Quadruple Adaptive Redundancy with Fault Detection Estimator

Dohyeung Kim[1] and Richard Voyles[2]

***Abstract*— As a result of advances in technology, systems have grown more and more complex, leading to greater opportunities for failure. System fault has become an increasingly significant threat to the reliability and fault tolerance of automation systems. Redundancy of components within the system is one popular method for enhancing fault tolerance. One of the simple and effective methods for fault tolerance that has stood the test of time is Triple Modular Redundancy, which provides not one redundant copy of a critical system component, but two extra copies for a total of three. Ironically, providing three copies of a sensor, for example, only provides tolerance to one failure. It takes four copies to provide resilience to a double-point failure, but this can be expensive in terms of the cost of sensors and installation, wiring, and interface circuitry. We propose Quadruple Adaptive Redundancy, a new method that adds software-based estimation techniques, rather than additional hardware components, to achieve higher levels of robustness with virtually no incremental cost. In this paper, the performance of Quadruple Adaptive Redundancy is verified through computer simulations and compared to Triple Modular Redundancy with one and two induced sensor failures.**

I. INTRODUCTION

Control systems are growing in complexity and ubiquity. People see more and more applications in all areas of life such as factories, cars, unmanned aerial vehicles (UAVs), etc. New applications often demand new sensors and software techniques and, accordingly, more hardware parts leads to more things to design and more software parts to operate and calibrate the hardware. This increased complexity leads to more potential failures and more intricacy in understanding system state. As components multiply, the need for self-diagnostics increases. The increasing sophistication of sensors also demands additional self-diagnostics to discern error states. As a result, errors can occur in the hardware or the software, which further multiplies the need for built-in fault tolerance. Hence, designers need tools to help design these increasingly complex systems. The Collaborative Robotics Lab has been researching real-time system design tools to help with managing complex time-deterministic tasks. An example is Post-Based Objects [CHIM] encapsulated in our real-time operating system, PBO/RT [1][2], which simplifies the creation of periodic software modules and state-based application tasks. In more recent work, our ReFrESH architecture for building self-adaptive software systems encapsulates self-adaptivity and fault tolerance [3] is a tool to simplify the design of complex systems.

*This work was supported, in part, by the NSF Center for Robots and Sensors for the Human Well-Being

[1]D. Kim is with the Aerospace Engineering Dept., Purdue University, West Lafayette, IN 47907, USA kim1774@purdue.edu

[2]R. M. Voyles is with the Polytechnic Institute, Purdue University, West Lafayette, IN 47907, USA rvoyles@purdue.edu

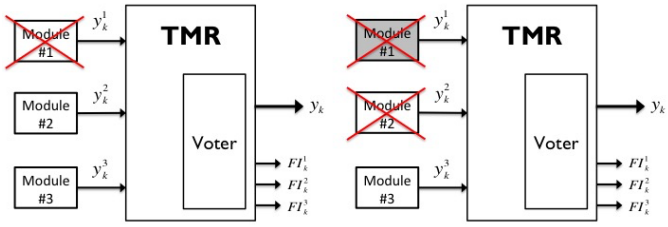

Fig. 1. (a) TMR when module1 is failure (b) TMR when module1&2 are failure

Triple Modular Redundancy (TMR) is a time-tested design tool to simplify redundant sensors, actuators and subsystems [4]. TMR is used to give systems the capability to maintain error-free operation in the presence of unexpected faults. Fig. 1 illustrates the basic concept and approach to TMR. It uses three functionally equivalent modules to provide redundant information coupled with a voting process to produce one reliable output, even in the presence of single-point failures. Through comparison of the three modules, simple computations can determine if all values agree and, if not, which module is at "fault." As illustrated in Fig. 1(a), if one value diverges from the other two, the median value, for example, accurately represents a good estimate of the desired value and the "voter" can flag the faulty module. In this way, the system is reliable in the case of a single point of failure with little reliance on the expertise of the designer. In Fig.1(b), if another module, however, fails, the median value is inconclusive with only two values and the voter cannot determine which module is broken. Therefore, TMR has the strength that it is simple to implement and adds significant robustness with minimal expertise, but it is robust only if one of the modules fails. Here, y_k^i is the measurement of i-th module at time step k, FI_k^i is the fault indicator of i-th module at time step k and y_k is the output of the redundant system at time step k.

To increase fault tolerance to allow double-point failures under such a scheme, it is necessary to employ four copies of redundant modules in a "fail operational/fail safe" configuration. An example is the NASA Space Shuttle computer system, which uses four computers for redundancy. We propose a novel method capable of handling double-point failures in "most circumstances." In order to increase the system for second-point failure, we propose Quadruple Adaptive Redundancy (QAR) in this paper. For two-point failures, installing a fourth sensor requires additional cost and increases the hardware size and complexity. Adding a

virtual sensor is a low cost way to add redundancy. Therefore, we establish an estimator that plays the role of the virtual sensor. Depending on the reliability of the estimator, QAR can estimate the proper value through a single-point failure and predict if it is reliable enough to tolerate a double-point failure, hence its adaptivity.

The rest of this paper is organized as follows: In Section II the related work in TMR is introduced. Section III provides the mathematical model of system and derive the QAR algorithm. Two scenarios in terms of the module fault are presented to demonstrate the proposed method through the experiment results in Section IV. Conclusions are given in Section V.

II. Related Work in TMR

Various TMR schemes have been widely used in critical system that can result in loss of life and serious disaster. Many researches have been done to increase the efficiency or performance aspect of the TMR. The selective fault tolerance method was applied to the TMR technique in order to reduce the size and energy consumption of the system[5][6]. The method can decrease the overhead through only using the critical set which is defined by the designer of system. Synchronous-feedback method was presented to improve the hardware resources problem of closed-loop TMR structure[7]. G. Latif-Shabgahi, et al. proposed the method to combine the self-diagnosis elements and voting algorithms to increase the reliability[8]. Also the adaptive voting algorithm was presented to improve the confidence of system[9]. This technique uses the history data of each module to access the reliability of module.

However, those TMR methods cannot handle the case of two modules fault without adding more module. Even if they can do it, the accuracy of the output depends on the sensor noise. To rise the robust of TMR, we can put one more module to system. However, it requires not only the increase of hardware size but also the additional cost. Therefore, we propose the Quadruple Adaptive Redundancy (QAR) method in order to increase the robustness. It has an estimator like a virtual sensor. TMR robust when one module fails but QAR robust when two sensors are broken without adding significant cost increase. The QAR is dual point failure. Also the QAR can produce the accurate estimate since the estimator includes the system identification and state estimation methods.

III. Quadruple Adaptive Redundancy

In this section, we derive mathematical models to describe the dynamics of the system and detail inside compositions that consist of the system identifier and state estimator scheme.

A. System Model

The equations of the system dynamic are following

$$x_k = A_k x_{k-1} + B_k u_{k-1} + w_{k-1} \tag{1}$$

$$y_k = C_k x_k + D_k u_k + v_k \tag{2}$$

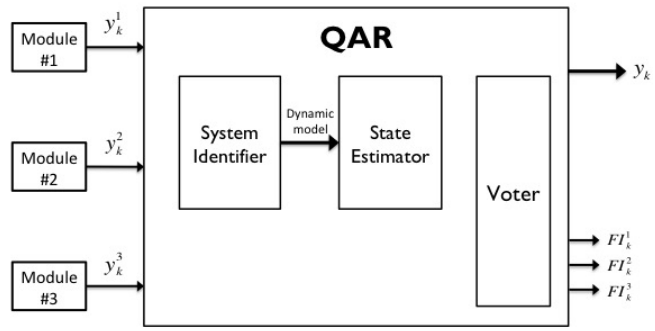

Fig. 2. Structure of Quadruple Adaptive Redundancy

where $x_k \in \mathbf{R}^n$, $u_k \in \mathbf{R}^p$, and $y_k \in \mathbf{R}^m$ are the state, input and output of system at time step k, respectively. A_k, B_k, C_k and D_k are the system dynamic matrices dimensions at time step k. $w_k \in \mathbf{R}^n$ and $v_k \in \mathbf{R}^m$ denotes the process noise and measurement noise at time step k. They are assumed as white Gaussian noise with zero means and $E\{w_k w_k^T\} = Q_k$ and $E\{v_k v_k^T\} = R_k$ covariances, $N(0, Q_k)$ and $N(0, R_k)$, respectively. $N(m, \Sigma)$ denotes the Gaussian distribution with mean m and covariance Σ. k denotes the discrete-time index.

B. System Identification (SI)

This section describes the method for determining the unknown dynamic of system. The SI block finds out the proper system dynamics with the maximum likelihood function and provides the model information and the reliability of the model information to the state estimator block. The maximum likelihood method is very sensitive to the amount of measurements. Therefore, if it cannot get enough data to identify the system model, the inaccurate outcome of the SI block can affect the performance of the QAR. Depending on the reliability of the SI, the QAR decides whether or not the state estimator block is used as the fourth sensor. For the state and parameter estimation, the Kalman filter algorithm[10] is used.

1) Filtering algorithm: The filter consists of the prediction and correction step. The previous posteriori estimate is propagated according to the prediction equation and the result is the priori estimate.

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + B_k u_{k-1} \tag{3}$$

$$\hat{P}_{k|k-1} = A_k \hat{P}_{k-1|k-1} A_k^T + Q_k \tag{4}$$

where $\hat{x}_{k|k-1}$, $\hat{P}_{k|k-1}$ are the state and covariance of the priori estimate at the time step k. $\hat{x}_{k-1|k-1}$, $\hat{P}_{k-1|k-1}$ are the state and covariance of the posteriori estimate at the time step $k-1$. Then the posteriori estimates of the state and the unknown parameters are update with the priori estimate and the measurement by the update equations.

$$K_k = \hat{P}_{k|k-1} C_k^T (C_k \hat{P}_{k|k-1} C_k^T + R_k)^{-1} \tag{5}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - C_k \hat{x}_{k|k-1}) \tag{6}$$

$$\hat{P}_{k|k} = (I - K_k C_k) \hat{P}_{k|k-1} \tag{7}$$

where I is an identity matrix. With two steps, the posteriori estimates can be calculated recursively.

2) Parameter Identification with Maximum Likelihood: At time step k, the residual between the predicted state of the Kalman filter and measurement is defined by

$$\nu_{k|k-1} = y_k - C_k \hat{x}_{k|k-1} \tag{8}$$

and its covariance can be derived by

$$\Sigma_{k|k-1} = E[\nu_{k|k-1}\nu_{k|k-1}^T] \tag{9}$$
$$= C_k E[\hat{x}_{k|k-1}\hat{x}_{k|k-1}^T]C_k^T + E[v_k v_k^T] \tag{10}$$
$$= C_k \hat{P}_{k|k-1} C_k^T + R_k \tag{11}$$

Let $\gamma \in \mathbf{R}^d$ be the unknown parameters, which is the subset of the parameter space Γ. Y^N denotes the sequence measurement vector set $\{y_1, ..., y_N\}$. The likelihood function of the state estimate is expressed with conditional probability density function.

$$l(Y^N|\gamma) = f(y_N, y_{N-1}, \ldots, y_1|\gamma) \tag{12}$$

Let's assume that the measurements are following the Markov process. The y_i only depend on the previous value y_{i-1}. Using the Bayes' theorem, the likelihood function can be rewritten

$$l(Y^N|\gamma) = f(y_N|y_{N-1}, \gamma) \ldots f(y_2|y_1, \gamma) f(y_1|\gamma) \tag{13}$$

Also we assume the measurement noise, $v_k \in \mathbf{R}^p$, is the Gaussian distribution. The probability density function takes the following equation

$$f(y_k|y_{k-1}, \gamma) = \frac{1}{(2\pi)^{\frac{p}{2}}\sqrt{|\Sigma_{k|k-1}|}} exp\{-\frac{1}{2}\nu_{k|k-1}^T \Sigma_{k|k-1}^{-1} \nu_{k|k-1}\} \tag{14}$$

Therefore, the predictive log-likelihood function[11] of the data set is

$$L(Y^N|\gamma) = -\tfrac{1}{2}\textstyle\sum_{k=1}^N p\ln(2\pi) + \ln|\Sigma_{k|k-1}| + \nu_{k|k-1}^T \Sigma_{k|k-1}^{-1} \nu_{k|k-1} \tag{15}$$

In order to estimate the unknown parameters, we have to find the parameters to maximize the log-likelihood function.

$$\hat{\gamma} = argmax\, L(Y^k|\gamma) \tag{16}$$

Then, the estimated parameter is sent to the state estimation block to figure out the system false and estimate the state of system. It is difficult to obtain the appropriate system dynamic model using the SI algorithm without enough various inputs and output to recognize the characteristic of the system. Also it takes very long time to find the unknown parameters or may not be able to estimate them, if the dimension of the state is huge. Therefore, it is necessary to have sufficient data and computational time in order to extract the system model. The reliability of the estimated parameters is the key factor of the state estimation block since the block will not be run if the confidence does not satisfy the required condition. Hence, the QAR behaves the same as the TMR.

C. State Estimation (SE)

In this section, we introduce about the hypothesis fault detection algorithm. Here, the estimation algorithm is the same as the Kalman filter of the System Identification block but it uses the dynamic model which is provided by the SI block. With two algorithms and three measurements, the SE block produces the faulty information of each module and the estimated measurement and then send the data to the voter block.

1) Hypothesis Fault Detection algorithm: There are diverse methods to assess the hypothesis such as the sequential probability ratio test (SPRT)[12], the cumulative sum (CUSUM)[13], and the generalized likelihood ratio (GLR) test[14]. In this paper, the Normalized Innovation Squared (NID) of the residual[10] is used to evaluate the hypothesis test

$$NID_k = \nu_{k|k-1}^T \Sigma_{k|k-1}^{-1} \nu_{k|k-1} \tag{17}$$

The NID follows a chi-square distribution with p degree of freedom. Let $\mathbf{H}_0$ and $\mathbf{H}_1$ denote the events are the normality and abnormality of the module, respectively.

$$\begin{cases} \mathbf{H}_0: & FI_k = 0 \quad if \quad NID_k \leq TH \\ \mathbf{H}_1: & FI_k = 1 \quad if \quad NID_k > TH \end{cases} \tag{18}$$

where FI_k is the fault indicator of the module. There is a probability of the false according to the threshold value, TH because the testing process is stochastic. If the NID is greater than the threshold then the hypothesis, $\mathbf{H}_1$, is true, and the test declares that the module fails.

D. Voting algorithm

Depending on the likelihood of the system identification, the output must be selected among the three measurements and one estimated measurement. If the value of the predictive log-likelihood function is low certain value, the SI block does not generate the system dynamic model. Therefore, the QAR works exactly like the TRM when the confidence of the estimate is low. However, the QAR chooses the estimated measurement produced by the state estimation block as its output if the confidence of the estimated dynamic model is high enough.

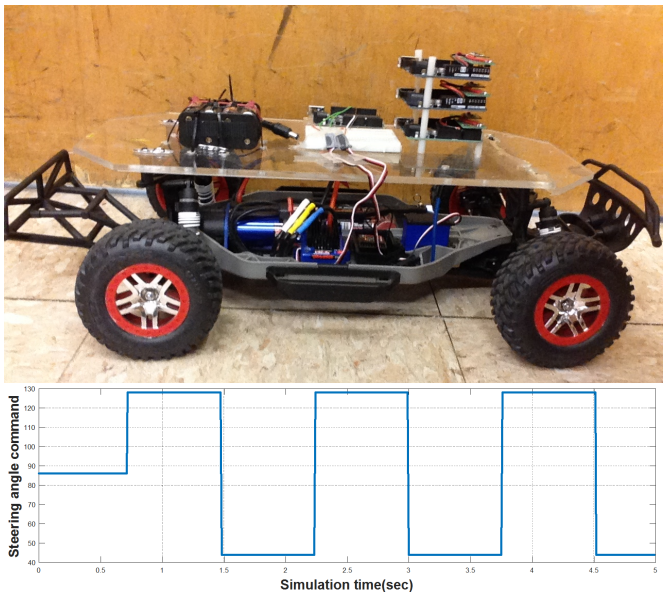

Fig. 3. Experiment system (1/10 scale car) (a) and Input data of the system (b).

IV. Experiment Result

To verify the performance of the TMR and QAR, we use a 1/10 scale rally car and three identical IMU sensors as Fig. 3(a). In order to control the direction of the car, its motor is provided with the PWM (Pulse Width Modulation) signal as an input. And three IMU sensors measure the acceleration of the car as an output. The measurements must have noise since the ground is rugged and the car shakes itself. Let's assume the noise distribution as a Gaussian distribution with zero mean and R_k covariance and we know the system input exactly.

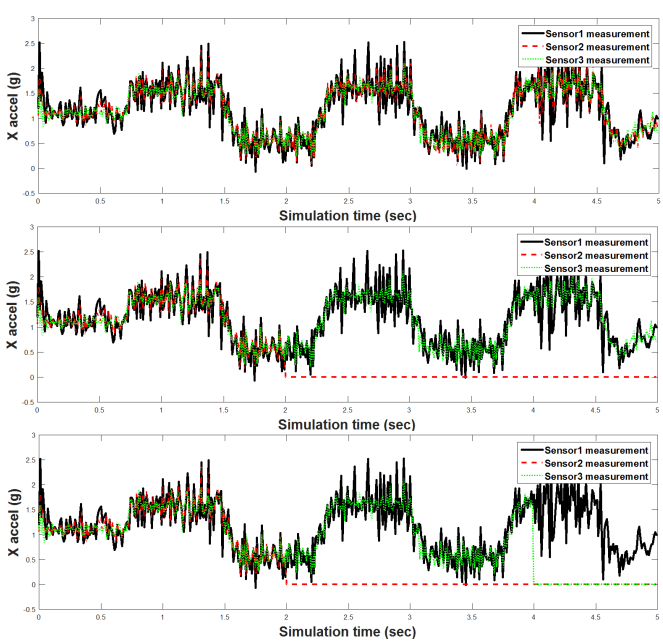

Fig. 4. Three measurements when three sensors are normal (a), sensor2 fails (b) and sensor2 and 3 fail (c).

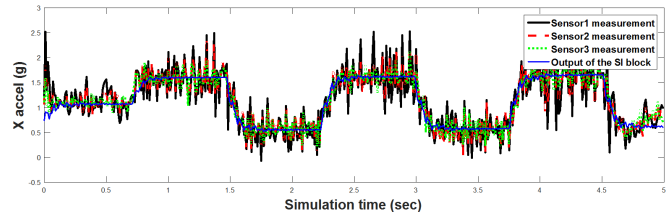

Fig. 5. Three measurements of the sensors and the output of the SI block.

For the experiment, the total simulation time is around 5 seconds and the sampling time is 0.013 seconds. Based on this information, three scenarios depending on the state of the sensors are performed. The first scenario is that three sensors are normal. Second one is that the sensor2 fails when the simulation time is 2 seconds. Finally, the sensor2 and 3 are broken at 2 and 4 seconds, respectively. Fig. 3(b) shows the input of the system. As you see the picture, the system is provided with the step input and its value is changed each time. The input is a PMW signal that is generated by the Arduino board and provided to the motor which is charge of the steering angle of the car.

Fig. 4(a), (b) and (c) illustrate the measurements when three sensors are normal, the sensor2 fails and the sensor2 and 3 are broken, respectively. The value of the broken sensor is zero since we assume that the sensor cannot send its value to the board when the sensor fails. In Fig. 4(b) and (c), the values of the sensor2 and 3 go to zero at 2 and 4 seconds, respectively. Because of the installation height and vibration degree of three sensors, the deviations of three sensors noise are different. Based on the data, we assume the noise of three sensors has zero mean and 0.4, 0.5 and 0.8 standard deviation.

One of these measurements is used in the SI block to find the proper system dynamic model. Fig. 5 denotes the measurements of three sensors and the output of the SI block. The identified system model does not equal to the dynamic model of the actual system since we don't know the exact system and the measurement also includes noise component.

Fig 6(a), (b) and (c) depict the fault indicator of the TMR in the three cases. The TMR does not need to know the system model. The simple method to find the fault of the sensor is the comparison between the values of sensors. The fault detection test technique is following

$$|y_k^i - y_k^j| < TMR\ deviation \qquad (i \neq j) \tag{19}$$

where i and j are the sensor number. We set that the deviation value is 0.3. Originally, there are several faults of the fault indicator because of the measurement noise. We put even though the sensors are normal like Fig. 6(a). In Fig. 6(b), the indicator changes at 2 seconds. The figure shows the fault identifier system of the TMR operates well. However, the fault indicator of the TMR does not perform properly any more if two sensor are broken around 4 seconds. In Fig. 6(c), the sensor2 and 3 are broken but the values of

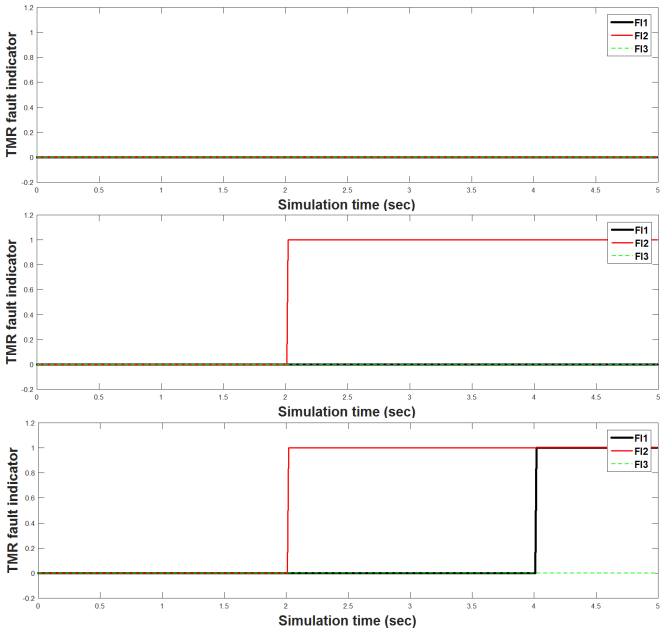

Fig. 6. Fault indicator of TMR when three sensors are normal (a), sensor2 fails (b) and sensor2 and 3 fail (c).

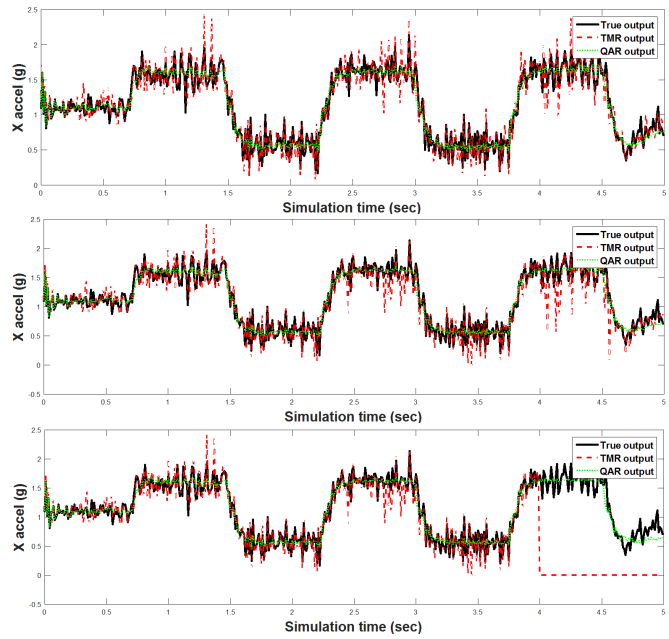

Fig. 8. True output and TMR/QAR outputs when three sensors are normal (a), sensor2 fails (b) and sensor2 and 3 fail (c).

the fault indicator of the sensor1 and 2 are 1. The TMR cannot recognize the faulty of the system when two sensors are broken.

Fig 7(a), (b) and (c) show that the QAR estimates the faulty of the system depending on the malfunctions of the sensors. The fault detection test of the QAR is following

$$NID_k = \nu_{k|k-1}^T \Sigma_{k|k-1}^{-1} \nu_{k|k-1} < 1\,(1\,sigma) \tag{20}$$

We set the value of the threshold with 1 which means three sigma. Fig. 7(c) demonstrates that the QAR can figure out the faulty of two sensors at 4 seconds. Compare with the fault indicator of the TMR, the QAR find the fault of the sensor exactly when two sensors are failure. Therefore, this result shows the QAR is more robust with respect to the system fault than the TMR.

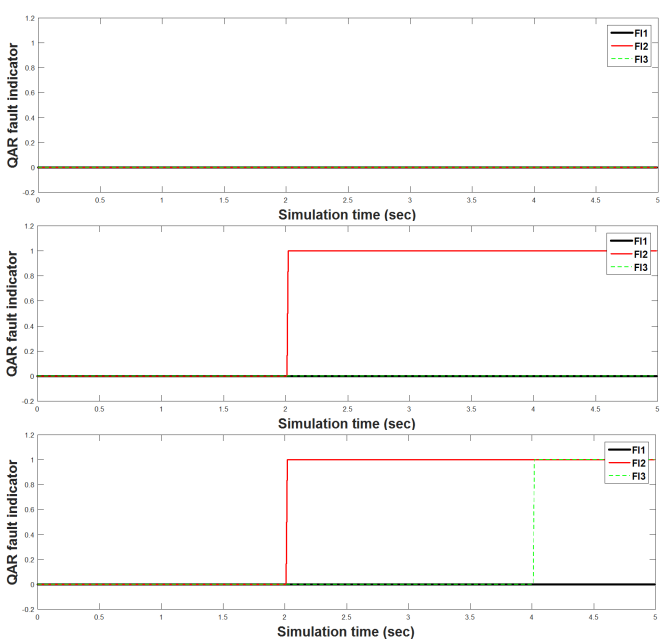

Fig. 7. Fault indicator of QAR when three sensors are normal (a), sensor2 fails (b) and sensor2 and 3 fail (c).

Fig 8(a), (b) and (c) illustrate the true system output and the output of TMR and QAR. As the Fig 8(a) and (b), the outputs vary a little because the TMR and QAR can figure out the false of the sensor and handle the fault when the sensors are normal or even one sensor are failure. However, Fig 8(c) shows the weakness of the TMR clearly. The output of the TMR cannot follow the true output of the system even though the output of the QAR estimate quite well the true output.

V. Conclusion

This paper has investigated the redundant method against the system fault. TMR is most widely used method to detect and the deal with system malfunctions. To improve its robustness, we present the QAR method that includes the system identification, filtering and hypothesis fault detection algorithms using the additional virtual module. Therefore, the proposed Quadruple Adaptive Redundancy method has been demonstrated with three cases of the sensor breakdown through the real system experiment. The QAR improve not only the robustness of system with the hypothesis fault detection method but also the performance of the estimate.

References

[1] D. Stewart, R. Volpe, and P. Khosla, Design of dynamically reconfigurable real-time software using port-based objects, Software Engineering, IEEE Transactions on, vol. 23, no. 12, pp. 759-776, 1997.

[2] D. Stewart, R. Volpe, and P. Khosla, Integration of Real-Time Software Modules for Reconfigurable Sensor-Based Control Systems, International Symposium on Intelligent Robotics, January, 1993.

[3] Y Cui, RM Voyles, JT Lane, A Krishnamoorthy, MH Mahoor, A mechanism for real-time decision making and system maintenance for resource constrained robotic systems through ReFrESH, in Autonomous Robots 39 (4), 487-502, 2015.

[4] R. E. Lyons and W. Vanderkulk, The use of triple-modular redundancy to improve computer reliability, IBM Journal of Research and Development, vol. 6, no. 2, pp. 200-209, 1962.

[5] M. Augustin, M. Gossel and R. Kraemer, Implementation of Selective Fault Tolerance with Conventional Synthesis Tools, in IEEE 14th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Cottbus, Germany, pp. 213-218, 2011.

[6] M. Augustin, M. Gossel, R. Kraemer, "Reducing the Area Overhead of TMR-Systems by Protecting Specific Signals", in Proceedings of the IEEE i6'h international On-Line Testing Symposium, Los Alamitos, CA, USA: IEEE Computer Society, pp. 268-273, 2010.

[7] L. Rui and K. Yan-jia, A Method of Synchronous-feedback Based State Machine with Triple Modular Redundancy, Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference, pp. 136-139, 2014.

[8] G. Latif-Shabgahi, J. M. Bass and S. Bennett, Integrating Selected Fault Masking and Self-Diagnosis Mechanisms, Parallel and Distributed Processing, 1999. PDP '99. Proceedings of the Seventh Euromicro Workshop on pp.97-104, 1999.

[9] G. Latif-Shabgahi, S. Bennett, Adaptive majority voter: a novel voting algorithm for realtime fault-tolerant control systems, EUROMICRO Conference, 1999. Proceedings. 25th Vol. 2, pp.113-120, 1999.

[10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction, New York: Wiley, 2001.

[11] F. Kwasniok, Estimation of noise parameters in dynamical system identification with Kalman filters, Physical Review E Scope Description to Include Biological Physics, Vol. 86, Iss. 3, 036214-1-036214-8, 2012.

[12] D. P. Malladi, and J. L. Speyer, A Generalized Shiryayev Sequential Probability Ratio Test for Change Detection and Isolation, IEEE Transactions on Automatic Control, Vol. 44, No. 8, pp. 1522-1534, 1999.

[13] I. V. Nikiforov, A Generalized Change Detection Problem, IEEE Transactions on Information Theory, Vol. 41, No. 1, pp. 171-187, 1995.

[14] D. Dionne, Y. Oshman, and D. Shinar, Novel Adaptive Generalized Likelihood Ratio Detector with Application to Maneuvering Target Tracking, Journal of Guidance, Control, and Dynamics, Vol. 29, No. 2, pp. 465-474, 2006.