\dot{y}_i = g_i(t, x_1, x_2, \ldots x_n, u_1, u_2, \ldots u_m), i = 1,2,3, \ldots p$

Where $u_i, i = 1(1)m$ denote the inputs or the stimuli; $y_i,\ i = 1(1)p$ denote outputs or responses; $x_i, i = 1(1)n$ represent state variables; t denotes time $\dot{x}_i$ denote time derivatives of state variables; $f_i, i = 1,2,3, \ldots n$, real value functions $1 + m + n$ real variables; $g_i, i = 1,2,3, \ldots p$, real value function of $1 + m + n$ real variables.

A complete description of the system will usually require a set of initial conditions.

$x_i(t_0) = x_i(0), i = 1,2,3, \ldots, n \ where\ t_0\ is\ initial\ time.$

In most cases of practical application, there often arises the need to impose constraints on the quantities. $f_i, g_i\ and\ u_i$

Open-Loop and Closed-Loop Control Systems

The Controlled system can be classified into two categories: open–loop and closed-loop. An open-loop controlled system is one in which the control action is independent of the output, whereas a closed-loop controlled system is one in which the control action is somehow dependent on the output. As a result, overheating of the car engine is categorised as a closed-loop control system because the respective inputs depend upon the respective outputs of the system (Lanlege et al., 2015). Feedback is also a characteristic of a closed-loop controlled system. This is defined as the property of a closed-loop control system which allow the output (or some other control variable) to be compared with the input to the system (or the output to other internal components or subsystems) in such a way that an appropriate action can be created as a function of the output and input. Some of the characteristics of feedback include the following:

i. Increases oscillation accuracy
ii. Tendency towards instability
iii. Reduces the sensitivity of the output ratio to system parameter variations
iv. Reduces the effect of nonlinearity
v. Reducing the effect of external disturbance or noise
vi. Increases bandwidth.

Open-Loop Control System

An open–loop controlled system is one in which the control action is independent of the output of the system. Manual control is also an open-loop control system. Fig. 1 shows the block diagram of an open-loop control system in which the process output is completely independent of the controller action (Ibitayo et al., 2016).

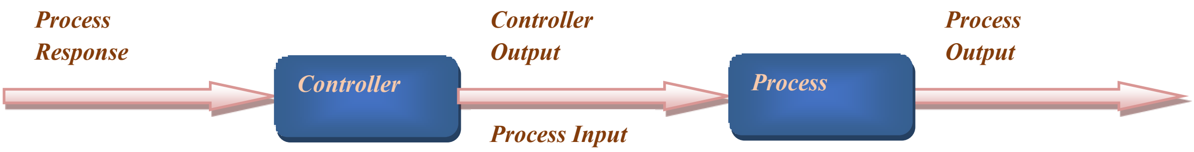

Figure 1: Open-Loop Control System

Feedback Loop of Control System

When designing a control system, feedback is a common and powerful tool. The feedback loop is a tool that considers the system output and enables the system to adjust its performance to achieve the desired system result.

In any control system, the output is affected by changes in environmental conditions or disturbances. Therefore, one signal is taken from the output and transmitted to the input. This signal is then compared with the reference input, and an error signal is generated. The error signal is transmitted to the controller, and the output is corrected. This system is called a feedback system. A block diagram of the feedback system is shown below.

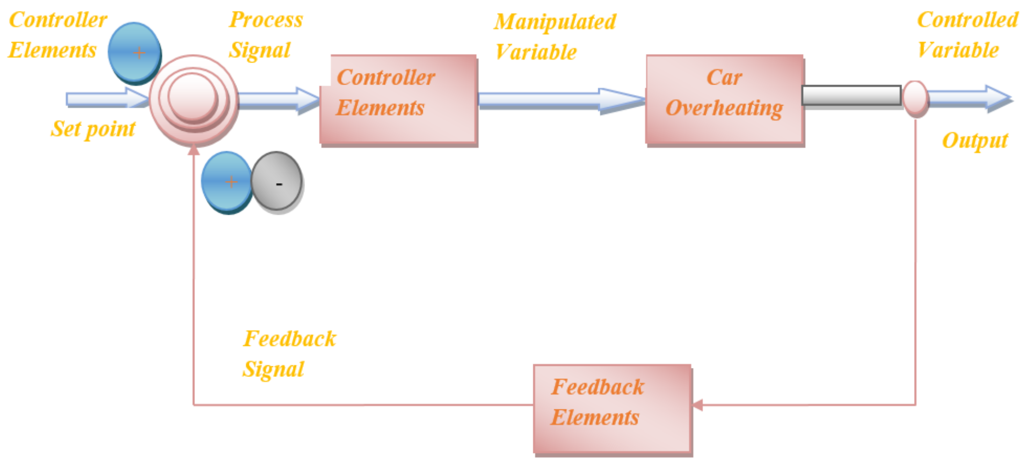

Figure 2: Feedback Loop of Control System

For a given set of differential equations describing a certain dynamic system (a system that changes with time), there is a need to know the set of state variables and their number in the system. That is;

i. How many state variables are involved?
ii. What are these state variables?

For figure (i) above, the number of state variables is equal to the total number of initial conditions required to completely solve the differential equations of overheating for a car engine. For instance, if a dynamic system is described by a single second-order differential equation, then two initial conditions are required to completely solve the differential equation. Thus, there are two state variables for this system. For figure (ii), these variables for which initial conditions are required for the solution of the governing differential equation defined above are chosen as the required state variables.

General Formulation

Once the state variables are appropriately selected the next step is to construct the state variable equations. These state equations are system of first-order differential equations in the state variables on the left hand side and algebraic system (function) of the state variables as system input and possibly time on the right hand side. (Lee & Salapaka, 2009).

In general, for a multi-input multi-output system with $m$state variables, we have:

$x_1, x_2, x_3, \dots x_m, p\ inputs\ u_1, u_2, u_3, \dots u_p$, and r outputs $y_1, y_2, y_3, \dots y_r$, the state variable equations re given in this form:

$$\left.\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \\ \dot{x}_2 &= f_2(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \\ \dot{x}_3 &= f_3(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \\ &\vdots \\ \dot{x}_m &= f_m(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \end{aligned}\right\} \qquad (1)$$

where f_j are in general non-linear functions of the arguments

Similarly, the system output variables may also be expressed as follows:

$$\left.\begin{aligned} \dot{y}_i &= g_1(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \\ \dot{y}_2 &= g_2(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \\ \dot{y}_3 &= g_3(x_1, x_2, \dots x_m, u_1, u_2, \dots u_p, t) \end{aligned}\right\} \qquad (2)$$

$where\ \ g_k's,$ are in general non linear functions.

In the event that non-linear elements are present in the system the functions $f_j(j = 1(1)m)$ and $g_k(k = 1(1)m)$ also turn out to be non-linear and quite complex in nature thereby making the analysis or solution complicated.

Matrix Vector Formalism

This involves the representation of equations (1 and 2) more conveniently using matrix-vector form by the following definitions:

$$\left.\begin{aligned} &X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \text{ is state vector, } \quad F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \text{ as the function, } \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \text{ output vector;} \\ &G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} \text{ a lg e braic Function and } U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \text{ input vector} \end{aligned}\right\} \qquad (3)$$

Thus; the state variable equations are represented by

$$\dot{X} = F(x, u, t) \qquad (4)$$

And the system output as

$$\dot{Y} = G(x, u, t) \qquad (5)$$

As expected, the complexity associated with the general formulation reduces considerably for the case of a linear system. If all the elements in the model of a dynamical system are linear, the algebraic functions appearing in equations (1) and (2) will take the following special forms:

$$\left.\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{13}x_3 + b_{11}u_1 \cdots + b_{1r}u_r \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{23}x_3 + b_{21}u_1 \cdots + b_{2r}u_r \\ \dot{x}_3 &= a_{31}x_1 + a_{32}x_2 + \cdots + a_{33}x_3 + b_{31}u_1 \cdots + b_{3r}u_r \\ &\vdots \\ \dot{x}_{\backslash m-1} &= a_{m-1,1}x_1 + a_{m-2,2}x_2 + \cdots + a_{m-3,m}x_m + b_{m-1,1}u_1 \cdots + b_{m-1,p}u_p \\ \dot{x}_{\backslash m} &= a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mm}x_m + b_{m1}u_1 \cdots + b_{mp}u_p \end{aligned}\right\} \qquad (6)$$

$$\left.\begin{array}{l} y_1 = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1m}\ \ x_m + d_{11}u_1 + d_{12}u_2 + \cdots + d_{1p}u_p \\ y_2 = c_{21}x_1 + c_{22}x_2 + \cdots + c_{2m}\ \ x_m + d_{21}u_1 + d_{22}u_2 + \cdots + d_{2p}u_p \\ y_1 = c_{31}x_1 + c_{32}x_2 + \cdots + c_{3m}\ \ x_m + d_{31}u_1 + d_{32}u_2 + \cdots + d_{3p}u_p \\ \vdots \\ y_{r-1} = c_{r-1,1}x_1 + c_{r-2,2}x_2 + \cdots + c_{r-1,m}x_m + d_{r-1,1}u_1 \cdots + d_{r-1,2}u_2 + \cdots d_{r-1,p}u_p \\ y_r = c_{r1}x_1 + c_{r2}x_2 + \cdots + c_{3m}x_m + d_{r1}u_1 \cdots + d_{r2}u_2 + \cdots d_{rp}u_p \end{array}\right\} \qquad (7)$$

By defining the following quantities to enable us to represent the formulations above in matrix-vector form:

$$\left.A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \\ a_{m1} & a_{m2} \cdots & & a_{mm} \end{bmatrix}_{m\times m} = State\ matrix,\ B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \vdots & \\ b_{m1} & b_{m2} \cdots & & b_{mp} \end{bmatrix}_{m\times p} = Input\ matrix\right\} \qquad (8)$$

$$\left.C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \vdots & \\ c_{r1} & c_{r2} \cdots & & c_{rm} \end{bmatrix}_{r\times m} = Output\ Matrix,\ D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \vdots & \\ d_{r1} & d_{r2} \cdots & & d_{rp} \end{bmatrix}_{r\times p}\right\} \qquad (9)$$

Therefore, the final form of the system in the matrix – vector form is given as:

$$\dot{X} = AX + BU \quad State\ Equation \qquad (10)$$

$$\dot{Y} = CX + DU \quad Output\ Equation \qquad (11)$$

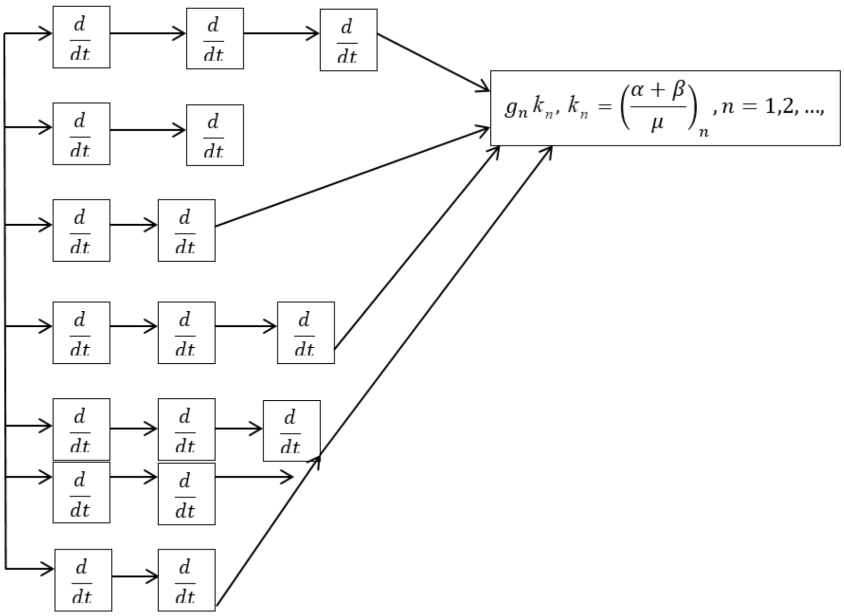

Figure 3: System Representation Using Block Diagram

Where α and β represent responses of sequences of decisions to be responded as either (Yes/No), respectively, while μ represents the total chances of decisive events that requires an absolute response at a time, is the input generating function, and is the equiprobable response generating function.

Governing Differential Equations for Diagnosing Overheating and Coolant Leaks

$$\frac{1}{k_1}(\dddot{p} + \ddot{p} + \dot{p} + p) = g_1 \qquad (12)$$

$$\frac{1}{k_2}[\ddot{q} + \dot{q} + q] = g_2 \qquad (13)$$

$$\frac{1}{k_3}[\ddot{r} + \dot{r} + r] = g_3 \qquad (14)$$

$$\frac{1}{k_4}[\dddot{s} + \ddot{s} + \dot{s} + s] = g_4 \qquad (15)$$

$$\frac{1}{k_5}[\dddot{t} + \ddot{t} + \dot{t} + t] = g_5 \qquad (16)$$

$$\frac{1}{k_6}[\ddot{u} + \dot{u} + u] = g_6 \qquad (17)$$

$$\frac{1}{k_7}[\ddot{v} + \dot{v} + v] = g_7 \qquad (18)$$

By equation (12) and (13) we thus have the State Variable Equations (SVE) as:

$$\left.\begin{array}{l} i_1 = p, \quad i_2 = q, i_3 = \dot{p}, i_4 = \dot{q}, i_4 = \ddot{q}, i_5 = \ddot{p}, i_5 = \dddot{p} \\ i_1 = i_3 \\ i_2 = i_4 \\ i_3 = i_5 \\ i_4 = -i_2 - i_4 + k_2 g_2 \\ i_5 = -1_1 - 1_3 - 1_5 + k_1 g_1 \end{array}\right\} \qquad (19)$$

Transforming equation (19) into a matrix vector form gives the matrix state space representation (MSSR) of the state variable equation (SVE); hence, the State equation (SE) is deduced as follow in equation. (20):

i.e.

$$\left.\begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ k_2 \\ k_1 \end{pmatrix} g(t)\right\} \qquad (20)$$

However, the State Equation (SE) for diagnosing overheating and coolant leak is given by Equation (21) Equation (10) for inputs p and q taken as I below:

$\dot{I} = AI + Bg$ (21)

Again, equation (14) and (15) are used to formulate another SVE as well as MSSR for the given system for the described system to be definitively diagnosed. Equation (22) below gives the SVE for the described system as:

$$\left.\begin{aligned} & s_1 = s, s_2 = r, s_3 = \dot{s}, s_4 = \dot{r}, s_5 = \ddot{s}, and \quad \dot{s}_4 = \ddot{r} \\ & \dot{s}_1 = s_3 \\ & \dot{s}_2 = s_4 \\ & \dot{s}_3 = s_5 \\ & \dot{s}_4 = -s_2 - s_4 + k_3 g_3 \\ & \dot{s}_5 = -s_1 - s_3 - s_5 + k_4 g_4 \end{aligned}\right\} \tag{22}$$

where

Converting equation (22) into a matrix vector form yield the matrix state space representation (MSSR) of the state variable equation (SVE) and hence the state equation (SE) is deduced as follow in equation (24):

$$\left.\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \\ \dot{s}_4 \\ \dot{s}_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ k_3 \\ k_4 \end{pmatrix} g(t) \right\} \tag{23}$$

The State Equation (SE) for the process diagnosis is given by equation (24) through equation (10) for the input S as follow:

$\dot{S} = AS + Bg$ (24)

Using equations (16) and (17), the State Variable Equations (SVE) are obtained as follow:

$$\left.\begin{aligned} & u_1 = T, \quad u_2 = U, \quad u_3 = \dot{T}, \quad u_4 = \dot{u} \quad u_5 = \ddot{T}, \quad u_5 = \dddot{T}, \quad \dot{u}_4 = \dddot{T} \\ & u_2 = T, \quad u_3 = \dot{u}, \quad u_4 = \dot{T}, \quad u_5 = \ddot{T} \\ & \dot{u}_1 = u_3 \\ & \dot{u}_2 = u_4 \\ & \dot{u}_3 = u_5 \\ & u_4 = -u_2 - u_4 - k_6 g_6 \\ & u_5 = -u_1 - u_3 - u_5 + k_5 g_5 \end{aligned}\right\} \tag{25}$$

Expressing equation (25) in a matrix vector form gives the matrix state space representation (MSSR) of the state variable equation (SVE) and hence, state equation (SE) is deduced from equation (26) through equation (10) for input u as:

$$\left.\begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ k_6 \\ k_5 \end{pmatrix} g(t) \right\} \tag{26}$$

Therefore, the State Equation (SE) representing the process diagnosis is defined from equation (27) to equation (10) with input u as:

$\dot{U} = AU + Bg$ (27)

Equation (18) consequently provides the State Variable Equation (SVE) as follows:

$$\left.\begin{aligned} & v_1 = v, \quad v_2 = \dot{v}, \quad \dot{v}_2 = \ddot{v} \\ & \dot{v}_1 = v_2 \\ & \dot{v}_2 = -v_1 - v_2 + k_7 g_7 \end{aligned}\right\} \tag{28}$$

Expressing equation (28) in matrix-vector form yields the Matrix State Space Representation (MSSR) of the State Variable Equation SVE). Consequently, the MSSR is derived as shown in equation (29) through equation (10) for input v as follows:

$$\left.\begin{pmatrix} \dot{v}_1 \\ \dot{v}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 8 \end{pmatrix} g \right\} \tag{29}$$

Finally, the State Equation (SE) for the process diagnosis is defined by equation (30) through equation (10) for input v.

$\dot{V} = AV + Bg$ (30)

Practical Illustration of an Advance system using Dynamic Control System

Output Equations

From equation (12), let's assume the input is g_1(Steaming or Leak?) i.e.:

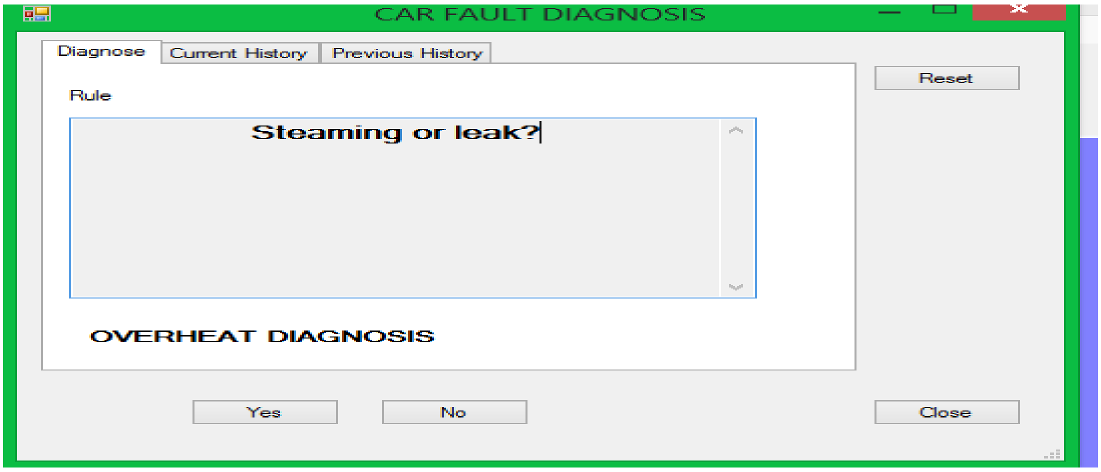

Figure 4: Steaming/Leak Verification Interface

If the response is yes, the resulting output equation is as follows:

$$y_1 = \alpha \frac{(AI+BG)}{\alpha+\beta}; \text{ ie:} A = [1 \quad 0 \quad 0 \quad 0 \quad 0], I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix}, B = [0 \quad 0 \quad 0 \quad 0 \quad 0] \; G = \left.\begin{pmatrix} 0 \\ 0 \\ 0 \\ k_2 \\ k_1 \end{pmatrix}\right\} \tag{31}$$

The graphical output of the equivalent Advance system is presented as follows:

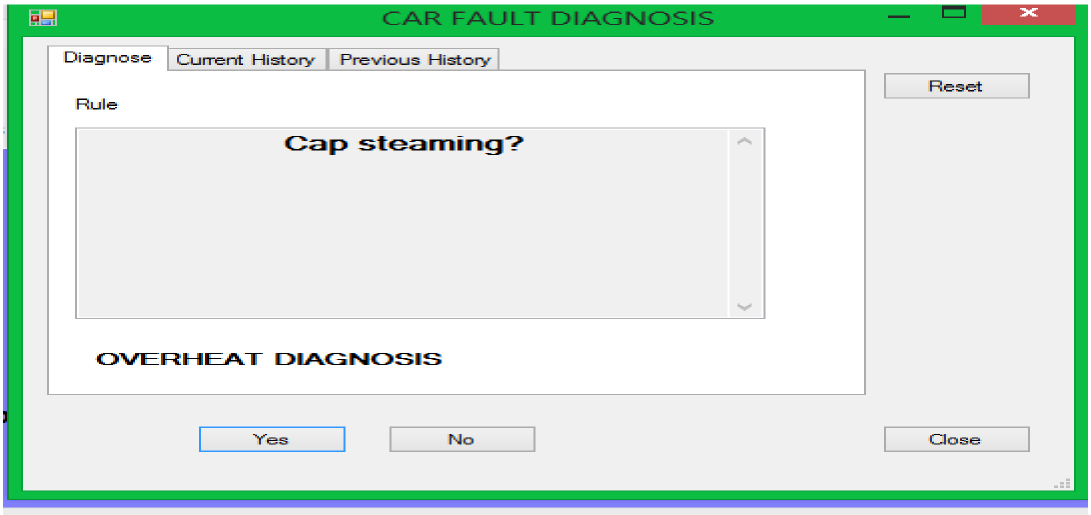

Figure 5: Interface Trying to verify Cap Steaming Faults

Similarly, if the input (implied response) such as "Cap Steaming?" is answered with Yes (Input Response), then the governing output equation is determined as:

$$\left. y_2 = \alpha \frac{(AI+BG)}{\alpha+\beta};\ \text{ie:} A = [0 \quad 1 \quad 0 \quad 0 \quad 0],\ I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix},\ B = [0 \quad 0 \quad 0 \quad 0 \quad 0] \text{ ie } y_2 = I_2,\ G = \begin{pmatrix} 0 \\ 0 \\ 0 \\ k_2 \\ k_1 \end{pmatrix} \right\} \tag{32}$$

The graphical output of the equivalent Advance system is presented as follows:

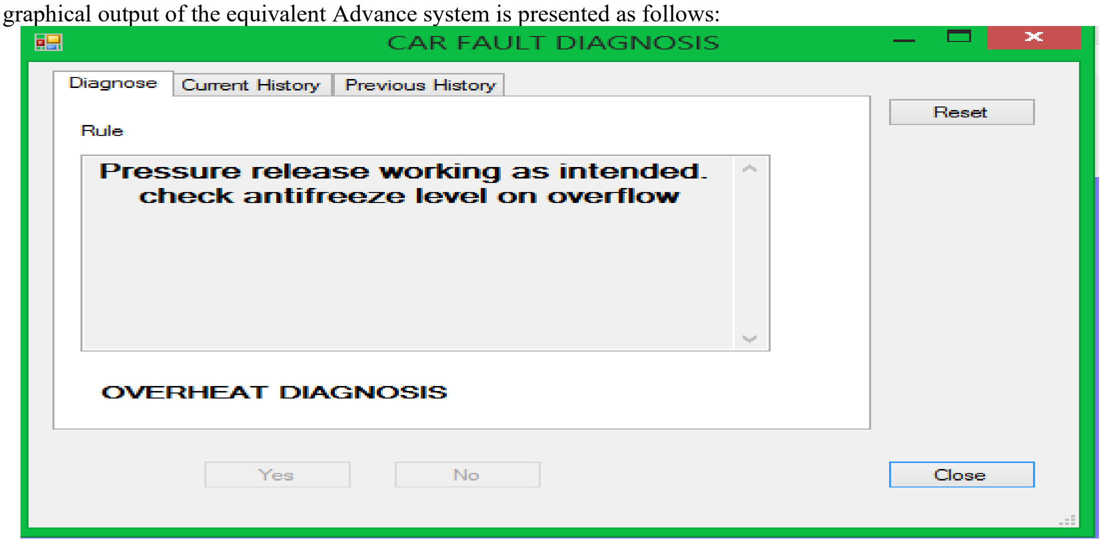

Figure 6: Interface Deciding to 'Check Antifreeze Level on Overflow'

Thus, the suggested output gives a general solution for the sequence of responses (Yes). In other words, the detected problem is checking antifreeze level on overflow.

Conversely, if the response is NO for the input (i.e., Steaming or Leakage?) in fig.5, the output equation is thus formulated by equation (12) as follows:

$$\left. y_2 = \beta \frac{(AI+BG)}{\alpha+\beta};\ \text{ie:} A = [0 \quad 1 \quad 0 \quad 0 \quad 0],\ I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix},\ B = [0 \quad 0 \quad 0 \quad 0 \quad 0]\ G = \begin{pmatrix} 0 \\ 0 \\ 0 \\ k_2 \\ k_1 \end{pmatrix} \right\} \tag{33}$$

The equivalent interface of the Advance system for this output is as follows:

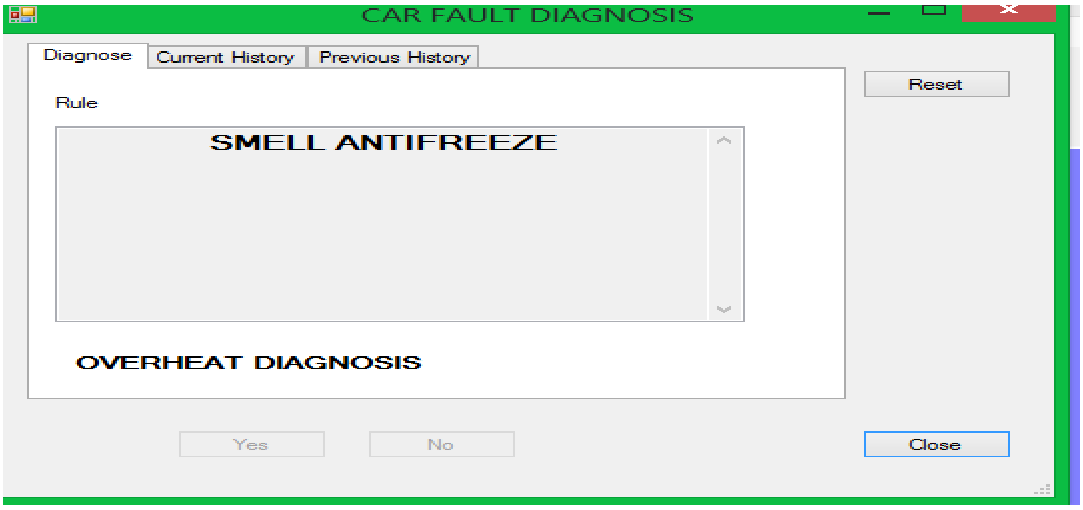

Figure 7: An Interface Displaying 'Smell Antifreeze' as the Detected Problem

Hence, the suggested output provides the general solution to the problem for the sequence of responses (No). This indicates that the detected issue suggests the presence of antifreeze smell, as illustrated in Fig. 7.

Signal Flow Diagrams

In this section, as appropriate signal flow diagrams of equations modeled in this paper will be illustrated.

The signal flow diagram for equations (18), (21) and (24) is given below:

Using equation (18) the signal flow is given in figure. 8:

Figure 8: Signal flow diagram for equation 18

Similarly, Figure 9: Illustrates the Signal Flow Diagram for Equation (21):

Figure 9: Signal flow diagram for equation 21

Figure 10: Once more, the signal flow diagram for equation (24) is deduced in the fig. 10 below:

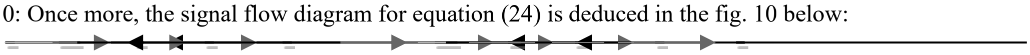

Figure 10: Signal flow diagram for equation 24

RESULTS AND DISCUSSION

Result

To overcome these deficiencies, the DCS provides a control method for which a decision made by an ES or any other decisive system can be represented using a provided referenced signal in a dynamical system, including: controlling a controlled element of the dynamical system using a control function; outputting, by the control function, a control signal depending on a reference signal indicating a desired setting of the controlled element and a feedback signal indicating an actual state of the controlled element; and modifying an internal state of the control function at one or more discrete time instants while applying the control function at periods not including the discrete time instants, wherein the modification of the internal state depends on a signal external to the control function (Pantazi, 2007).

The following equation gives a general stage process of decision making for detecting car engine overheating by an ES using a modelled DCS output general equation:

$$\left.\begin{array}{l} y = AI + BG, \forall \quad \text{given input/output say} \\ g_n \; y_n = (\alpha\beta)_n \frac{(AI+BG)}{\mu_n}; \\ ie: C = [n \times n] identity \ matrix \end{array}\right\} \quad (34)$$

$$\left.\begin{array}{l} E_n = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \\ \vdots \\ E_n \end{bmatrix}, B = [0 \quad 0 \quad 0 \quad 0 \quad 0] \text{ ie: } y_n = K_n \\ = E_n \forall n = 1,2,3, \ldots G = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_1 \\ k_4 \end{pmatrix} \end{array}\right\} \quad (35)$$

Discussion

Evidently, DCS and ES have typically been observed and demonstrated. It is also important to note that DCS is a sufficient mathematical model that uses mathematical derivatives to give a representation real-life decisive system, such as the ES, in the best form, as seen in the cases where SVE, MSSR, and SE, as well as the block representation of DDC, were used in representing and describing the decisive processes of an ES.

In addition, any decision made within the domain (g_n) surely has a corresponding co-domain as responses (y_n)at any given point in time. However, equations (31-33) and figures 1-5 have provided a real representation through which the main subject is said to have succeeded. That is to say, there exists a resulting input/output for both ES and DCS for any given car engine overheating detection diagnosis by an ES. In the case of DCS, equations (34 and 35) represent the input/output responses for any diagnosis of car engine overheating as corresponding to the ES alternative resolutions of any given input/output.

CONCLUSION

DCS has been used to describe the dynamic state of the decisions made by an ES. A Series of input and output equations is represented as the equivalence input/output of an ES. It is should be noted that the DCS can be used to diagnosing/detection of car engine overheating in real-life DCS applications as a finite dimensional system.

REFERENCES

Ahmad, T. (2006). Expert systems: Principles and programming. *Scalable Computing: Practice and Experience*, *7*(4).

Al-Taani, A. T. (2005). An expert system for car failure diagnosis. IEC (Prague), 5, 457–560.

Durkin, J. (1990b). Expert systems: Design and development. Macmillan Publishing Company.

Durkin, J. (1990a). Research review: Application of expert systems in the sciences.

Ibitayo, L., Mohammed, G., Rabiu, I., & Abdulrahman, A. (2016). Expert system for detecting and diagnosing car engine brake failure fault using dynamic control system (DCS). Applied & Computational Mathematics, 5 (1), 2–9.

Ibitayo, O. M., Ayeni, A. A., & Ayinde, A. B. (2015). Control systems engineering: Design, analysis, and applications.

Lanlege, D. I., Garba, U. M., Gana, U. M., & Abdulganiyu, A. (2015). Expert system for detecting and diagnosing car engine starter cranks fault using dynamic control system. Leonardo Journal of Sciences, (27), 41-62

Lee, C., & Salapaka, S. M. (2009). Fast robust nanopositioning A linear-matrix-inequalities-based optimal control approach. IEEE/ASME Transactions on Mechatronics, 14 (4), 414–422.

Neeta, A., Radhika, M., & Sushma, S. (2010). Human-computer interaction in expert systems.

Pantazi, A. (2009, December 10–11). Track-follow control for tape storage. Workshop on Dynamics and Control of Micro and Nanoscale Systems, IBM Research - Zurich.

Pantazi, A., Sebastian, A., Antonakopoulos, T. A., Bächtold, P., Bonaccio, A. R., Bonan, J & Eleftheriou, E. (2008). Probe-based ultrahigh-density storage technology. IBM Journal of Research and Development, 52 (4.5), 493–511.

Pantazi, A., Sebastian, A., Cherubini, G., Lantz, M., Pozidis, H., Rothuizen, H., & Eleftheriou, E. (2007). Control of MEMS-based scanning-probe data-storage devices. IEEE Transactions on Control Systems Technology, 15 (5), 824–841

Rasmussen, A. N., Muratore, J. F., & Heindel, T. A. (1990). The INCO advance system project: CLIPS in shuttle mission control. First CLIPS Conference, 305, 1157.

Salama, A. A., Kadarsah, S., & Nabende, P. (2012). Diagnosis of machine failure using expert systems. (Verma, et al., 2010)

Verma, J., & Aggarwal, J. (2010). An approach towards designing of car troubleshooting advance system. International Journal of Computer Applications, 1 (23), 107–114.

Widodo, W., Suendarti, M., & Hasbullah, H. (2020). Exploring the effect of knowledge management and social intelligence on professional performance of mathematics teachers: a mediating by achievement motivation. *Journal of Xidian University*, *14*(6), 749-757.