# Event Tracking for Real-Time Unaware Sensitivity Analysis (EventTracker)

Siamak Tavakoli, *Member, IEEE*, Alireza Mousavi, *Member, IEEE*, Peter Broomhead
Brunel University, United Kingdom

***Abstract*—This paper introduces a platform for online Sensitivity Analysis (SA) that is applicable in large scale real-time data acquisition (DAQ) systems. Here we use the term real-time in the context of a system that has to respond to externally generated input stimuli within a finite and specified period. Complex industrial systems such as manufacturing, healthcare, transport, and finance require high quality information on which to base timely responses to events occurring in their volatile environments. The motivation for the proposed EventTracker platform is the assumption that modern industrial systems are able to capture data in real-time and have the necessary technological flexibility to adjust to changing system requirements. The flexibility to adapt can only be assured if data is succinctly interpreted and translated into corrective actions in a timely manner. An important factor that facilitates data interpretation and information modelling is an appreciation of the affect system inputs have on each output at the time of occurrence. Many existing sensitivity analysis methods appear to hamper efficient and timely analysis due to a reliance on historical data, or sluggishness in providing a timely solution that would be of use in real-time applications. This inefficiency is further compounded by computational limitations and the complexity of some existing models.**

**In dealing with real-time event driven systems, the underpinning logic of the proposed method is based on the assumption that in the vast majority of cases changes in input variables will trigger events. Every single or combination of events could subsequently result in a change to the system state. The proposed event tracking sensitivity analysis method describes variables and the system state as a collection of events. The higher the numeric occurrence of an input variable at the trigger level during an event monitoring interval, the greater is its impact on the final analysis of the system state.**

**Experiments were designed to compare the proposed event tracking sensitivity analysis method with a comparable method (that of Entropy). An improvement of 10% in computational efficiency without loss in accuracy was observed. The comparison also showed that the time taken to perform the sensitivity analysis was 0.5% of that required when using the comparable Entropy based method.**

***Index Terms*—Discrete Event Systems, Event Tracking, Real-Time Systems, Sensitivity, Supervisory Control and Data Acquisition**

## I. Introduction

DATA acquisition systems that deal with large quantities of input variables and have higher sampling frequencies result in high bandwidth communication and place a heavy computational load on the higher tier data processing and information systems within their hierarchy. The focus of researchers and practitioners in this area has been to minimize this computational overhead by eliminating input variables that have the least impact on the system, this so called sensitivity analysis is discussed in [1]-[6]. Sensitivity analysis techniques help system analysts to focus on the most valuable information, information that most significantly impacts on system behaviour. Sensitivity indexing is a systematic approach for expressing the impact that any input variable has on the output parameters in a system. From the same perspective, sensitivity analysis is a systematic approach for expressing relationships between inputs and outputs of a system. Determining the true impact an input has on the output of a system is a real challenge due to the epistemic uncertainty that exists in the relationship between the respective variables. Selection of an appropriate method for sensitivity analysis depends on a number of factors and assumptions made with respect to this relationship. These factors are:

### *A. The Analytical Relationship between the Input and the Output Data*

The majority of Sensitivity Analysis (SA) methods attempt to determine the impact of changes in one variable in relation to others by means of analytical models that describes the relationship that exists between them. Methods such as Differential Analysis, Coupled/Decoupled Direct and Green's Function are classified among the analytical SA methods described in [7]. However, the non-linear and non-monotonic relationship between inputs and outputs of a given system may not necessarily lend themselves to the use of such analytical methods [8]. Situations pertain where the existence of a direct physical model in terms of mathematical equations does not exist between the respective system variables. In such cases a number of SA methods make use of statistical techniques in an attempt to extract relevant relationship features from the distribution of data series relating to input-output variables. For example Fourier Amplitude Sensitivity Test (FAST) [9], Morris [10], [18], Monte-Carlo [19] and Latin Hypercube [7] fall into this category of SA methods. The shortcomings of these methods lie in their reliance on historical data and the generation of data samples which the system analyst then fits to known probabilistic equations. One method that is less reliant on analytical methods for extracting sensitivity indices is the Entropy method [8]. In this respect the Entropy method is the most comparable and closest technique to the 'EventTracker' method and will be used to establish the sensitivity, accuracy and efficiency of the proposed method.

### B. *The Statistical Distribution of Input Variables*

The sensitivity indices of a system are normally influenced by the distribution of the input data series. For example, nonlinear relationships between input and output series in a model cannot be recognized by correlation-based sensitivity analysis methods alone [20]. Variance-based and Entropy-based indices are expected to be more sensitive to heteroscedastic data [8], whilst the homoscedasticity of data series can be higher in discrete signals and much higher between binary signals.

### C. *The Computational Overhead*

Sensitivity analysis is a computationally hungry process. In domain-wide sensitivity analysis methods, large batches of input variables are captured in a specified periodic time interval and subsequently values of sensitivity are determined using historical data analysis. For example, sampling-based methods need to generate new and equivalent sized batches of sample values for both output and input data regardless of the original sampling rates. The magnitude of resources required by such algorithms and their associated data processing requirements are comparable to the expected savings resulting from their application. For example, in correlation-based methods [20] there is a dependency on equally sized data batches for both input and output series of the model. In such cases the sampled data series either needs interpolation or extrapolation to maintain equal batch sizes, this in itself places additional computational load on the system.

In order to overcome some of the shortcomings found in existing methods, we introduce an effective and efficient way to perform sensitivity analysis of data in two time series. In the following sections, after a brief introduction to existing SA methods, a detailed description of event-driven data types and their impact on sensitivity analysis is provided. The proposed EventTracker method and its application in a case study are discussed. The advantages and application of EventTracker in an industrial case study is presented in the penultimate section. Conclusions are then drawn in the final section of the paper.

## II. Related work

### A. *Differential Analysis*

In differential analysis the impact of an independent variable on the dependent variable is assessed by identification of the perturbation behaviour of the dependent variable due to the changes in the independent variable [21]. This is achieved by finding the coefficients of the differential equation that governs the relationship between the independent and dependent variables [21]. Methods like Neumann expansion [22] and perturbation method [23] can help when extracting these coefficients through the approximation of differential equation. However, it can never be guaranteed that the often complex and nonlinear relationship that exists between system variables can be approximated with a sufficiently low error margin using differential equations alone [7].

### B. *Green’s Function Method*

Due to their nonlinearity, the task of differentiating model equations is in itself a difficult process. Green’s function can act as a catalyst in helping achieve the sensitivity equations [7]. In this method, the task of performing differentiation is effectively replaced by the sequence of finding the impulse response of the model [24], and then implementing the subsequent integration operations.

The concept of Green’s function stems from the fact that the total output of a linear time-invariant system can be formulated by a summation of terms that adds all outputs of the system for all single points [25]. In other words, each continuous function could be replaced by an infinite sum of delta functions whose distances approach zero. It is important to note that only a linear and time-invariant system can benefit from this approach. One further constraint in the application of Green’s function is that it only works with ordinary differential equations, equations that govern dependent variables with respect to independent variables. Often in real applications it is difficult to separate the relationship of independent and dependent variables. Additionally, working with one variable at a time for high dimensional systems could be computationally expensive and cumbersome.

### C. *Coupled/Decoupled Direct Method*

In the coupled direct method, after differentiation of model equations, the subsequent sensitivity equations are solved together with the original model equations. In the decoupled direct method they are solved separately [7]. This gives the impression that decoupled direct method is advantageous in terms of computational cost. Although the decoupled direct method is reported to be more efficient than Green’s function method [7]. In common with other analytical methods, prior knowledge of the model equations is a requirement. The couple/decoupled methods’ also exhibit the feature of being model-oriented and expert-hungry, these features makes them less attractive for practical applications when compared to SA methods that do not require model equations.

### D. *Monte Carlo and Latin Hypercube Methods*

Random data sample generation is the main characteristic of the Monte Carlo method. It provides the required values of independent variables from which dependent variables are produced [26]. The random sampling scheme occurs in no particular order, nor is it based on any criteria that would help with the efficiency of computation [7]. For example, in the Latin Hypercube Sampling (LHS) method [17], the range of each input parameter is divided into intervals of equal probability. Within sets of input parameter samples, each input parameter takes a random value from one of its intervals with the proviso that there is no repeat of that interval for a full sampling cycle [17]. In this way, there is a better chance and greater probability that all segments of data will be considered within the distribution; and that in doing so a more informative distribution of output parameters will be generated in a shorter period [7].

In taking a general overview of the Monte Carlo method, as depicted in Figure 1, from the stream of available data, the probability distribution of input variables is first estimated (i.e. the curve fitting blocks). Then based on these distributions, random sample generation occurs, (i.e. sampler blocks). After the model is applied to the generated samples, the output values are processed for estimation and extraction of their distribution attributes [27].

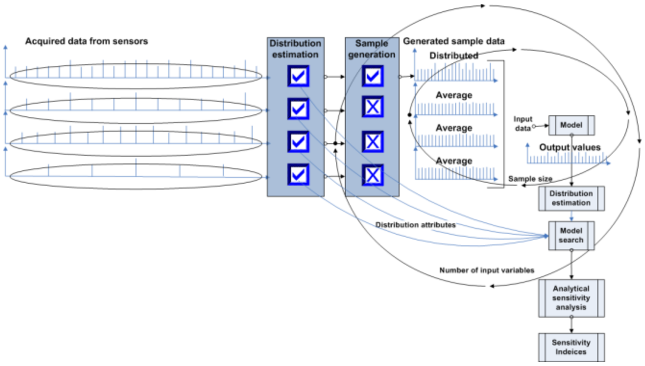


Fig. 1 General view of Monte Carlo method for sensitivity analysis

One significant challenge faced when applying Monte Carlo methods in real-time applications is the very effort required to estimate the distribution of the input variables prior to sample generation. In the application of the Monte Carlo sampling method to sensitivity analysis, in order to infer the impact of each input variable on the output variable, data samples of only one input variable (the checked box in Figure 1) is generated at a time whilst the other input variables (the cross-marked boxes in Figure 1) are held at a fixed value; for example the average value. This cycle repeats for each input variable. Reference [29] refers to this feature as a 'double-loop nested sampling procedure' which can potentially be very computationally expensive, particularly with large numbers and higher dimension of input variables.

## E. Morris Method

In the Morris method, as a parameter screening technique [17], changes in the value of an output variable is measured per changes in each input variable. Changes of only one input variable $(\theta)$ is applied to the equation $EE_i(\theta)=\frac{y(\theta+\Delta)-y(\theta)}{\Delta}$ to calculate values of the elementary effect ( $EE_i$ ), with input step change size dictated by $\Delta$ [10]. The resulting set of $EE_i$ values are then processed for distribution estimation. Each cycle of output distribution estimation requires $M=2rn$ model executions, where $r$ represents the number of output values required for the estimation of a stable distribution and $n$ is the number input variables [17]. More economical extensions of the Morris method can reduce the total number of cycles; for example by using each generated model output in more than one calculation [18]. However, a typically low value for $M$ could be as high as 21000 executions (1000 output values and 20 inputs applied to $M=r(n+1)$ in an improved Morris method [17]). Thus the Morris method cannot satisfy the requirements of sensitivity analysis in a time-constrained application.

## F. Analysis of Variance (ANOVA) Methods

The One-At-a-Time (OAT) method of processing input variables may at times be incapable of capturing the complexity of the relationship that exists between multiple input variables (i.e. second and higher orders) and an output variable. By decomposing and measuring the variance of the output distribution, a number of SA methods separately relate individual input variables to output variables [6]. The ANOVA based SA methods follow this logic and are in general more computationally efficient [5].

To achieve the decomposition elements and determine the corresponding sensitivity indices, when no explicit relationship exists between inputs and output (i.e. when an analytical approach is not possible), a numerical approach that in general is based on sample generation (i.e. Monte Carlo) can be adopted [4]. Using this technique, the level of computational overhead, in terms of model runs required to produce output values per each input sample grows rapidly [6]. For example, with 10 input variables and 1000 samples, the number of model execution runs is 1,024,000; a significantly high value. Therefore this method is not attractive for use in real-time applications.

## G. Fourier Amplitude Sensitivity Test (FAST)

Fourier Amplitude Sensitivity Test (FAST) [3] and its extended version [15] are examples of improvements in computational efficiency of the ANOVA-based SA methods. FAST (and extended FAST) are distinguishable from other ANOVA methods by their input data sample generation scheme, in which, samples for each input variable are generated according to a periodic function within the limits of the input variable [17]. In other words, in the FAST method the data distribution of input variables cannot be estimated from the acquired historical data. Instead, all distributions of input variables are considered to be uniform and within a specified range. The subsequently generated samples in this range follow a periodical function [30]. The periodic nature of the sample generation scheme (i.e. change of $s$ ) causes the model output values (for each $i$ ) to be periodic in terms of $s$. Therefore, by using numerical Fourier analysis on the values of the outputs, the magnitude of the Fourier spectrum at each frequency $w_i$ represents the sensitivity index of the corresponding input variable. Components in this process are shown in Figure 2.

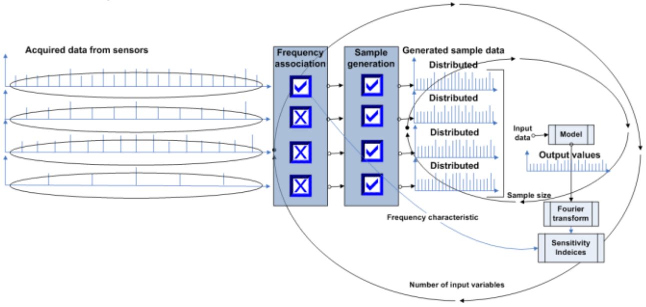


Fig. 2 General view of FAST method for sensitivity analysis

As is shown in Figure 2, some aspects of the computational cost that exist with the Monte Carlo method (i.e. the distribution estimation) is omitted by the FAST method and is replaced with the simpler tasks of boundary detection and frequency association. Furthermore, for finding the Fourier spectrum, the output value distribution estimation is also replaced by a numerical Fourier Transform (FT) method. In order to explicitly identify the power coefficient associated with the frequency of each input variable, the unique frequencies ( $w_i$ ) need to be correctly chosen. Typically the range of frequencies $w_i$ is divided into high and low ranges. A high frequency is assigned to the input variable subject to power spectrum coefficient identification and the remaining input variables are assigned a frequency from the low range. In this way the distance between the high frequency and all other low frequencies of inputs within the spectrum allows clear identification of the coefficient, or sensitivity indices. In Figure 2 the checked box representing the frequency association module shows that input variable number 1 has a high frequency of occurrence (in the generated sample) as compared to other input variables (depicted by crossed boxes). As a result the power coefficient of the frequency for input variable number 1 can be inferred with high confidence.

Comparing FAST with sampling-based SA methods, it appears that the number of model executions required is high [30]. The reason for this can be attributed to the 'double-loop' nested sampling procedure [29]. On the other hand, the computational overhead of the FAST method is lower than sampling-based SA methods due to the simpler tasks involved in the nested loops. Sample generation and Fourier transform in FAST are usually less computationally costly than the tasks of sample generation, distribution estimation, and distribution-based function fitting (i.e. searching for a suitable model). References [28] and [29] tackle the issue associated with the computational cost of the '*double loop sample generation strategy*' and the restrictive conditions that apply in the evaluation of dependent variables based on independent variables. In sampling-based SA methods this is addressed by proposing an approximation approach that measures the entropy of variable distributions from original samples. The method uses the same decomposition equation as discussed in section F, the only difference is that the determination of variance in the sample data distributions is replaced by determination of entropy. This appears to have helped in reducing computational overheads.

### H. *Entropy-Based Epistemic Sensitivity Analysis*

In order to determine sensitivity indices then one only needs to establish the values of independent input variables (denoted by $X$ ) and dependent output variables (denoted by $Y$ ) [28]. The sensitivity indices using the Entropy method can then be calculated using $H(Y) - H(Y \mid X)$. Where $H(Y)$ are the entropy values and $H(Y \mid X)$ are the values of conditional entropy.

The method replaces the time consuming sample generation of $X$ and evaluation of $Y$ by Simple-Random Sampling (SRS) using piecewise uniform density function estimations.

Figure 3 shows that only a single execution is required to generate sufficient samples for estimation of the sensitivity indices. Reference [29] demonstrates the feasibility of the estimation approach in a test case with fifteen independent and two dependent variables. Reasonable results were achieved with far lower computational cost. However, obtaining the appropriate indicator functions for each independent variable requires prior knowledge of their probability distributions [29].

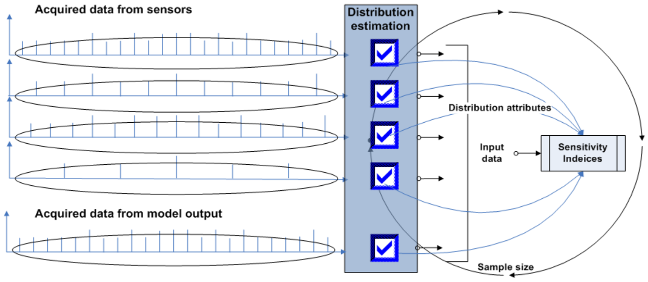


Fig. 3 General view of Entropy-based method for sensitivity analysis

## III. Methodology for Event Tracking Sensitivity Analysis (EventTracker)

The proposed event tracking SA method uses an input-output occurrence [+, -] matrix. This matrix is populated at predefined time intervals. The current platform is designed to allow a user (with domain knowledge) to set the initial system update time interval. For example, in safety sensitive systems such as power plant reactor monitoring, the rate of populating the data tables will be a short interval. Whereas in scenarios that employ less time critical systems, such as finance, then the interval will be longer. This matrix is designed to map the relationships between causes that trigger events (Trigger Data) and the data that describes the actual events (Event Data). In this way the 'EventTracker' method is able to construct a discrete event framework where events are loosely coupled with respect to their triggers for the purpose of sensitivity analysis. A description of Discrete Event System, Trigger Data, and Event Data are provided in the following subsections.

### A. *Discrete Event Systems*

As opposed to continuous systems, a Discrete Event System (DES) is defined by the disparate occurrence of events in a specified time span. In other words, the state of the system changes when the input variables and consequently the outputs of the system change. Each state transition of the system is called an event. Therefore, in DES, only the attributes that represent the occurrence of an event are considered. These attributes are discussed in the following section.

### B. *Trigger Data and Event Data*

Any input variable whose value results in the registration of an event is defined as Trigger Data (TD) in our DES. The series of data that represent the state of the system at a given time is described as Event Data (ED). It is possible that the numbers of EDs and TDs in a system are different. For example, a number of TD series may be responsible for changing a single ED series. It should be noted that various

TD series could have differing impact on specified ED series.

$$ED \because \{TD_1, TD_2, \ldots, TD_n\} \qquad (1)$$

This is because individual or combination of input variables may have different effects on different system outputs.

### C. *An Example of a Baking Process*

Here an example of a baking process will be used to explain and illustrate the underpinning rationale for the proposed ‘EventTracker’ method.

One of the ways to detect system state transitions is to detect and track the changes in its input variables. Figure 4 is a simplified illustration of a baking machine with a single heater. Two light reflector sensors (S01 and S02) are installed on the machine; the sensors send signals to the EventTracker software model. Sensors S01 and S02 provide data relating to the entry and exit of ‘components’ into and from the baking machine. Their signal data either carries no voltage (i.e. binary 0) or a pulse of voltage (i.e. binary 1) indicating the presence of a component entering or exiting the baking machine. The occurrence of these respective signals (i.e. events) determines the duration of the baking process (Baking Time).

The combination of the data provided by the two sensors is used to measure a production process performance factor. This performance factor is the instantaneous resource utilization (RU) of the baking machine. The baking machine utilization is defined as the ratio of the total heater occupancy in relation to the overall capacity of the baking machine [11].

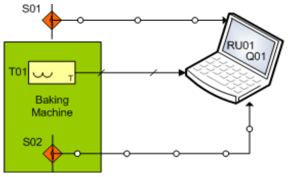


Fig. 4 An imaginary Baking System with two sensors

Figure 5 shows the relationship between each event triggered by S01 and S02 with respect to changes in RU. Each change to the RU in a given time span can be expressed as an event and the positive value of the S01 and S02 sensor inputs as triggers, then RU can be defined as Event Data (ED). Both S01 and S02 can be considered as Trigger Data (TD).

$$\begin{array}{l} if(S01_t - S01_{t-1}) \geq \theta \xrightarrow{Trigger} TD_t \\ if(RU_t - RU_{t-1}) \geq \Psi \xrightarrow{Event} ED_t \end{array} \qquad (2)$$

Where, $S01_t$ is the S01 signal at time $t$, $\theta$ is the signal change threshold, $RU_t$ is the resource utilization at time $t$, and $\Psi$ is the utilization change threshold.

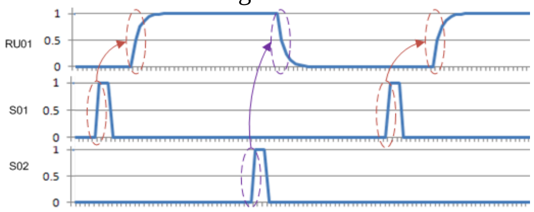


Fig. 5 Causal relationship between two switch signal data S01, S02, and the performance factor RU

### D. *Methods and Parameters for Event Tracking*

The EventTracker platform is based on four functional parameters that are initialised by a user with domain knowledge. The Search Slot (SS) and the Analysis Span (AS) parameters are about tracing the values of the acquired data series. Whereas the remaining two parameters Event Threshold (ET) and Trigger Threshold (TT) are about the magnitude of transition detection and the overall system state analysis. Subsequently these parameters are automatically optimised by the EventTracker platform as discussed further in sections V and VI.

- *Search Slot (SS)*

The SS is a fixed time slot within which batches of TD and ED are captured. It can also be described as the scan rate. The scan rate is determined by a system expert.

- *Analysis Span (AS)*

The AS is the time span within which a period of sensitivity analysis occurs. An analysis span is comprised of a number of consecutive SS. The number of TD and ED observation will then be used to determine and apply sensitivity indices at the end of an AS. The new sensitivity indices are assigned to the TD and carried forward, in other words there is a possibility that the sensitivity indices of $TD_t$ is different from $TD_{t+1}$ (see Table II).

- *Event Threshold (ET)*

The fluctuations in the ED series that are interpreted as triggers are determined in comparison with the Event Threshold (ET). This value is expressed as a proportion of the overall range of ED series values occurring in an AS. It is expressed as a percentage.

- *Trigger Threshold (TT)*

The fluctuations in the TD series that are interpreted as triggers are determined in comparison with the Trigger Threshold (TT). TT (like ET) is expressed as a percentage of the overall range of TD series values occurring in an AS.

These thresholds determine whether a signal represents a real change in the system state or not. Given the system state changes then it is assumed an event has occurred.

### E. *The Assumptions of the Proposed Method*

The EventTracker method is based on a number of assumptions. These are listed as:

*Assumption 1- Triggers and Events:*

Only those fluctuations in the data series that are interpreted as triggers (TD data series) and as events (ED data series) are taken into account. The basis for this interpretation is the threshold (ET and TT) settings.

*Assumption 2- Thresholds:*

Thresholds are pre-specified, but there values are short lived and are dependent on signal fluctuation in the data series. ET and TT are evaluated once every AS on the assumption that within that period a representative range of fluctuations in the data series is likely to occur. Therefore, a trigger or an event occurs when the difference between the maximum value and the minimum value of a data series within a SS exceeds the associated data series threshold.

*Assumption 3-Homogeneity of Data Series:*

The threshold value for each data series remains fixed for the period of the AS. This implies that in all search slots of a data series, the range of possible values in which a transition may occur is assumed to be a fixed co-variant. In other words, each data series is assumed to have the same probability distribution over all AS.

### F. *EventTracker Algorithm*

The algorithm is designed to respond quickly and in essence has a life cycle that is equivalent to an AS. This life cycle is divided into several SS. Within each slot, TDs and EDs are captured from two time series and used to provide a value which is translated into a sensitivity index. This index is then added to the indices of subsequent search slots. At the end of each AS, the sensitivity indices of all data series are linearly normalized. The main functions of the EventTracker algorithm are depicted in figures 6 and 7. The main steps of the algorithm are as follows:

*Stepwise Scan*

A First-In-First-Out queue is allocated for every batch of data in a search slot. The size of the queues is unbounded. The content of the queues are flushed at the end of each search slot. The data is then passed to the EventTracker detection and scoring algorithm. The next search slot continues to fill the queue immediately. Using this technique no data is lost. Figure 6 shows a few stepwise scans and their analysis operations in the search slots.

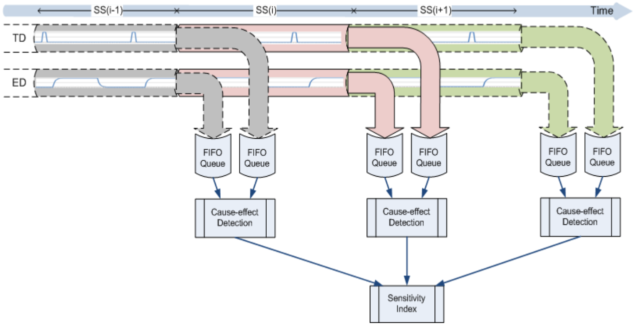


Fig. 6 Overall functionality diagram of EventTracker algorithm

*Trigger-Event Detection*

Figure 7 shows that within each SS a pair of {ED, TD} are examined for evidence of trigger and event. The batch of TD values is searched for fluctuations greater than the specified TT threshold, and ED values similarly checked for changes larger than the ET threshold. This functionality results in a true value being generated provided at least one of the above changes is found in a particular batch.

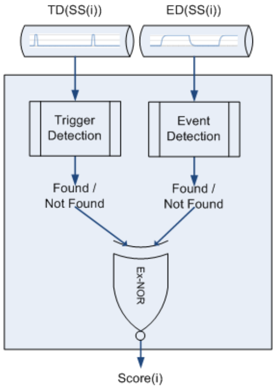


Fig. 7 Trigger-Event Detection functionality on each Search Slot

*Two-way Matching Score*

In each SS the simultaneous existence or non-existence of a change in each pair of data batches is scored as +1, otherwise the score is -1. This operation is similar to a weighted logical Exclusive-NOR and is shown in Table I. This approach is adopted to better emphasize the impact of inputs on a given output rather than simply scoring +1 for existence and 0 for non-existence.

TABLE I
WEIGHTED EXCLUSIVE-NOR FUNCTIONALITY

| Input 1 | Input 2 | Output |
|---|---|---|
| 0 | 0 | +1 |
| 0 | 1 | -1 |
| 1 | 0 | -1 |
| 1 | 1 | +1 |

*Summation of Two-way Matching Scores*

The +1 and -1 score for each SS is added to the overall score depicted by equation (3). Sensitivity Index (SI) of the measured ED and TD values after time $t$ (or in discrete form after search slot $n$ ). Where $n$ is the number of SS in an AS. SI can be calculated as:

$$SI_{(t)} = \sum_{1}^{n} Search\ Slot\ Scores \qquad (3)$$

*The Normalization Process*

At the end of each SS the values of the sensitivity indices are linearly scaled to the unit range (4). In other words, given a lower bound $l$ and an upper bound $u$ for the set of all indices, each final value of sensitivity index is transformed to a value in the range [0,1]; thus:

$$\tilde{S} = \frac{SI - l}{u - l} \qquad (4)$$

A summary of the algorithm performance is shown in Table II. In this table the flow of matching scores and sensitivity indices (SI1, SI2, SI3) for one ED with respect to three TDs (TD1, TD2, TD3) over 10 SS is shown. Star symbols in Table II indicate a detected event or trigger in the values of ED, TD1, TD2 and TD3 within each search slot. Each value of S1, S2 and S3 is -1 or +1 depending on the exclusive match between ED and TD1, TD2 and TD3 respectively. SIn1 to SIn3 represent Normalized Sensitivity Indices values for SI1 to SI3.

TABLE II
AN EXAMPLE PRODUCTION OF SENSITIVITY INDEX BY EVENTTRACKER METHOD

| Search Slot | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ED | * | * |  | * |  | * | * | * |  | * | * |
| TD1 |  |  | * |  |  | * |  |  | * | * | * |
| S1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 |
| SI1 | -1 | -2 | -3 | -4 | -3 | -2 | -3 | -4 | -5 | -4 | -3 |
| SIn1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TD2 | * |  |  |  | * | * | * | * |  | * |  |
| S2 | 1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 |
| SI2 | 1 | 0 | 1 | 0 | -1 | 0 | 1 | 2 | 3 | 4 | 3 |
| SIn2 | 1.00 | 1.00 | 1.00 | 0.67 | 0.33 | 0.33 | 0.67 | 0.75 | 0.80 | 0.80 | 0.75 |
| TD3 |  | * |  | * |  | * |  | * |  | * |  |
| S3 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | 1 | 1 | -1 |
| SI3 | -1 | 0 | 1 | 2 | 3 | 4 | 3 | 4 | 5 | 6 | 5 |
| SIn3 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

The normalized sensitivity indices (SIn) in Table II show that ED is the most sensitive to TD3 and least sensitive to TD1. Figure 8 shows the values of SIn.

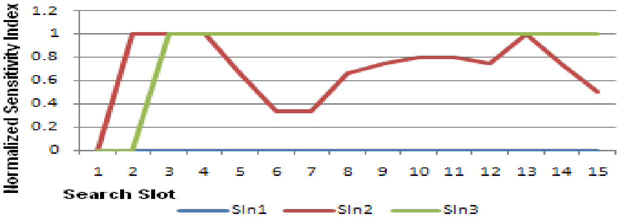


Fig. 8 Normalized sensitivity indices as in Table II

The overall average SIn values are shown in figure 9, this figure illustrate the lateral movement of the respective values towards a value that is analogous to a steady state.

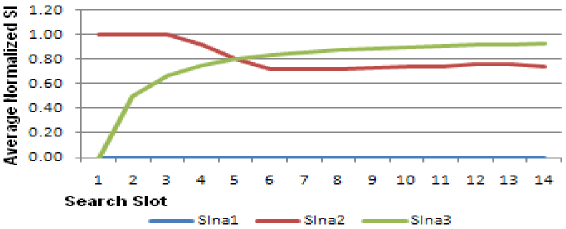


Fig. 9 Averaged normalized sensitivity indices as in Table II

In situations where normalized indices are not a desirable means to determine the values of sensitivity index, then the current or instantaneous value can alternately be used.

## IV. A CASE STUDY FOR EVENTTRACKER

A side-panel manufacturing line in a refrigerator production plant was used in a real world application of the EventTracker platform. The process layout in this plant groups machines into six sequential operations. The four machines that process the material: GR-1 (Loader), GR-2 (Shearing Unit), GR-3A (Tilter) and GR-3(Forming Machine) were subject to this study [12]. Table III (appendix) summarizes the type and purpose of each input signal. All signals used in the side-panel manufacturing line carry binary data of either one or two bit magnitude. The following section discusses how these signals are interpreted as triggers and events that describe the production system.

### A. *Side-panel Manufacturing Process Discrete-Event Model*

A real-time discrete-event model of the production process was developed to facilitate the measurement of a number of production performance parameters. The proposed scenario coupled with the signals and their location with respect to the model is depicted in the following Event-driven Process Chain (EPC) diagrams [figures 10-13]. Each diagram shows a process and its constituent operations. Also shown are the signals that trigger start and completion of tasks (i.e. events). The parameter chosen for production performance analysis is the Instantaneous Resource Utilization.

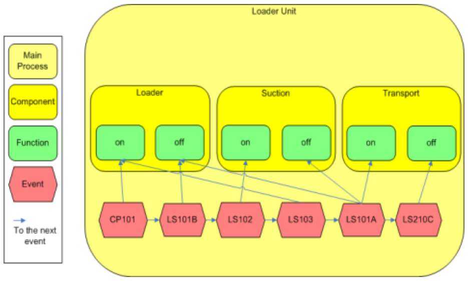


Fig. 10 Loader processes and events

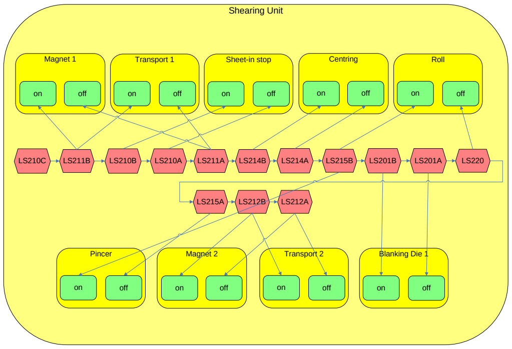


Fig. 11 Shearing Unit processes and events

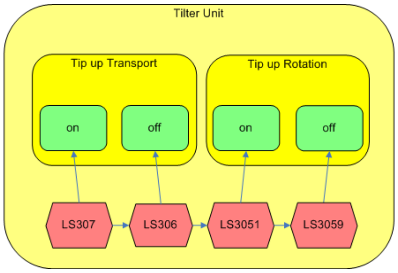


Fig. 12 Tilter Unit processes and events

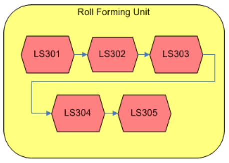


Fig. 13 Roll Forming Unit processes and events

The input variables are represented by 28 signals generated from sensors installed on the production line. National Instruments LabVIEW software tool was used to develop the data acquisition platform to collect the signal data from the shop floor. The acquired data is fed into a commercial discrete event simulation package (Arena™ Rockwell Automation). Previous integration work has addressed the need for direct translation of multiple input signals into inputs of the simulation model [13]. The so-called Real-time Model Matching Mechanism (R3M) [13] was implemented to clearly define the relationship between signals and processes in the model. For example, the model is capable of relating signals from LS301 and LS305 that relate to the start and end of the forming process shown in figure 13. However, it should be noted that this awareness is not used by the EventTracker method to estimate sensitivity indices. Simply that the model provides the means to validate the work, insomuch as each TD with a higher importance level is reported, whilst the less important TDs are classified as 'false negatives'.

### B. *Real-time Data Streaming into the Modeller*

The data from the side-panel manufacturing line is fed into the model in real-time during a period of 2 minutes (i.e. 500 data points giving a rate of 5 samples per second). The event data collected was then used to measure the instantaneous utilization of 4 machines in the production line. The 28 TD input values were linked to four ED series.

The implementation of EventTracker on the production line is shown in figure 14. The data modeller generates a two dimensional array of sensitivity indices in the time domain. To find the most suitable ET and TT values, production engineers were consulted and a number of production cycles were observed. One future improvement to the current platform is the provision of the necessary functionality to measure ETs and TTs automatically with respect to the collected data.

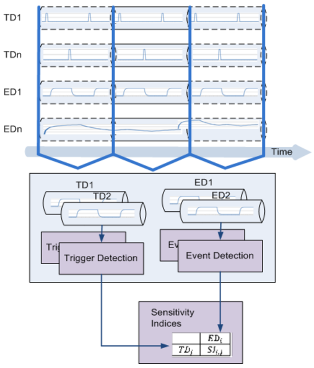


Fig. 14 Implementation of EventTracker for multiple Input Variables and Output Parameters

## V. Assessment of Efficiency and Validity of the Event Tracking Sensitivity Process

Efficiency tests for the EventTracker SA method were carried out to assess whether the method resulted in increased computational speed and reduced overheads. A validity test was also performed to ensure that the quality of analysis was as reliable and robust as that obtained when using a similar established SA method. The key objective of EventTracker is to record all events, but filter out the least important ones so that the optimum number of data points can be transferred to the data processing unit.

The following steps were performed to validate the event filtering algorithm used in EventTracker:

**Step 1:** Establish the maximum number of least important trigger data and eliminate them from the analysis.

**Step 2:** Find the optimum SS, ET and TT values.

**Step 3:** Compare the results with a scenario where the total number of TDs have been included in the analysis (i.e. no reduction in data load).

This last step was used to validate the filtering process in steps 1 and 2.

In order to determine the maximum number of the least important TD series, EventTracker was configured to read all 28 TD series (see Table III, appendix) and generate sensitivity indices for the four ED series. ET and TT values of 50% and a SS period of 5 seconds were used in this example. The results are shown in figure 15. The four line charts represent the values of normalized sensitivity indices for the four ED series (machine utilization). The normalized sensitivity indices are scaled according to the vertical axis.

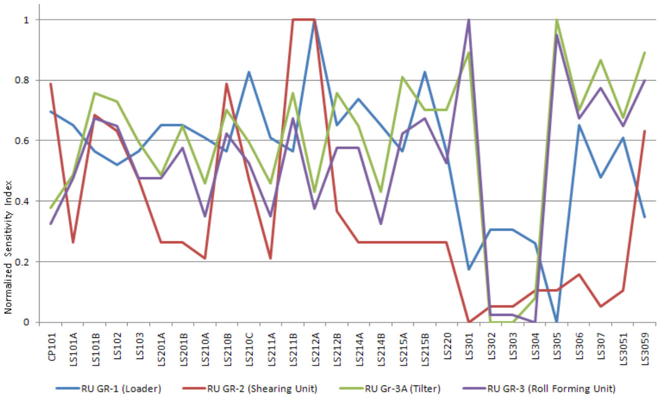


Fig. 15 EventTracker Sensitivity Indices of 4 EDs with respect to 28 TDs

Figure 15, show that the event data is not sensitive to all of the 28 TDs used in the example. This process allows us to eliminate the unimportant trigger data from the analysis.

A cut-off threshold (CT) is defined for each series of indices within an ED series. Their values lie between the minimum and maximum index values for that range. As in (5):

$$CT = Min(SI_{ED}) + CR^{*}(Max(SI_{ED}) - Min(SI_{ED})) \quad (5)$$

Where, $CR$ is the Cut-off Ratio in the range $0 \le CR \le 1$. For example, if CR is 0.5, then the value of the cut-off thresholds are all in the middle of their associated sensitivity indices range. Figure 16 and 17 show the Normalized Sensitivity Indices (NSI) for ED series of the machines in the production line (i.e. RUGR1 the Loader and RUGR3-A the Tilter) respectively. These values show the importance of each sensor in calculating the utilisation of each machine. In both charts the NSI is in the range 0 to 1. The CT value for both series is 0.5 (and is depicted by the green dashed line). For a normalized range then the values of CT and CR are effectively the same. The chart in figure 16 shows that seven TDs (red bars) are below the threshold and are considered as least important for the ED series that are used to determine the utilisation of RUGR1. In figure 17, we can see that 10 TDs (red bars) are below the threshold and are considered as least important for the ED series RUGR3A. Three TDs (i.e. LS302, LS303, and LS304) are common to both ED series as least important TDs.

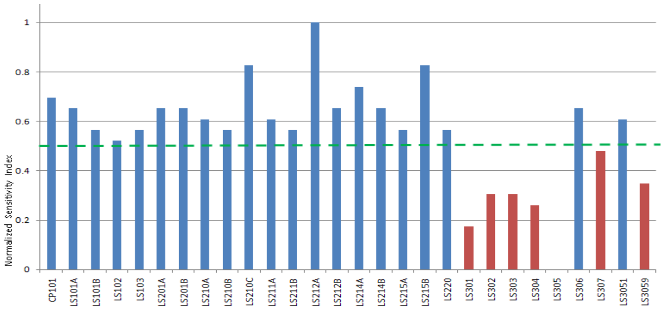


Fig. 16 Normalized sensitivity indices of RUGR1 with respect to 28 TDs

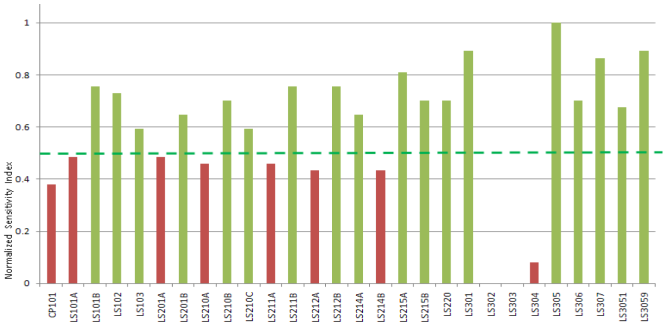


Fig. 17 Normalized sensitivity indices of RUGR3-A with respect to 28 TDs

Reviewing the results from figures 16 and 17, one may conclude that the sensitivity analysis may have generated false negatives. In particular we can observe that a number of the TDs (red bars) have NSI values greater than 0, but below the threshold.

In order to measure and eliminate false negatives from the system a false negative test was conducted, the results of which are shown in figure 18. This figure shows the percentage of least important TDs with respect to different values of CT. The higher the CT value, the greater is the percentage of TDs that have been filtered out.

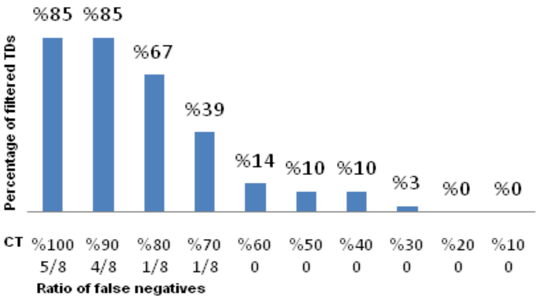


Fig. 18 Percentage of filtered TDs per CT and Ratio of False Negative

Figure 18 shows the percentage of filtered TDs with respect to different CTs. For example, with a CT value of 70%, 39% of the TDS are filtered or considered as less important. But in reality, 1 of the TDs that is influential has also been filtered in this case (i.e. the ratio of 1/8 in the lower part of x axis, which shows the percentage of false negatives). This percentage of false negative is high and undesirable. Experimentation revealed that for this industrial scenario, with a CT of 60%, the percentage of false negative falls to 0 In other words we have detected that at least one of the originally eliminated TDs has significant effect on our ED and should be re-instated for the purpose of sensitivity analysis. The results of the experimentation are reported in Table IV (appendix).

### *A. Sensitivity of EventTracker Method to the Method Parameters*

In order to test the dependency of EventTracker on its parameters, sensitivity indices resulting from different values of ET, TT and SS were compared. Figure 19 shows the percentages of less important TDs based on different values of ET and TT over differing values of CR. Figure 20 shows the percentages of less important TDs based on different values of SS over differing values of CR. It appears from Figure 19[1] that ET and TT values do not make a significant difference to the indices, whilst SS values have a greater impact on the indices (Figure 20). The region of no false negatives (the three thicker line charts in figure 20) have at least 1 TD that needs to be considered for re-instatement (as per Figure 18). Figure 20 suggests that the SS value should not be shorter than 2 seconds and not greater than 8 seconds in order to achieve the best savings in computational overhead.

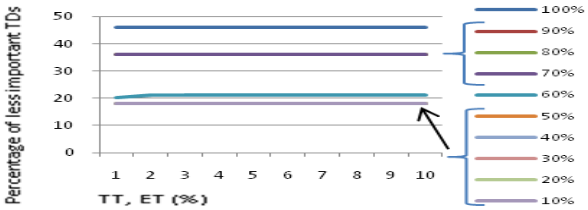


Fig. 19 Percentages of less important TDs per percentage of ET and TT with different CR values

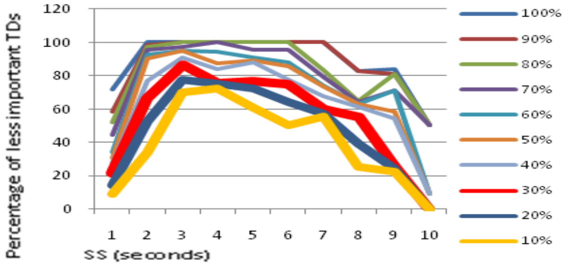


Fig. 20 Percentages of less important TDs per percentage of TS with different CR values

### *B. EventTracker Method Before and After Input Variable Selection*

Following analysis 4 TDs were discarded, the remaining 24 (Table IV, appendix) were used to measure the instantaneous machine utilization. The utilization of the 4 machines in the first instance with the full 28 TDs was compared with the short-listed 24 TDs. The results are shown in Figure 25 (appendix). This figure shows that the accuracy of calculations was not compromised by using 24 rather than the full 28 TDs. With the full 28 TDs the EventTracker algorithm took 6.875

[1] Similarity of values of the data series caused matching line charts

seconds to calculate the utilizations, whereas when using 24 TDs only 3.5625 seconds were spent to achieve the same results. By eliminating the 4 ‘redundant’ sensor values from the calculations, a reduction of approximately 52% was achieved in computation time. The average CPU utilization remained almost constant during the period of analysis. In addition to reducing the computational time, the algorithm achieved reductions in communication load and more importantly reduced the number of sensors required on the production line. This reduction in data acquisition equipment subsequently saves installation, maintenance and a reduction in the complexity of the supervisory control and data acquisition (SCADA) systems required by the industrial plant.

## VI. A Comparison between EventTracker and Entropy-based Sensitivity Analysis (ESA) methods

To validate the proposed sensitivity analysis method a comparison between EventTracker and an Entropy-based Sensitivity Analysis (ESA) method was conducted. The rational for choosing ESA over other SA techniques such as ANOVA, is the similarity that exists between EventTracker and ESA. EventTracker also does not have a reliance on the availability of statistically reliable or the homoscedasticity of historical data [15], [16], [17].

An Entropy-based sensitivity analysis method had been proposed by [8]. In this method the sensitivity index of a model output with respect to a model input is defined as the reduction in the entropy of the output, given the input does not have any uncertainty (i.e. when its values are all known). Further details of the method can be found at [8]. Although this method (like ANOVA-based methods) needs analytical determination of the density functions associated with the input and output series, [8] has proposed a method for the direct estimation of the sensitivity index from samples. The ESA estimation method is implemented as part of this work for the purpose of performance analysis comparison with the EventTracker method. The results of the ESA method are shown in figures 21 and 22.

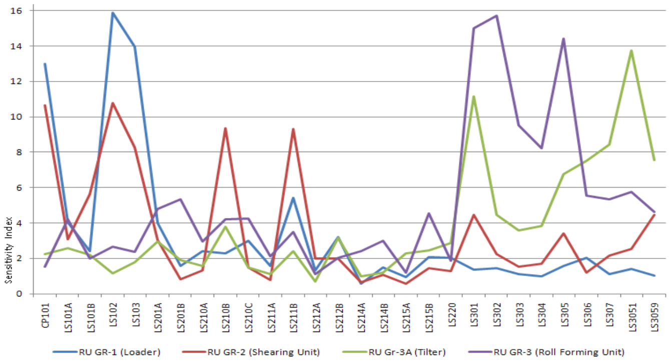


Fig. 21 ESA Sensitivity Indices of 4 EDs with respect to 28 TDs

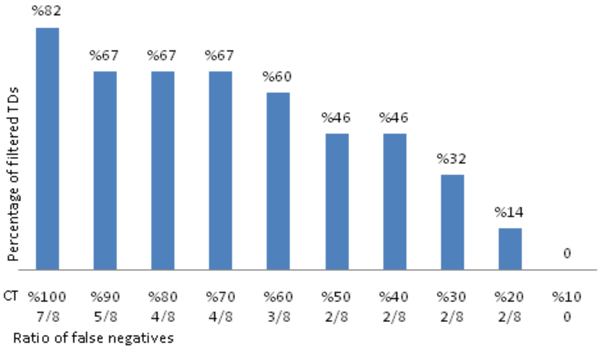


Fig. 22 ESA Percentage of filtered TDs per CT

The EventTracker and ESA methods are compared in figure 23 based on the region of ‘no false negatives’. The figure shows that on average the ESA method filters out more of the TDs. It also shows that the ESA method produces more false negatives. The EventTracker method reports up to 14% of TDs as less important without any false negative, whereas the ESA method produces 37.5% false negatives (i.e. 3 out of 8).

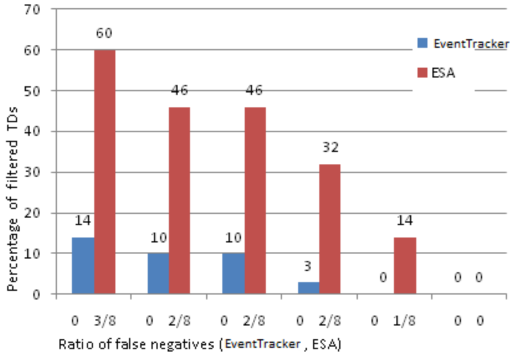


Fig. 23 Comparison of proportion of less important TDs with low false negative ratios on EventTracker and ESA methods

In comparing the levels of CPU usage between the two methods, it was observed that the ESA method continuously took on average up to 50% of the available CPU output for a 1348.87 seconds duration. The EventTracker method took 55% of the available CPU output, but for a much shorter period of 6.875 seconds. With a typical sampling rate of five samples per second, the ESA method would appears to be less efficient in comparison to the EventTracker method when used for real-time analysis.

## VII. Conclusion

This paper proposes a sensitivity analysis (SA) methodology for use in large scale ‘real-time’ data acquisition systems. The method in comparison to Entropy based SA (ESA) technique was shown to be faster, more accurate and less computationally burdensome. The reason that ESA was used as the basis for comparison is that like EventTracker, the ESA method is a SA method that relies least on historical data. The underpinning logic behind the EventTracker method is the capture of cause-effect relationships between input variables (triggers) and output variables (events) over a given period of time.

EventTracker is an event-driven sensitivity analysis method and not a probability-based approach. The process is deterministic in the sense that it is only instigated when an event with a pre-determined threshold is detected. There is no reliance on statistical or model based equations, only on the interpretation of transition between system states and in that sense the technique is completely "*unaware*".

One of the strengths of the proposed method is the freedom of choice it offers the user to specify a scan rate based on the very nature of the application itself. For example, applications such as weather or financial forecasting require longer intervals between events; whilst others such as reactor safety systems in power plants require a shorter scan interval. The platform in its current form provides the flexibility for a system analyst to choose an appropriate value based on their experience and local knowledge. As part of future work, one objective is to develop an autonomous and intelligent scheduling method that finds the optimal scan rate based on the data collected directly from the system.

A key feature of the technique is its ability to rapidly filter inconsequential data, data that at times may very well overwhelm the data processing platform. With regard to the time domain, the EventTracker method may be classified as a Local Sensitivity Analysis (LSA) method. Moreover in estimating sensitivity indices, EventTracker does not require prior knowledge of the analytical or statistical relationship that may very well exist between input and output variables. EventTracker in this sense can be considered to be a truly Global Sensitivity Analysis method. The approach does not require any prior estimation of the data distribution (see figure 24).

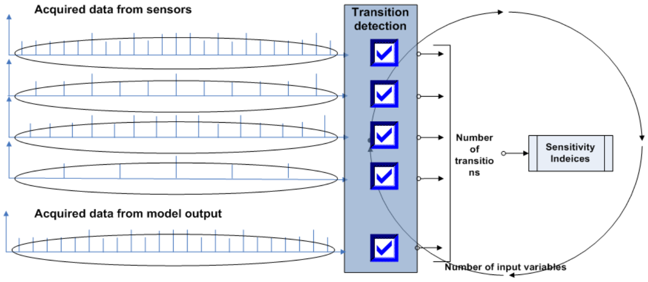

Acquired data from sensors
Acquired data from model output
Transition detection
Number of transitions
Sensitivity Indeices
Number of input variables

Fig. 24 General view of EventTracker method for sensitivity analysis

The performance model is capable of meeting the demands of 'real-time' execution. This approach to sensitivity analysis can be used in large scale distributed data analysis, such as climate change analysis, global manufacturing and logistics operations or interlinked financial applications.

One key advantage of the method is the reduction in cost, complexity of installation and maintenance of any associated SCADA systems.

## ACKNOWLEDGMENT

The authors acknowledge the financial and continued support provided by the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom.

## APPENDIX

TABLE III: LIST OF SIGNALS FOR FOUR MACHINES OPERATIONS THE SIDE-PANEL MANUFACTURING LINE

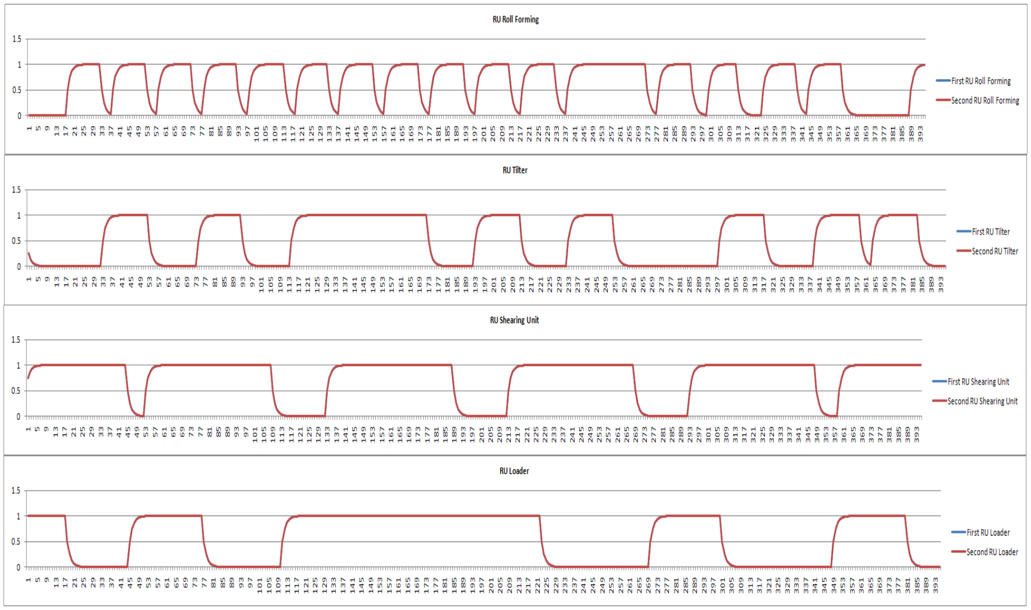
| Task Group | Signal | Sensor/Actuator | Role | Data Type |
|---|---|---|---|---|
| GR-1 | CP101 | Actuator | Manual loader up/down | 2-bit digital |
| | LS101A | Sensor | Loader up | 1-bit digital |
| | LS101B | Sensor | Loader down | 1-bit digital |
| | LS102 | Sensor | Sheet presence (align) | 1-bit digital |
| | LSDP | Sensor | Double sheet | 1-bit digital |
| | M101 | Actuator | Transport frw/rev | 2-bit digital |
| | M102 | Actuator | Manual trolley frw/rev | 2-bit digital |
| GR-2 | M201 | Actuator | Manual transport 1 | 2-bit digital |
| | M202 | Actuator | Manual transport 2 | 2-bit digital |
| | CP210 | Actuator | Sheet-in stopped up/down | 2-bit digital |
| | LS210A | Sensor | Sheet-in stopped up | 1-bit digital |
| | LS210B | Sensor | Sheet-in stopped down | 1-bit digital |
| | LS210C | Sensor | Slowing stopped CP210 | 1-bit digital |
| | CP211 | Actuator | Manual magnet 1 up/down | 2-bit digital |
| | LS211A | Sensor | Magnet 1 CP211 up | 1-bit digital |
| | LS211B | Sensor | Magnet 1 CP211 down | 1-bit digital |
| | CP212-213 | Actuator | Manual magnet 2 up/down | 2-bit digital |
| | LS212A | Sensor | Magnet 2 CP212 up | 1-bit digital |
| | LS212B | Sensor | Magnet 2 CP212 down | 1-bit digital |
| | LS213A | Sensor | Magnet 2 CP213 up | 1-bit digital |
| | LS213B | Sensor | Magnet 2 CP213 down | 1-bit digital |
| | CP214 | Actuator | Manual centring forw/rev | 2-bit digital |
| | LS214A | Sensor | Centring forw | 1-bit digital |
| | LS214B | Sensor | Centring rev | 1-bit digital |
| | CP215 | Actuator | Manual pincer open/close | 2-bit digital |

| | | | | |
|---|---|---|---|---|
| | LS215A | Sensor | Pincer open | 1-bit digital |
| | LS215B | Sensor | Pincer close | 1-bit digital |
| | LS220 | Sensor | Out block | 1-bit digital |
| | M203 | Actuator | Manual roll machine forw/rev | 2-bit digital |
| | LS203C | Sensor | Axel manilpulator forw | 1-bit digital |
| | LS203D | Sensor | Axel manilpulator rev | 1-bit digital |
| | LS203E | Sensor | Axel manilpulator home | 1-bit digital |
| | M204 | Actuator | Axel move guide forw/rev | 2-bit digital |
| | LS204C | Sensor | Axel move guide forw | 1-bit digital |
| | LS204D | Sensor | Axel move guide rev | 1-bit digital |
| | LS204E | Sensor | Axel move guide home | 1-bit digital |
| | M207 | Actuator | Axel centring forw/rev | 2-bit digital |
| | LS207C | Sensor | Axel centring forw | 1-bit digital |
| | LS207D | Sensor | Axel centring rev | 1-bit digital |
| | LS207E | Sensor | Axel centring home | 1-bit digital |
| | CO201 | Actuator | Manual blanking die up/down | 2-bit digital |
| | LS201A | Sensor | Blanking cylinder CO201 up | 1-bit digital |
| | LS201B | Sensor | Blanking cylinder CO201 down | 1-bit digital |
| GR-3A | M303 | Actuator | Manual tip up transport forw/rev | 2-bit digital |
| | M304 | Actuator | Manual tip up rotation forw/rev | 2-bit digital |
| | CP305 | Actuator | Manual tip up block forw/rev | 2-bit digital |
| | LS305A | Sensor | Tip up block forw | 1-bit digital |
| | LS305I | Sensor | Tip up block rev | 1-bit digital |
| | LS303A | Sensor | Tip up rotation up | 1-bit digital |
| | LS303B | Sensor | Tip up rotation down | 1-bit digital |
| | LS303C | Sensor | Tip up rotation slowing up | 1-bit digital |
| | LS303D | Sensor | Tip up rotation slowing down | 1-bit digital |
| | LS306 | Sensor | Tip up stop outside | 1-bit digital |
| | LS307 | Sensor | Tip up stop inside | 1-bit digital |
| GR-3 | M301 | Actuator | Manual roll machine forw/rev | 2-bit digital |
| | LS301 | Sensor | Roll machine location 1 | 1-bit digital |
| | LS302 | Sensor | Roll machine location 2 | 1-bit digital |
| | LS303 | Sensor | Roll machine location 3 | 1-bit digital |
| | LS304 | Sensor | Roll machine location 4 | 1-bit digital |
| | LS305 | Sensor | Roll machine location 5 | 1-bit digital |
| | M302 | Actuator | Move guide forw/rev | 2-bit digital |
| | LS302C | Sensor | Move guide forw | 1-bit digital |
| | LS302D | Sensor | Move guide rev | 1-bit digital |
| | LS302E | Sensor | Move guide home | 1-bit digital |

TABLE IV: LIST OF MORE IMPORTANT TDS (GREEN CELLS) AND LESS IMPORTANT TDS (RED CELLS) REPORTED AFTER THE EVENTTRACKER METHOD HAD ANALYZED 28 PAIRS OF {ED, TD} WITH 60% CR. UNDERLINED NUMBERS INDICATE TRULY MORE IMPORTANT TDS ACCORDING TO THE MODEL STRUCTURE.

| | RU Loader | RU Shearing Unit | RU Tilter | RU Roll Forming | |
|---|---|---|---|---|---|
| **CP101** | **0.695652** | 0.789474 | 0.37838 | 0.325 | More Important |
| **LS101A** | 0.652174 | 0.263158 | 0.48649 | 0.475 | More Important |
| **LS101B** | 0.565217 | 0.684211 | 0.75676 | 0.675 | More Important |
| **LS102** | 0.521739 | 0.631579 | 0.72973 | 0.65 | More Important |
| **LS103** | 0.565217 | 0.473684 | 0.59459 | 0.475 | Less Important |
| **LS201A** | 0.652174 | 0.263158 | 0.48649 | 0.475 | More Important |
| **LS201B** | 0.652174 | 0.263158 | 0.64865 | 0.575 | More Important |
| **LS210A** | 0.608696 | 0.210526 | 0.45946 | 0.35 | More Important |
| **LS210B** | 0.565217 | 0.789474 | 0.7027 | 0.625 | More Important |
| **LS210C** | **0.826087** | 0.473684 | 0.59459 | 0.525 | More Important |
| **LS211A** | 0.608696 | 0.210526 | 0.45946 | 0.35 | More Important |
| **LS211B** | 0.565217 | **1** | 0.75676 | 0.675 | More Important |
| **LS212A** | 1 | **1** | 0.43243 | 0.375 | More Important |
| **LS212B** | 0.652174 | 0.368421 | 0.75676 | 0.575 | More Important |
| **LS214A** | 0.73913 | 0.263158 | 0.64865 | 0.575 | More Important |
| **LS214B** | 0.652174 | 0.263158 | 0.43243 | 0.325 | More Important |
| **LS215A** | 0.565217 | 0.263158 | 0.81081 | 0.625 | More Important |
| **LS215B** | 0.826087 | 0.263158 | 0.7027 | 0.675 | More Important |
| **LS220** | 0.565217 | 0.263158 | 0.7027 | 0.525 | More Important |
| **LS301** | 0.173913 | 0 | 0.89189 | **1** | More Important |
| **LS302** | 0.304348 | 0.052632 | 0 | 0.025 | Less Important |
| **LS303** | 0.304348 | 0.052632 | 0 | 0.025 | Less Important |
| **LS304** | 0.26087 | 0.105263 | 0.08108 | 0 | Less Important |
| **LS305** | 0 | 0.105263 | 1 | **0.95** | More Important |
| **LS306** | 0.652174 | 0.157895 | 0.7027 | 0.675 | More Important |
| **LS307** | 0.478261 | 0.052632 | **0.86486** | 0.775 | More Important |
| **LS3051** | 0.608696 | 0.105263 | 0.67568 | 0.65 | More Important |
| **LS3059** | 0.347826 | 0.631579 | **0.89189** | 0.8 | More Important |

FIG. 25 MATCHING ED VALUES BEFORE AND AFTER DE-SELECTION OF REPORTED LESS IMPORTANT TDS. EACH DIAGRAM HOLDS TWO IDENTICAL DATA SERIES



## REFERENCES

[1] H. L. Cloke, F. Pappenberger, J. P. Renaud, “Multi-Method Global Sensitivity Analysis (MMGSA) for modelling floodplain hydrological processes”, *J. Hydrol. Process.*, 22, 1660–1674, 2008.

[2] D. P. Durkee, E. A. Pohl, E. F. Mykytka, ”Sensitivity analysis of availability estimates to input data characterization using design of experiments”, *Int. J. Quality& Reliability Engineering*, 14, 311–317, 1998.

[3] R. I. Cukier, H. B. Levine, K. E, Shuler, “Nonlinear Sensitivity Analysis of Multiparameter Model Systems”, *J. Computational Physics*, 26, 1-42, 1978.

[4] E. Borgonovo, L. Peccati, “Global sensitivity analysis in inventory management”, *Int. J. Production Economics*, 108, 302–313, 2007.

[5] J. K. Ravalico, H. R. Maier, G. C. Dandy, J. P. Norton, B. F. W. Crokef, “A Comparison of Sensitivity Analysis Techniques for Complex Models”, *Proc. Int’l Cong. Modeling and Simulation*, 2533, 2539, Dec. 2005.

[6] A. Saltelli, “Sensitivity Analysis for Importance Assessment”, *J. Risk Analysis*, Vol. 22, No.3, 579-590, 2002.

[7] S. S. Isukapalli, “Uncertainty Analysis of Transport-Transformation Models”, *PhD thesis*, New Burnswick Rutgers, The State University of New Jersey, New Jersey, Jan. 1999.

[8] B. K. Hausmann, “Epistemic Sensitivity Analysis Based on the Concept of Entropy”, *Inter’l Symp. Sensitivity Analysis of Model Output (SAMO2001),* No3, pp. 53-57, June 2001.

[9] G. J. McRae, J. W. Rlden, J. H. Seinfeld, “Global-sensitivity-analysis--a-computational-implementation-of-the-Fourier-Amplitude-Sensitivity-Test-(FAST)”, *J. Computers & Chemical Engineering*, Vol. 6, No. 1, PP. 15-7.5.1982.

[10] Y. Jin, H. Yue, M. Brown, Y. Liang, D. B. Kell , “Improving data fitting of a signal transduction model by global sensitivity analysis”, *Proc. the 2007 American Control Conference*, July 2007.

[11] W. D. Kelton, “Simulation with Arena”, *McGraw Hill*, 2007

[12] Philver, “Philver Operator’s Manual”, *TecnoTeam*, 2000.

[13] S. Tavakoli, A. Mousavi, A. Komashie, “A Generic Framework for Real-Time Discrete Event Simulation (DES) Modelling”, *Proc. The 2008 Winter Simulation Conference*, Dec. 2008.

[14] R. Lowry, “Concepts and Applications of Inferential Statistics”, *Freebookcentre.net*, Available at: http://www.freebookcentre.net/special-books-download/Concepts-and-Applications-of-Inferential-Statistics-(Richard-Lowry).html

[15] C. Xu, G. Z. Gertner, “A general first-order global sensitivity analysis method”, *J. Reliability Engineering and System Safety*, 93, 1060-1071, 2008.

[16] R. A. Cropp, R. D. Braddock, “The New Morris Method: an efficient second-order screening method”, *J. Reliability Engineering and System Safety*, 78, 77-83, 2002.

[17] D. J. W. De Pauw, K. Steppe, B. De Baets, “Unravelling the output uncertainty of a tree water flow and storage model using several global sensitivity analysis methods”, *J. Biosystems Engineering*, 101, 87-99, 2008.

[18] R. D. Braddock, S. Yu. Schreider, “Application of the Morris algorithm for sensitivity analysis of the REALM model for the Goulburn irrigation system”, *J. Environmental Modeling and Assessment*, 11, 297-313, 2006.

[19] I.M. Sobol, “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”, *J. Mathematics and Computers in Simulation,* 55, 271–280, 2001.

[20] C. Annis, “Correlation”, *Statistical Engineering*, Available at: http://www.statisticalengineering.com/correlation.htm.

[21] A. Ambrosetti, A. Malchiodi, “Nonlinear analysis and semilinear elliptic problems”, *Cambridge University Press, Cambridge, UK*, 2007.

[22] B. Lallemand, G. Plessis, T. Tison, P. Level, "Neumann expansion for fuzzy finite element analysis", *Engineering Computations (Swansea, Wales*), vol. 16, no. 5, pp. 572-583, 1999,

[23] A. Buonomo, A. Lo Schiavo, "Nonlinear distortion analysis via perturbation method*", International Journal of Circuit Theory and Applications*, vol. 38, no. 5, pp. 515-526, 2010.

[24] D.G. Duffy, “Advanced engineering mathematics with MATLAB”, 2nd edn, *Chapman & Hall/CRC Press, Boca Raton, FL, USA*, 2003.

[25] G. Beylkin, C. Kurcz, & L. Monzón, "Fast algorithms for Helmholtz Green's functions", *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 464, no. 2100, pp. 3301-3326, 2008.

[26] C.P. Robert, G. Casella, “Monte Carlo statistical methods”, 2nd edn, *Springer,* New York, 2004.

[27] R.W. Shonkwiler, F. Mendivil, “Explorations in Monte Carlo methods”, *Springer, Dordrecht,* London, 2009.

[28] B. Krzykacz-Hausmann, "Epistemic Sensitivity Analysis Based on the Concept of Entropy", *Inter’l Symp. Sensitivity Analysis of Model Output (SAMO2001)*, pp. 53, 2001.

[29] B. Krzykacz-Hausmann, "An approximate sensitivity analysis of results from complex computer models in the presence of epistemic and aleatory uncertainties", *Reliability Engineering and System Safety*, vol. 91, no. 10-11, pp. 1210-1218, 2006.

[30] A. Saltelli, S. Tarantola, & K.P. Chan, "A quantitative model-independent method for global sensitivity analysis of model output", *Technometrics*, vol. 41, no. 1, pp. 39-56, 1999.

**S. Tavakoli** is a PhD Research Student in the School of Engineering and Design at Brunel University. He received a B.Sc. in Electronic Engineering from Amirkabir University of Technology in Tehran. His current research interests include real-time data acquisition and simulation for decision making tools. Siamak is a member of IEEE, Robotics and Automation Society, and Industrial Electronics Society. His email address is Siamak.Tavakoli@Brunel.ac.uk.

**A. Mousavi** is a Lecturer in the School of Engineering and Design at Brunel University. His research interests include simulation modelling and analysis, especially techniques to improve simulation analysis in practical applications. Ali.Mousavi@brunel.ac.uk and his Web address is www.brunel.ac.uk/~emstaam

**P. Broomhead** is a Production Technologist and Lecturer in Computing at Brunel University. His research interests include simulation modelling and analysis of manufacturing systems, parallel and practical applications in distributed computing. His email address is Peter.Broomhead@brunel.ac.uk.