

Fault Identification using Varying Weighted Sum Method in Proposed Multi Agent based Traffic Regulation and Automation System (MATRAS)

Rajdeep Chatterjee
School of Computer Engineering
KIIT University
Bhubaneswar, India

Deep Mukherjee
School of Electronics Engineering
KIIT University
Bhubaneswar, India

ABSTRACT

MATRAS is a newly proposed traffic monitoring and controlling model based on the Multi-Agent System. Indian Railways has a huge, robust and extremely complex network which is the largest in the World. But we are still practicing very traditional approaches in the Indian Railways, which requires frequent human interventions. As a result, any sort of mistake or deliberation could lead us to a fatal situation. The new MATRAS which is based on the Multi-Agent System is developed in such a way that it can easily be fitted into the current Indian Railway System. It not only automates the system and in turn eliminates human intervention to a certain level of the organizational hierarchy, but also takes the necessary actions voluntarily in order to run the system properly. Varying Weighted Sum method is also introduced for fault identification in the system.

Keywords

Multi agent, Traffic, Railways, Automation, Fault, Knowledge base.

1. INTRODUCTION

Indian Railways (IR) [1] was founded on the sixteenth of April 1853. It is the world's largest railway network comprising 115000 kms of track covers a route of 65000 km and has 7500 stations. As per the last report published in December 2012, it transported over 25 million passengers daily (over 9 billion on an annual basis). Around 1.5 million people work day and night in various departments. Indian Railways is divided into zones, which are further sub-divided into divisions. Each of the division has its own divisional headquarter. There are a total of 68 divisions. Every division is led by a Divisional Railway Manager (DRM) who reports to the General Manager (GM) of the zone. Divisional officers head all the departments namely signal, safety, mechanical, civil etc. and report to the DRM. One or two Additional Divisional Managers (ADRM) assists the DRM in the working of the division. In general a division has an average track length of about 1000 kms and staff strength of about 15000. We have observed that currently the Indian Railways is using manual and time consuming traditional approach to manage the system. As millions of people are working in various departments across the country throughout the year, even a small human mistake or error due to the absence of a thorough monitoring and maintenance lead to a huge disaster. It was already mentioned that millions of passengers avail of the railways on daily basis, so any kind of accident can incur not only a huge monetary loss but also take a toll on human lives. Human life is priceless and an asset for a country. So an intelligent and automated system which can take care of

these issues efficiently and effectively in a reliable way could be the replacement to the current traditional approaches of IR.

As this is a huge & complex system, here we are only focusing on the railway monitoring and controlling involving the Signal & Telecommunication department of the railways. Our proposed system will work independently in a division because each division has its own set of resources to run the system [1][8]. The Senior Divisional Signal & Telecommunication Engineer (SrDSTE) is heading the Signal & Telecommunication department of an entire division of the IR. There are also other personnel to assist the SrDSTE in his/her work. A Station is headed by a Station Master and managed through a Control Room. Each station has a Block Instrument which is actually responsible for signaling and traffic of trains. Currently signaling and other traffic related jobs are handled by station masters and his/her subordinates.

2. MULTIAGENT SYSTEM

Agents possess the capability of making decisions regarding their course of action in order to accomplish their design objectives [2][3]. Agent can be of various types: software, robotic, biological. Here we will discuss software agents. Our focus is to develop a Multi agent system which will intelligently handle its design goals. This kind of system is also known as intelligent agent. Intelligent agents are those which are capable of robust operation even in dynamic and unpredictable environments [4]. Multiple agents join together to solve a complex problem, while maintaining their autonomous behavior [5][6].

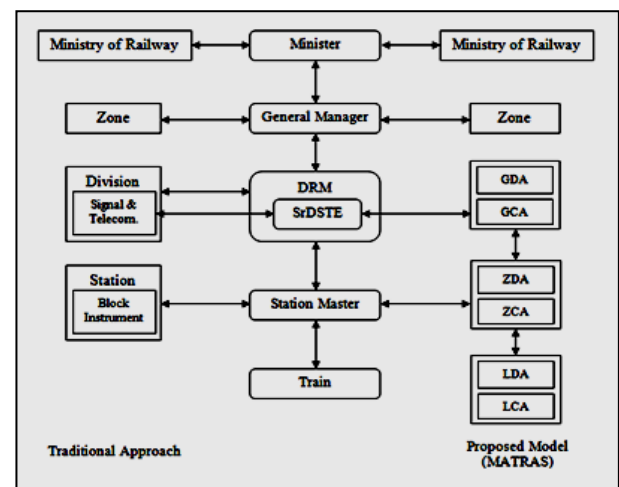


Fig 1: Indian Railways hierarchy

3. TRADITIONAL MODEL

Indian Railways uses block instrument which controls the coordinated movement of trains on the block sections. The

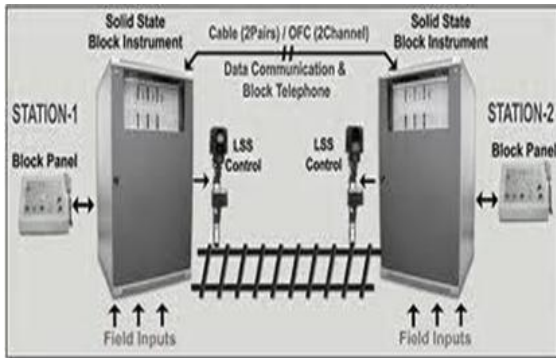


Fig 2: Schematic diagram of two Block Instruments

block instruments of the two stations or signal boxes at either end of the block section are electrically interconnected for this purpose. In physical token block areas, the block instrument is also responsible for allotting the tokens and is also popularly known as the token instrument. In working of manual block for double lines, the block instruments connect the two signal boxes have to be coordinated by the signal operators at the two signal boxes. Any signal box that allows a train into its block section first of all it must set the line clear indication on the block instrument which is also reflected as an indication in the block instrument of the signal box in the rear. The signal section until such a line clear indication is received. While a train is entering the block section, the signal box in front sets the block instrument to Train on line and while the train is passing the signal box in front, the block instrument indication is set to line closed. If there are no trains expected to be approaching or holding the block section, the normal status of the state is for the indications on both block instruments to be line closed. These signal operators are also coordinating with the nearby control room which is headed by the station master box in the rear allow any trains to proceed into the next block section until such a line clear indication is received. While a train is entering the block section, the signal box in front sets the block instrument to Train on line and while the train is passing the signal box in front, the block instrument indication is set to line closed. If there are no trains expected to be approaching or holding the block section, the normal status of the state is for the indications on both block instruments to be line closed. These signal operators are also coordinating with the nearby control room which is headed by the station master.

Table 1. Role of Agents

Agents	Role
Local Control Agent (LCA)	Placed along the railway tracks with a certain gap between two consecutive LCAs, sensing few pre-defined parameters and send input to their local DA.
Local Decision Agent (LDA)	Keeping track of parameters coming from local CAs and taking appropriate actions under the current circumstances at that moment.
Zonal Control Agent (ZCA)	Keeping track of the Signal & Telecommunication status around a station within a certain perimeter, managing trains passing by the station
Zonal Decision Agent (ZDA)	Generating reports and analyzing it at Station level, also interacting with human experts in order to resolve issues
Global Control Agent (GCA)	Managing the connected Zonal Agents and keep track of the Signal & Telecommunication status of the Division
Global Decision Agent (GDA)	Generating reports and analyze it at Divisional level, also interact with human expert in order to resolve issues
Train Agent (TA)	Keeping track of all the parameters of the train, initiating appropriate action when it is required, periodically (in 2 minutes) sending status report to the nearby ZCA
Signal Agent (SA)	Based on the LDA or ZCA, giving appropriate signal to route trains (guide the drivers of train)
Unmanned Railway Crossing Agent (URCA)	Depending on the signal, taking voluntarily steps by firstly alerting people who are crossing the railway tracks, and secondly shutting the crossing until the train has completely passed
Human Agent (HA)	Human expert to monitor and analyze the system

4. PROPOSED MODEL MATRAS

We propose MATRAS keeping the technicalities and the official hierarchy of the existing railway system in India so that it can easily be fitted into the current system. MATRAS is a Multi agent system based model which can automate the whole railway system and thus eliminate the human intervention completely [7][8]. But when we discussed this issue with a number of railway officials, they preferred to retain a certain degree of human interaction with the MATRAS at certain levels [9]. As a Multi agent system gives us the scope that an agent can also communicate with humans (i.e. human agent), we keep this feature in our proposed

MATRAS as well. Our MATRAS is divided into multiple agents of multiple types [10][11]. We can categorize them into three divisions- Global Agents (GA), Zonal Agents (ZA) and Local Agents (LA). These agents in turn are divided into Decision Agent (DA) and Control Agent (CA). However, our terminologies of global, zonal and local are not the same as those used by the IR in their organization. Additionally, there are another four types of agents, the first is Train Agent (TA), the second is Signal Agent (SA), the third is Unmanned Railway Crossing Agent (URCA) and fourth one is Human Agent (HA).

There can be three cases in our proposed system. The first case is termed as the regular case (RC), the second case is a partial non-regular case (PNC) and the last one is non-regular case. In RC the trains are managed locally by LDAs, whereas in PNC the traffic is guided by ZDAs and in NC it requires the help of GDA. The types and role of the various agents used in our proposed MATRAS are briefly described in the Table 1.

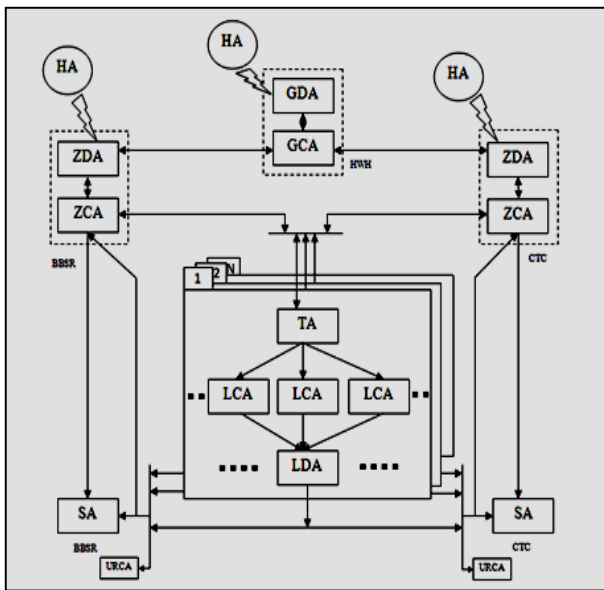


Fig 3: Proposed model MATRAS

LCAs are placed along the railway tracks at equal distances. A certain number of LCAs are connected to one LDA. As a Division oversees around 1000 kms of track, we assume and keep a few constants in the following way. One LDA covers 1 KM (1000 m) and if the distance between two LCAs is 10 m (say l) then there should be 100 LCAs under a single LDA. As we keep the coverage area of an LDA to be one kilometer, thus we eliminate the LDA-LDA handover. Also, it is obvious

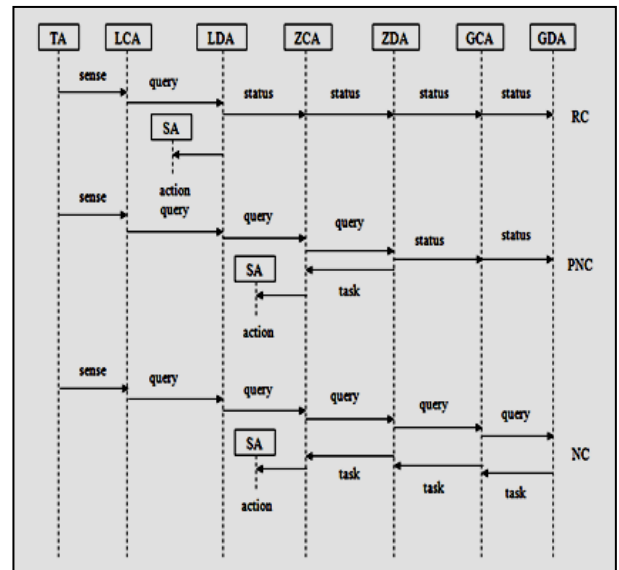


Fig 4: Agent sequence diagram

that the length of any type of train is within the range of 1000 m. Hence, this assumption reduces network congestions. Let the train speed be S , the actual train length be T and the number of LCAs used in monitoring a train be k . A train is moving with a speed of S , so in every second there is a displacement in its position. We call this displacement as Δd and if there is any change in speed, mostly acceleration, then we consider a Δe termed as displacement due to acceleration.

Let S speed/unit-time.

$$\Delta d = \text{displacement}, \quad m = \left\lceil \frac{\Delta d}{l} \right\rceil; \quad (1)$$

Now, one LDA needs k LCAs to track a train with the following condition:

$$(l * m) + T + (l * m) + \Delta e \leq k * l \equiv T + 2ml + \Delta e \leq k * l$$

$$\text{So } k = \left\lceil \frac{T + 2ml + \Delta e}{l} \right\rceil \quad (2)$$

Generally LCAs are in idle state and each of the LCAs keep tracking five parameters viz. vibration of line (vol), continuity of track (cot), speed of the train (sot), signal ahead (sah) and line alignment (lal) in every second. If the input of vol has crossed a threshold level, then only that LCA becomes active. As we need k LCAs to monitor a train, firstly k LCAs get inputs for its $5 * k$ parameters and send these to an LDA. Now, an LDA based on its functions and with the help of its local knowledge base decides the appropriate action in the current circumstances. The output of an LDA is either 1 or 0 or -1 and the LDA sends an input to SA. Subsequently, the SA sends the appropriate signal to put on i.e. either green (1) or yellow (0) or red (-1) light.

The driver of the train takes this signal change to its cognizance and takes the necessary steps to either stop or slow down or keep the speed as it is. Also the SA controls

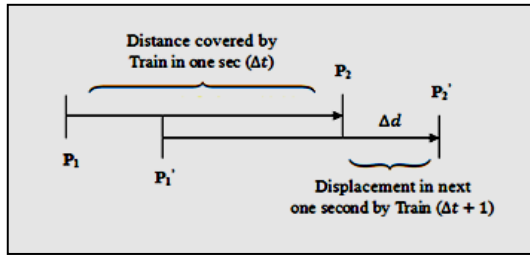


Fig 5: Displacement of a train in unit time

unmanned railway crossings through URCA. Additionally the TA takes care of its own controls and keeps track of three parameters like speed (S), alertness of driver (aod) and emergency (sos). TA sends the status of these parameters to two nearby ZCAs in every two minutes. The time frame is two minutes because IR uses a policy in which if the train system remains idle for two minutes then train is forced to stop. One ZCA is the station from where the train is coming and the second is the station where it is going to. Here we should explain that the ZAs are placed in stations which have certain criteria. These criteria include conditions like whether the station is situated in a capital city, how many trains stop there and how many passengers avail of the station on a daily basis. These ZAs are connected to one GA. A single GA is responsible for a Division of Indian Railways. Again there can be a case where some problem has arisen in a nearby location, but the LDA cannot recognize it. So it is the responsibility of the ZDA to decide and the ZCA sets the signals appropriately through the SA. In the cases like derailment or a heavy rush of passengers, the station master can decide the course and direct the ZDA to take steps to accommodate these anomalies in the system. The used parameters in MATRAS are discussed in Table 2.

Table 2. Parameters used in MATRAS

Parameters	Values	Meaning
Continuity of Track (<i>cot</i>)	0,1	0 : continuity for <i>l</i> kms 1 : break in continuity in <i>l</i> kms
Aspect of Signal ahead (<i>sah</i>)	-1,0,1	-1 : red (stop) 0 : yellow (caution) 1 : green (proceed)
Emergency (<i>sos</i>)	0,1	0 : normal 1 : emergency
Alertness of the Driver (<i>aod</i>)	0,1	0 : not alert 1 : alert
Line Alignment (<i>lal</i>)	[0,1]	0 : not ok (otherwise) 1 : ok (percentage with a threshold value TH_{lal})
Speed of Train (<i>sot</i>)	[0,1]	0 : not ok in current circumstances 1 : ok (percentage with a threshold value TH_{sot} in current circumstances)

Parameters	Values	Meaning
Vibration of Line (<i>vol</i>)	[0,1]	0 : not ok (otherwise) 1 : ok (percentage with a threshold value TH_{vol})

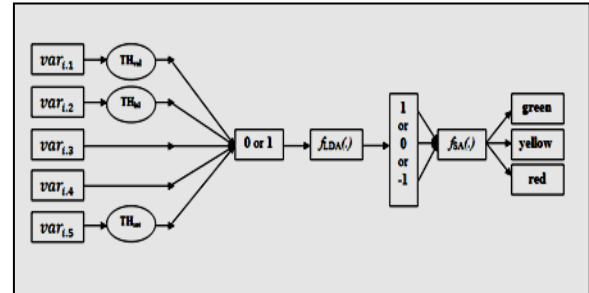


Fig 6: Block diagram of LCA-LDA-SA interaction

4.1 Fault Identification in MATRAS

Here the aim is to find out faults occur at real-time. We can identify faults by evaluating input parameters from the LCAs.

In fault identification, there are two objectives- (i) the type of faults and (ii) where are the faults? As per the Table 2, we know that LCA sends parameters in form of zeros and ones. Except signal ahead (*sah*), it has three types of values either -1 or 0 or -1. We are proposing a method known as Varying Weighted Sum (VWS) to identify faults rather say precedence of faults. Later we introduce two tables – sorted objects and sorted weights and based on a threshold value we are combining multiple objects which are most responsible for making faults. As in our case objects are nothing but LCAs and from them we can easily find out the locations where the problems are. After identifying the faults and their locations our proposed model MATRAS sends a message to a nearby station to dispatch maintenance team (with expertise on the relevant fault type) to that spot. In this case we have five parameters and we give five weights to each of the parameters. Vibration of line (*vol*) is depending on the continuity of track (*cot*) and line alignment (*lal*). So we keep *vol* to a constant weight and we distribute the remaining weights to the other four parameters.

In VSM, we have few assumptions regarding weight distribution. The sum of all weights for an object should be one.

$$\sum_j w_j = 1 \tag{3}$$

Then we have to choose a weight (w_i) which should be greater than the rest of the weights assigned to the other parameters.

$$\exists w_i \text{ such that } w_i > w_j, \quad i \neq j \tag{4}$$

Where $j=1, 2, 3 \dots n$ and $i \neq j$

Now we should uniformly distribute the remaining weights to the parameters (as per Eq. 5). If we cannot make a uniform weight distribution because of our choice of weights, then we have to assign a constant weight to one parameter using Eq. 6. Of course, the selection of this parameter should be done after a consultation with the domain experts.

$$w_{j1} = w_{j2} = w_{j3} = \dots = w_{jn-1} \quad (5)$$

$$\exists w_{jn} = 1 - (w_i + \sum_{j=1}^{n-1} w_j) \quad (6)$$

As per our VWS method, we have four different cases where cot, sot, lal and sah have the highest weight. We have shown the computation for one of the cases where cot has the highest weight 0.3 and we keep vol to a constant weight 0.1. Here we consider only ten LCAs which are sending formatted sensing data to a LDA (i.e. k is 10). We denoted this case by \overline{cot} . Similarly we have values 7.5, 7.6 and 7.6 for \overline{sot} , \overline{lal} and \overline{sah} respectively (results of Table 4). From these values we can conclude that cot is the faultiest and the values also tell us that which parameter is causing more faults for a given set of LCAs. This method is actually varying the weights for the parameters. It means we give high priority to each parameter for each of the cases. If we find that one of the parameter generates faults from most of the objects (LCAs), then the sum of weighted sum vector will give us less value (sum of Weighted Sum Vectors) than the other cases. From Table 4, we obtain that cot is generating more faults than the other parameters. Thus we achieve our first objective. Now we have to find out the LCAs which are causing more faulty parameters than others. Their locations are very important for us and the agent identification number gives us their (LCAs) exact positions.

In Table 5, we find sets of objects (\mathbb{E}) based on their common weighted sum value. These values are given in Table 6 for all four cases.

$$\text{for each } e_i \in \mathbb{E} \exists W_i \text{ iff } W_i \leq th_f \quad (7)$$

$$\text{if } \mathbb{F} \cap e_i = \emptyset \text{ then } \mathbb{F} = \mathbb{F} \cup \{e_i\} \text{ for } W_i < W_j \quad (8)$$

where $j = 1, 2, 3 \dots n$ and $i \neq j$

Now the next part is to identify the location. This can be done through identifying the LCAs that generate more faulty values. We are sorting the weighted sum vector in ascending order and also group the objects (LCA) which have the same weighted sum (w_i) for each of the cases.

Initially we have to start with a single element object-set having smallest weight and combine it with an empty set \mathbb{F} (Eq. 9). This process will continue for all the single element object-sets having weight less than the prescribed threshold (here we take 0.5 as our threshold for faulty object) (Eq. 10, 11). Therefore we continue with two elements object-sets having smaller weights and combine them with the fault set \mathbb{F} but these new objects should not present in \mathbb{F} previously (Eq. 8, 12). Finally we have a set \mathbb{F} containing the objects (LCAs) causing faults beyond the specified tolerance level (Eq. 13).

$$\mathbb{F} = \{\emptyset\} \quad (9)$$

$$\mathbb{F} = \mathbb{F} \cup \{7\} \quad (10)$$

$$\mathbb{F} = \mathbb{F} \cup \{3\} \quad (11)$$

$$\mathbb{F} = \mathbb{F} \cup \{5, 9\} \quad (12)$$

$$\mathbb{F} = \{7, 3, 5, 9\} \quad (13)$$

Now if we analyze fault set \mathbb{F} , we find that seventh one is the faultiest because the sah value in the seventh LCA is -1 i.e. red (stop). Again third, fifth and ninth LCAs have faults in cot and sot. As each of the LCA has a unique Identification number. So we can easily locate the positions of the faulty LCAs and send expert team to fix the problem. Here we only consider ten LCAs in our example, but in real scenario the value of k will come from Eq. 2. We have discussed the method of fault identification in our proposed model MATRAS and hence the Varying Weighted Sum Method gives us not only the precedence of faulty parameters but also the locations of the LCAs which are generating the faults.

5. CONCLUSION AND FUTURE WORK

In this paper, we have restricted ourselves to only discuss a brief work flow model of the proposed architecture MATRAS along with we have explained a new method VWS to identify type of faults and its occurrence positions. If our proposed robust intelligent multi-agent based model MATRAS is implemented in the current Indian Railways, it will change the existing system into a fully automated or semi-automated system which will reduce human interference to as little as possible. At present the length of 115000 KMs of Indian Railways with require at least 115 * 105 LCAs and 11500 LDAs, by our assumptions (100 LCAs per Km). Therefore realistically, the fully implementation may not be an economically feasible one. But if we think about those routes which are more vulnerable to natural damage or human sabotage, then MATRAS could be a better alternative to the current system. Currently we are working on the ontology of multiple agents used in MATRAS and trying to develop an efficient communication network to support MATRAS. In future, we will address those issues in a rigorous way.

Though our proposed model MATRAS is developed keeping the Indian Railways in our mind, but with a little modification in the MATRAS, it can be used for Indian Coastal security as well. As an instance, we can then enable the LCAs to keep track of SONAR signatures, which should be equipped with advanced imagery capabilities and we would need no significant changes in the rest of the model.

6. ACKNOWLEDGMENTS

Our special thanks to Mr. Debasish Kumar Mishra, Junior Engineer, Department of Signal & Telecommunication, East Coast Railway Division (ECoR), South Eastern Railway (Indian Railways).

7. REFERENCES

- [1] Indian Railways, ECoR (Indian Railways), Bhubaneswar, India.
- [2] Michael Wooldridge: An Introduction to Multi agent Systems, Department of Computer Science, University of Liverpool, UK, JOHN WILEY & SONS, LTD.
- [3] Gerhard Weiss: Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence, The MIT Press Cambridge, London, England.
- [4] Saroj Kausik: Artificial Intelligence, Department of Computer Science, Indian Institute of Technology Delhi, CENGAGE Learning India PVT. LTD. 2012.
- [5] Shibakali Gupta, Rajdeep Chatterjee, Sripati Mukhopadhyay; Transaction and Performance analysis of Clinical Diagnosis System: A Novel Scheme using Multi Agent System, IJCA Proceedings on International Conference on Recent Trends in Information Technology

- and Computer Science 2012 ICRTITCS(7):11-15, February 2013.
- [6] Shibakali Gupta, S Pujari: A multi-agent system (MAS) based scheme for health care and medical diagnosis system, Intelligent Agent & Multi-Agent Systems, IAMA 2009. International Conference on pp. 1-3, IEEE proceedings.
- [7] Xiaojun Shen, Shirmohammadi, S. Desmarais, C. Georganas and Nicolas D: Multi-agent System Architecture for Collaborative E-Commerce, Electrical and Computer Engineering, CCECE '06, pp 284-286, May 2006.
- [8] P. G. Balaji, D. Srinivasan: Multi-Agent System in Urban Traffic Signal Control, Computational Intelligence Magazine, IEEE, vol. 5, issue 4, pp. 43–51, 2010.
- [9] Wei Wenjun , Fan Duowang: Study on Multi-agent Railway Signal Control System, Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2011 IEEE International Conference, pp. 235–238, 2011.
- [10] F. P. Maturana, P. Tichy, P. Slechta, F. Discenzo, R. J. Staron, and K. Hall: Distributed multi-agent architecture for Automation Systems, Expert Systems with Applications, vol. 26, no. 1, pp. 49–56, 2004.
- [11] M. C. Choy, D. Srinivasan, R. L. Cheu: Cooperative, Hybrid Agent Architecture for Real-Time Traffic Signal Control, IEEE Trans. Syst., Man, Cybern. A, vol. 33, no. 5, pp. 597–607, 2003.
- [12] History of Indian Railway - http://en.wikipedia.org/wiki/Indian_Railways

Table 3. Weighted Sum Vector

LCA(s)	vol	sot	lal	cot	sah	Weighted Sum Vector (WSV)
1	1*0.1	1*0.2	1*0.2	1*0.3	1*0.2	1
2	1*0.1	1*0.2	0*0.2	1*0.3	1*0.2	0.8
3	1*0.1	1*0.2	0*0.2	0*0.3	1*0.2	0.5
4	1*0.1	1*0.2	1*0.2	0*0.3	1*0.2	0.7
5	1*0.1	0*0.2	1*0.2	0*0.3	1*0.2	0.5
6	1*0.1	0*0.2	1*0.2	1*0.3	1*0.2	0.8
7	1*0.1	1*0.2	1*0.2	1*0.3	-1*0.2	0.6
8	1*0.1	1*0.2	1*0.2	1*0.3	1*0.2	1
9	1*0.1	0*0.2	1*0.2	0*0.3	1*0.2	0.5
10	1*0.1	1*0.2	1*0.2	1*0.3	1*0.2	1
cot					sum(WSV)	7.4

Table 4. Varying Weighted Sum Method

Case	Parameter with highest weight	Sum of Weighted Sum Vector
1	cot	7.4
2	sot	7.5
3	lal	7.6
4	sah	7.6

Table 5. Sorted Object-Set

Faulty Parameters	Sorted Object Sets (where faulty threshold $th_f=0.5$)
cot	{3,5,9},{7},{4},{2,6},{1,8,10}

\overline{cot}	{5,9},{3,7},{6},{2,4},{1,8,10}
\overline{lal}	{3},{5,7,9},{2},{4,6},{1,8,10}
\overline{sah}	{7},{3,5,9},{2,4,6},{1,8,10}

Table 6. Sorted Weight-Set

Faulty Parameters	Sorted Weights (where faulty threshold $th_f=0.5$)
\overline{cot}	0.5,0.6,0.7,0.8,1
\overline{sot}	0.5,0.6,0.7,0.8,1
\overline{lal}	0.5,0.6,0.7,0.8,1
\overline{sah}	0.4 ,0.6,0.8,1