Robust Fault Tolerant Rail Door State Monitoring Systems: Applying the Brooks-Iyengar Sensing Algorithm to Transportation Applications

Buke Ao
BYD Company Limited, People's Republic of China

In a moving train, the acceleration and deceleration adversely effects the sensor systems, which may induce errors into the system. Todays technology does not guarantee success and safety in all situations. There are two important situations that are critical for the safety of passengers when embarking and disembarking. Distributed sensing networks needed to control the train doors require a fusion of the sensor inputs to provide accurate automatic opening and closing with minimum traction. Brooks-Iyengar Distributed Sensing algorithm can be used to provide a fault tolerant automatic sensing platform for closing doors based on the following scenario. An automatic sensor network can be installed in the motor circuit to collect current data through a wireless protocol. The data can be transmitted by cellular communication to servers, where the Brooks-Iyengar distributed sensing algorithm can be applied to identify and categorize the data signals to safely and automatically open and close the doors. This paper describes the performance evaluation of the signal output of Brooks-Iyengar algorithm in this application. Based upon the performance results, the Brooks-Iyengar Algorithm provides the best robust algorithm for implementation under faulty sensor conditions, such as those encountered in real-world transportation applications.

Keywords: Sensor fusion, Distributed computing, Fault tolerance

1. INTRODUCTION

The Brooks-Iyengar hybrid algorithm (Brooks and Iyengar, 1996) for distributed control in the presence of noisy data combines Byzantine agreement with sensor fusion. It bridges the gap between sensor fusion and Byzantine fault tolerance (Ilyas, Mahgoub, and Kelly, 2004). The algorithm is fault-tolerant and distributed. It could also be used as a sensor fusion method. The precision and accuracy bound of this algorithm have been proved in 2016 (Ao, Wang, Yu, Brooks, and Iyengar, 2016). This seminal algorithm unified these disparate fields for the first time. Essentially, it combines Dolevs algorithm for approximate agreement (Dolev, 1981)(Lamport, Shostak, and Pease, 1982)(Dolev, Lynch, Pinter, Stark, and Weihl, 1986)(Lamport et al., 1982) with Mahaney and Schneiders fast convergence algorithm (FCA) (Mahaney and Schneider, 1985). Researchers have also extended the original Byzantine agreement to Byzantine Vector Consensus (BVC) (Vaidya and Garg, 2013)(Mendes and Herlihy, 2013). The algorithm assumes N Processing Elements (PEs), τ of which are faulty and can behave maliciously. It takes as input either real values with inherent inaccuracy or noise (which can be unknown), or a real value with a priori defined uncertainty, or an interval. The output of the algorithm is a real value with an explicitly specified accuracy. The algorithm runs in O(NlogN) where N is the number of PEs. It is possible to modify this algorithm to correspond to Crusaders Convergence Algorithm (CCA) (Mahaney and Schneider, 1985), however, the bandwidth requirement will also increase. The algorithm has applications in distributed control, software reliability, and high-performance computing (Mahaney and Schneider, 1985), and can be used to find the "fused" measurement of the weighted average of the midpoints of regions (Sahni and Xu, 2005).

The Brooks-Iyengar algorithm has been used in a variety of redundancy applications (Kumar, 2012) (Ao et al., 2016)(Brooks and Iyengar, 1996) , including a program demonstration through the US Defense Advanced Research Projects Agency (DARPA) with BBN using the Sensor Information Technology for the War Fighter (SensIT) program. SensIT program develop software for networks of distributed micro sensors, specifically in collaborative signal and information processing and fusion of data. More specifically, information was received from sensors in reconnaissance, surveillance, tracking, and targeting for battlefield operations. This work was an essential precursor to the Emergent Sensor Plexus MURI from Penn State Universitys Applied Research Laboratory (PSU/ARL), which incorporated the Brooks-Iyengar Algorithm to extend SensITs advances to create practical and survivable sensor network applications.

The Brooks-Iyengar Algorithm has also been extended into modern-day LINUX and Android operating systems. In these applications, the algorithm combines data to provide fault-tolerant data fusion which is used by 99% of the worlds top supercomputers, 79% of all smartphones worldwide, and 100% of users accessing the Internet, to provide seamless operations and service.

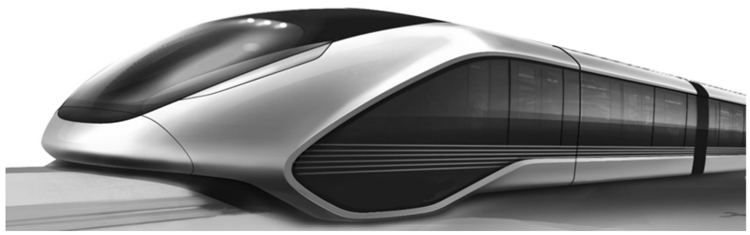

Figure 1. Apply Brooks-Iyengar Algorithm in The Train

2. FOUNDATIONAL FUSION RESULTS IN TRANSPORTATION APPLICATIONS

In this section of the paper, we present the theoretical applications of fusion, then demonstrate a detailed application and simulation of the Brooks-Iyengar Algorithm in solving the safety challenges of closing and opening the doors aboard moving trains. Accurately detect the state of train door such as the variable to be measured is critical to the application.

2.1 Theory

In a fusion system, we want to fuse different interval values from sensors where τ represents the number of faulty sensors. Suppose we have N sensors, which measure the variable value of $[l_1, h_1], ..., [l_N, h_N]$.

Notations:

— v_T: The ground truth value
— v: The output value of Brooks-Iyengar algorithm
— I_{BY}: The output interval of the Brooks-Iyengar algorithm
— $a_{N-2\tau}$: The left endpoint of the region where $N-2\tau$ non-faulty intervals overlap
— $b_{N-2\tau}$: The right endpoint of the region where $N-2\tau$ non-faulty intervals overlap
— g : A set of $N-\tau$ valid measurements
— f : A set of τ faulty measurements
— G : The set of all possible valid measurements, so $g \in G$
— F : The set of all possible faulty measurements, so $f \in F$

We build three theorems to describe the performance of Brooks-Iyengar algorithm in a fusion system. Theorem 1 and Theorem 2 identify the accuracy feature of Brooks-Iyengar algorithms. Theorem 3 is the comparison of Brooks-Iyengar algorithm with other related algorithms.

Theorem 1. *In a system of N sensors which of them are faulty, $v_T \in I_{BY} \subseteq [a_{N-2\tau}, b_{N-2\tau}]$ and*

$$|v - v_T| \leq |I_{BY}| \leq \min_{\tau+1}\{|v| : v \in g\}$$

Where, $|I_{BY}|$ is the length of the output interval, v and I_{BY} is the output value and interval of Brooks-Iyengar algorithm.

Proof. We assume all non-faulty intervals intersect on a common region, and we define it as I_{opt}. Since there are at least $N - \tau$ non-faulty intervals, so the weight of I_{opt} is at least $N - \tau$, and we have $I_{opt} \subseteq I_{BY}$, so $v_T \in I_{opt} \subseteq I_{BY}$. Since there are at most τ faulty intervals and the threshold of Brooks-Iyengar algorithm is $N - \tau$, so only regions that are formed by non-faulty intervals with weight equal or larger than $N - 2\tau$ intersect with τ faulty intervals could be subset of I_{BY}. So $[a_{N-2\tau}, b_{N-2\tau}]$ is the upper bound of I_{BY}, and $I_{BY} \subseteq [a_{N-2\tau}, b_{N-2\tau}]$. The equation, $|I_{BY}| \leq \min_{\tau+1}\{|v| : v \in g\}$ has already been proved by Theorem 2 (Marzullo, 1990) as outlined by Marzullo, and as illustrated below.

Theorem 2. *The interval $[a_{N-\tau}, b_{N-\tau}]$ is smallest interval that is guaranteed to contain the true value (Marzullo, 1990).*

Proof. the proof is in Algorithm 1 (Marzullo, 1990)p, as follows.

Therefore, the algorithm below provides the fusion value v of the Brooks-Iyengar algorithm, given the set of all possible valid measurements g and all possible faulty measurements f:

$v = BY(g, f)$: The fusion value v of Brooks-Iyengar algorithm given g and f p

Theorem 3. *In a system of N sensors in which τ of them are faulty, then we have two sets G and F, $v = BY(g, f)$, where $g \in G$ and $f \in F$, then we have:*

$$\max_{g,f} |v - v_T| \leq \max_{g,f} |v_{ABA} - v_T|$$

$$\max_{g,f} |v - v_T| \leq \max_{g,f} |v_{BVC} - v_T|$$

$$\max_{g,f} |v - v_T| \leq \max_{g,f} |v_{avg} - v_T|$$

Where v_{ABA}, v_{BVC} and v_{avg} are the results of Approximate Byzantine Agreement (ABA), Byzantine Vector Consensus (BVC) (Vaidya and Garg, 2013) and naive average algorithm, here we calculate the midpoints of the intervals as inputs of the two algorithms.

Proof. From Theorem 1 we know that, $v_T \in I_{BY} \subseteq [a_{N-2\tau}, b_{N-2\tau}]$, and from Proposition 4.1 in (Ao et al., 2016), we know there exists $g \in G$ and $f \in F$ such that $v_{ABA} \notin [a_{N-2\tau}, b_{N-2\tau}]$, which means $\max_{g,f} |v_{ABA} - v_T| \geq \max(|b_{N-2\tau} - v_T|, |a_{N-2\tau} - v_T|) \geq \max_{g,f} |v - v_T|$. So we have $\max_{g,f} |v - v_T| \leq \max_{g,f} |v_{ABA} - v_T|$. Similarly, from Proposition 4.2 in (Ao et al., 2016), there exists g,f such that $v_{BVC} \notin [a_{N-2}, b_{N-2}]$, and we could easily prove $\max_{g,f} |v - v_T| \leq \max_{g,f} |v_{BVC} - v_T|$. Equation (2) of (Ao et al., 2016) shows that the naive average could not tolerant fault so $\max_{g,f} |v_{avg} - v_T| = \infty$ and then $\max_{g,f} |v - v_T| \leq \max_{g,f} |v_{avg} - v_T|$.

3. IMPLEMENTATION

In this section, we show a cluster of sensors being used to detect the state of opening and closing of the trains doors. The situation could involve more than one hundred sensors, where as many as one third of the sensors could be faulty yet using the Brooks-Iyengar Algorithm, we can still maintain accurate results.

Accurately detecting the state of the train's door is very crucial for ensuring people's safety. However, traditional scheme uses a single sensor to detect the current or some other variables of the train's door system is not accurate when the sensor behaves fault. By leveraging Brooks-Iyengar algorithm, we could use multiple sensors to measure variables (current, etc.) robustly and accurately.

For example, let us use 4 sensors, where one of them is faulty. Each sensor's output is $[v_i - \sigma, v_i + \sigma], 1 \leq i \leq 4$, where the uncertainty bound for non-faulty and faulty sensors is identical for simplicity, so the ground truth v_T could be any point within the bound for non-faulty sensors but the output of fault sensor may not contain the ground truth v_T. The simulation is as follows.

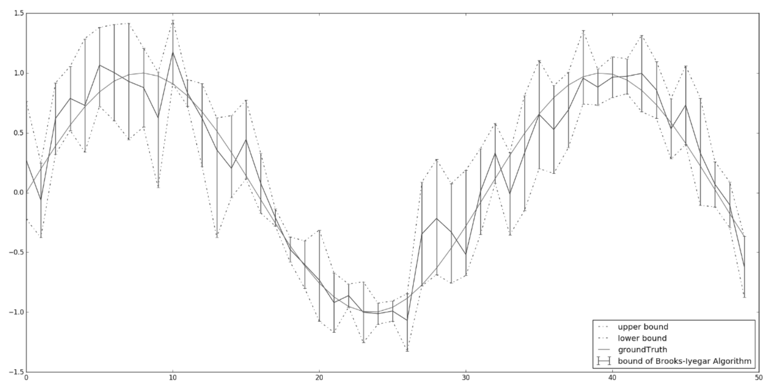

Figure 1. The Output of Brooks-Iyengar Algorithm

Figure 1 gives an example of the output of Brooks-Iyengar algorithm. The green line is ground truth that has been pre-defined. We have 4 sensors to measure the variable, where one of them is faulty. The blue line is the fused value of Brooks-Iyengar Algorithm and we could also find that the ground truth lies in the upper bound and lower bound of Brooks-Iyengar Algorithm.

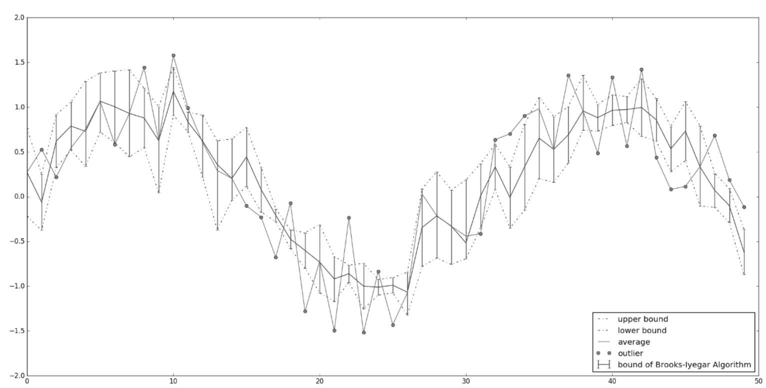

Figure 2. Comparison of Brooks-Iyengar Algorithm and Average

Figure 2 shows a comparison between Brooks-Iyengar algorithm and the naive average algorithm. We assume that the distance between ground truth and faulty sensor reading is Gaussian distribution. Then we run two algorithms in the same condition. From the simulation results, we could find that the bound of Brooks-Iyengar algorithm always contains the ground truth, while the output of naive average sometimes is far from the ground truth. Since the bound of Brooks-Iyengar algorithm smallest bound to contain the ground truth, the green line that is not in the bound must be faulty outputs, which are denoted by the red points.

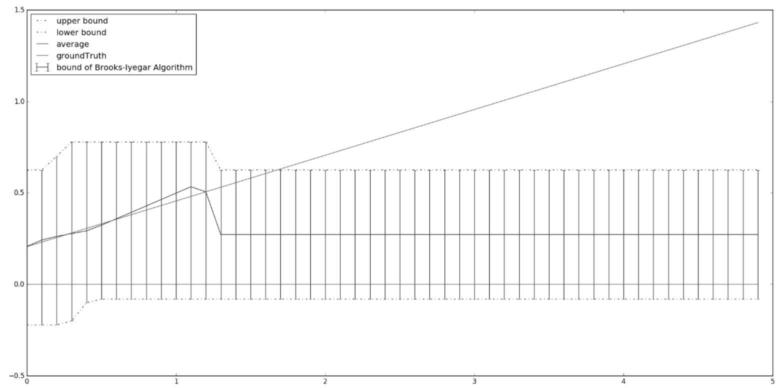

Figure 3. The Output of Two Algorithms When Faulty Values Increase

Figure 3 considers the output of Brooks-Iyengar algorithm and naive average, when the faulty sensor reading keeps increasing. We could find that when the faulty sensor readings increase, the output of naive average algorithm becomes larger and larger, which indicates its sensitive to bad sensor readings. Both the output value and bound of Brooks-Iyengar algorithm are robust when the faulty sensor reading becomes larger, which shows that the Brooks-Iyengar algorithm is robust to outlier or faulty sensor readings.

4. CONCLUSION

In this train example, with acceleration and deceleration adversely affecting the sensor systems, the authors induced an error in one of the sensors to examine the effectiveness of the Brooks-Iyengar Algorithm in these applications. Since todays technology does not guarantee success and safety in all situations, the Brooks-Iyengar algorithm can significantly improve the fault tolerance of these systems, providing a greater margin of safety for operations. In doing so, there are two important situations that are critical for the safety of passengers when embarking and disembarking. Distributed sensing networks needed to control the train doors require a fusion of the sensor inputs to provide accurate automatic opening and closing with minimum traction. An automatic sensor network can be installed in the motor circuit to collect current data through a wireless protocol. The data can be transmitted by cellular communication to servers, where the Brooks-Iyengar distributed sensing algorithm can be applied to identify and categorize the data signals to safely and automatically open and close the doors.

In this paper, the authors described and demonstrated the performance evaluation of the signal output of Brooks-Iyengar algorithm in this application. Based upon the performance results, the Brooks-Iyengar Algorithm provides the best robust algorithm for implementation under faulty sensor conditions, such as those encountered in real-world transportation applications.

References

Ao, B., Wang, Y., Yu, L., Brooks, R. R., and Iyengar, S. S. 2016. On precision bound of distributed fault-tolerant sensor fusion algorithms. *ACM Comput. Surv. 49,* 1 (May), 5:1–5:23.

Brooks, R. R. and Iyengar, S. S. 1996. Robust distributed computing and sensing algorithm. *Computer 29,* 6 (June), 53–60.

Dolev, D. 1981. The byzantine generals strike again. Tech. rep., Stanford, CA, USA.

Dolev, D., Lynch, N. A., Pinter, S. S., Stark, E. W., and Weihl, W. E. 1986. Reaching approximate agreement in the presence of faults. *J. ACM 33,* 3 (May), 499–516.

Ilyas, M., Mahgoub, I., and Kelly, L. 2004. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems.* CRC Press, Inc., Boca Raton, FL, USA.

Kumar, V. 2012. Computational and compressed sensing optimizations for information processing in sensor network. *IJNGC 3*.

Lamport, L., Shostak, R., and Pease, M. 1982. The byzantine generals problem. *ACM Trans. Program. Lang. Syst. 4,* 3 (July), 382–401.

Mahaney, S. R. and Schneider, F. B. 1985. Inexact agreement: Accuracy, precision, and graceful degradation. In *Proceedings of the Fourth Annual ACM Symposium on Principles of Distributed Computing*. PODC '85. ACM, New York, NY, USA, 237–249.

Marzullo, K. 1990. Tolerating failures of continuous-valued sensors. *ACM Trans. Comput. Syst. 8,* 4 (Nov.), 284–304.

Mendes, H. and Herlihy, M. 2013. Multidimensional approximate agreement in byzantine asynchronous systems. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*. STOC '13. ACM, New York, NY, USA, 391–400.

Sahni, S. and Xu, X. 2005. Algorithms for wireless sensor networks. *IJDSN 1*, 35–56.

Vaidya, N. H. and Garg, V. K. 2013. Byzantine vector consensus in complete graphs. In *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*. PODC '13. ACM, New York, NY, USA, 65–73.

Mr. Buke Ao received his master degree from School of Information and Communication Engineering, Beijing University of Post and Telecommunication (BUPT), China. His research interests include machine learning, mobile sensing, distributed computing, etc. Email: aobuke@hotmail.com