We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

11

Fault Detection, Isolation, and Control of Drive By Wire Systems

Sohel Anwar
Indiana University Purdue University Indianapolis
USA

1. Introduction

Fault tolerance is a property of a system that continues operating properly in the event of failure of some of its parts. It provides the ability of a system to provide a service complying with the specification in spite of faults. If operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively-designed system in which even a small failure can cause total breakdown.

Fault-tolerant design of a drive-by-wire (DBW) system provides key-method for achieving safe and reliable systems (Isermann et al, 2002; Shibahata, 2005). It is generally required in the design of all mechanical, electrical and software components of drive-by-wire systems. Extensive analysis is performed during the design and testing phase, combined with quality control methods during manufacturing. This design method reduces faulty behaviour dramatically and forms the basis for safe and reliable systems. The design method identify potential hazards and associated avoidance requirements, translates safety requirements into engineering requirements, provides design assessment and trade-off to support the ongoing design, assesses the relative compliance of the design to the requirements and document the findings, directs and monitors specialized safety testing, and monitors and reviews test and field issues for safety trends.

It is noted that certain faults and failures cannot be avoided totally. Components in a system can fail even after with a fault tolerant design. These faults should be tolerated by additional measures and compensated in such a way that they do not cause the overall system (or certain critical functions) to fail. The most obvious way to reach this goal is to implement duplication in order to avoid single points of failure.

Duplication can offer fault-tolerance in three different ways:

- Replication: Providing multiple identical instances of the same system, directing tasks or requests to all of them in parallel, and choosing the correct result on the basis of a quorum.
- Redundancy: Providing multiple identical instances of the same system and switching to one of the remaining instances in case of a failure (fall-back or backup).

- Diversity: Providing multiple different implementations of the same specification, and using them like replicated systems to cope with errors in a specific implementation.

For redundancy, ideally it is desirable to have as much redundancy as possible. Instead of just two, or three, sensors one would perhaps want ten to be even safer. But the design is limited by cost and weight. Diversity in redundant components is desirable. There is also redundancy in software, where different programming teams work on solving the same problem using different methods. This method gives higher levels of safety than just duplicating the same code when running multiple redundant systems. Otherwise the same error is likely to happen in the backup system as well. There are three types of hardware redundancies as described below:

1. Static hardware redundancy

Three or more parallel modules are used that have the same input signal and are all active. A voter compares their output signals and decides by majority which is the correct one. In a commonly used triple-redundant modular architecture one fault can be masked without the use of special error detection methods. The output is parsed through a voter that examines the values. Decision is made based on what the majority thinks is a correct view of the reality. There can be a single module that fails, sending faulty output values, and system still have a correct view. For a system having n redundant modules, system will tolerate (n-1)/2 faults. n must be odd so that there exists no tie. Static redundancy is based on the voting of the outputs of a number of modules to mask the effects of a fault within these units. The simplest form of this arrangement consists of three modules and a voter (TMR).

2. Dynamic hardware redundancy

It requires fewer modules at the cost of more information processing. A minimal configuration consists of two modules, where one module is in operation, and the second module takes over in case of an error. When the second module is continuously operating this method is called 'hot standby', which has the advantage of shorter downtime of the system. However since it is operating all the time aging of the module becomes a disadvantage. In a 'cold-standby' configuration, the backup system is normally out of function. It only becomes operational in case of an erroneous primary system. Both configurations need for error-detection methods observing if a module becomes faulty. Methods range from simple ones such as limit value and plausibility checks, parity checking and watchdog timers, to sophisticated signal-model-based or process-model-based methods. In dynamic redundancy fewer modules are used but we have to do more information processing. The system has to detect if a module is malfunctioning and reconfigure the system so that the module is shutdown and the backup module is brought online. Dynamic redundancy based on fault detection rather than fault masking, achieved by using two modules and some sort of comparison on their outputs that can detect possible faults. It offers lower component count but is not suitable for real-time applications.

3. Hybrid hardware redundancy

In this configuration, a combination of voting, fault-detection, and module switching techniques is used. It is a hybrid of static and dynamic redundancies.

In order to bring down the cost, the total number of redundant components must be reduced without compromising the fault tolerance. One possible solution to this problem

is to utilize analytical redundancy or model based fault detection, isolation, and accommodation. Model-based Fault Detection and Isolation explicitly use a mathematical model of the system. It is motivated by the conviction that utilizing deeper knowledge of the system results in more reliable diagnostic decisions. The main idea is "analytical redundancy" which makes comparison of measurement data with known mathematical model of the physical process. It is superior to "hardware redundancy" generated by installing multiple sensors for the same measured variable. They offer simplicity, flexibility in the structure, less hardware, less weight, and cost.

2. Analytical Redundancy

Model-based Fault Detection and Isolation is achieved by implementing more complex failure detection algorithms that take careful account of system dynamics; utilizing such algorithms, one may be able to reduce requirements for costly hardware redundancy. Analytical redundancy based FDI (failure detection and isolation) uses a model of the dynamic system to generate the redundancy required for failure detection. In many systems, all of the states cannot be measured because of cost, weight and size considerations, therefore, FDI schemes for such systems must extract the redundant information from dissimilar sensors, using the differential equations that relate their outputs. In addition to taking hardware issues into consideration, the designer should consider the issue of computational complexity. Most model-based FDI methods rely on analytical redundancy. In contrast to physical redundancy, when measurements from different sensors are compared, now sensory measurements are compared to analytically obtained values of the respective variable and the resulting differences are called residuals. The deviation of residuals from the ideal value of zero is the combined result of noise, modelling errors and faults. A logical pattern is generated showing which residuals can be considered normal and which ones indicate a fault. Such a pattern is called the signature of the failure. The final step of the procedure is the analysis of the logical patterns obtained from the residuals, with the aim of isolating the failures that cause them. Such analysis may be performed by comparison to a set of patterns known to belong to sample failures or by the use of some more complex logical procedure.

3. Case Study: Analytical Redundancy Based Fault Detection, Isolation, and Accommodation of a Steer By Wire System

Proposed Methodology

It is clear that a major challenge for steer by wire (SBW) system is to reduce the cost and the number of hardware components (e.g. sensors) while maintaining safety. Analytical redundancy method has been investigated by several applications before (Gertler, 1992; Dong & Hongyue, 1996; Suzuki et al, 1999; Venkateswaran et al, 2002; Anwar & Chen, 2006). However, most of these investigations focused on hardware redundancy based fault detection and simulation on aircraft applications as opposed to hardware experiment of analytical redundancies for automobiles with SBW system. The objective of this research is to study the feasibility of an analytical redundancy method for a SBW system through hardware-based experiment instead of software simulation.

The analytical redundancy method discussed in this paper makes the following assumptions: The probability of two sensors failing simultaneously is much less than that for only one sensor failing. Similarly, the probability of three sensors failing simultaneously is much less than that for two-sensor-fault.
It simulation studies, it has been shown that the accuracy of the Fault Detection and Isolation Accommodation (FDIA) algorithm won't be sacrificed by the reduction of the number of physical sensors since analytical redundancy will provide additional information (Anwar & Chen, 2006).
The research methodology in this paper involves several parts. These parts are:

- Build of a simplified SBW system hardware-in-loop (HIL) bench with necessary hardware components.
- Development of control models for SBW system. These models involve:
  - A modified 4th order vehicle model
  - A nonlinear observer model based on the vehicle model
  - A long range prediction model with different prediction horizons
  - A FDIA algorithm for sensor fault detection based on analytical redundancy.

Completion of the build of the HIL bench for the SBW system (illustrated in Figure 1) involves:
• A steering wheel module with three angular sensors and one electric motor for haptic feedback.
• A rapid prototyping controller from dSpace, MicroAutoBox Model 1401, which receives input signals from all angular sensors on the steering wheel, feedback signals from all angular sensors on the pinion of the rack and pinion assembly, sends commands to motor drivers and servo motors. And, all necessary wiring and cable devices between controllers, drivers, motors, and sensors.
• A set of motors: the motor drivers, Parker OEM770T, the servo motors, Parker BE342KJ-K10N, and a set of gears and a gear box converting the servo motor's rotation into rack's linear motion.
• A rack and pinion assembly with fixture. The rack and pinion assembly is that of a Volkswagen SI-9281-9-2, which provides lateral movement of the steering system. Three angular sensors situated on the pinion provide the feedback signal to controller. And two springs with spring coefficient 2000 N/in are attached a both ends of the rack to simulated the road loads for the SBW HIL bench.
For real vehicles, the road wheel friction force created at the contact patch of road and tire can be transmitted to rack via front wheels. In building the hardware of the SBW system in this paper (Figure 1), there are two springs that replace front wheels and these two springs also provide simulated force caused by front wheels contacting road surface. The rack was made for power steering wheel system (with pinion on top). In a close-loop control system, there must be a feedback from the road wheel system (rack/pinion). Thus, the pinion is taken as the feedback signal source for close-loop control (Figure 2).

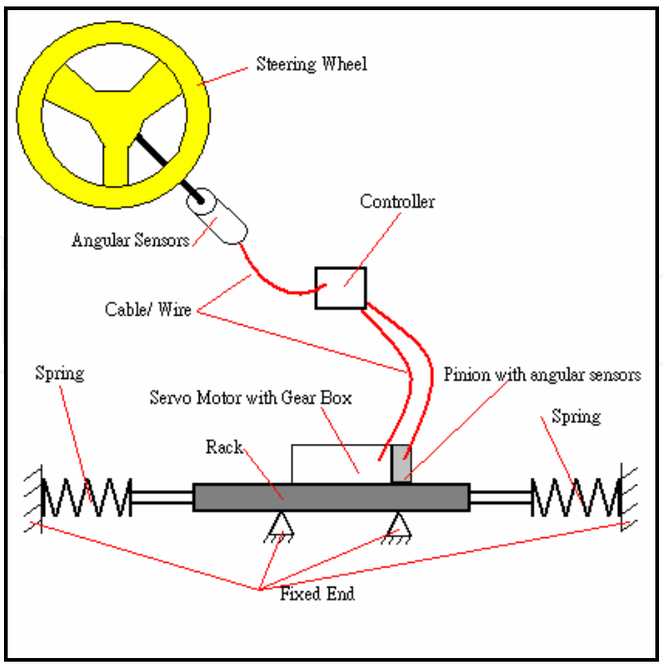

Fig. 1. Loop bench for simplified SBW system

Developing the overall analytical redundancy based FDIA and control model and performing the validation tests of FDIA algorithms is the main work in this research. Before modelling the FDIA algorithms as a part of fault tolerant control, it is necessary to understand the complete control model considered in this paper (Figure 3). The complete model shows that the fault tolerant control is made possible by FDIA. θ (pinion angle) from vehicle model and from predictor are checked by FDIA logic to isolate any sensor fault on the pinion. The vehicle model converts motor's current into pinion angle via the nonlinear observer. Ideally, we wished to make the steering wheel rotation and the pinion angle rotation simultaneously. But, due to the dynamic model used in the observer and usage of low pass filter to reduce noise in the feedback signals, a delay is introduced in the after the pinion angle is reconstructed using the observer. In order to reduce this delay, a long range predictor is used in the observer model that reduces the delay between steering wheel's angle and pinion angle (Anwar & Chen, 2006). And, the observer in the overall model provides the information about other state variables inside the controller as well (Hasan & Anwar, 2008; Nise, 1994).
Lastly, by taking the advantage of Matlab/Simulink and the code-compiling functions provided by dSpace, the steering system model, the modified vehicle model, the nonlinear observer, the predictor, and the FDIA algorithm were implemented in real-time environment which was then downloaded into the dSpace MicroAutoBox controller.

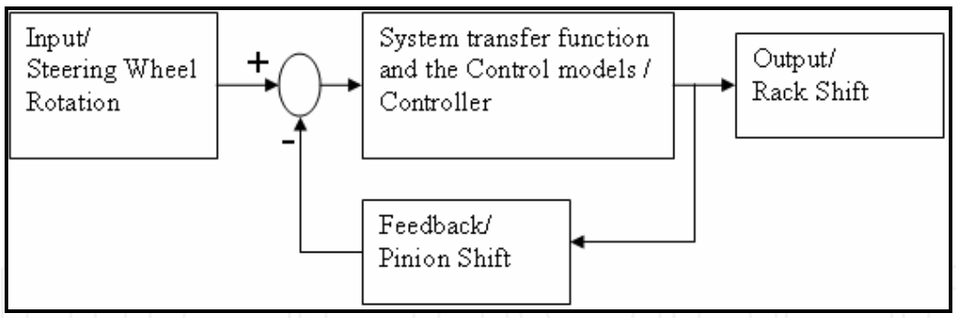

Fig. 2. Flow chart of simplified SBW system with mechanism/component

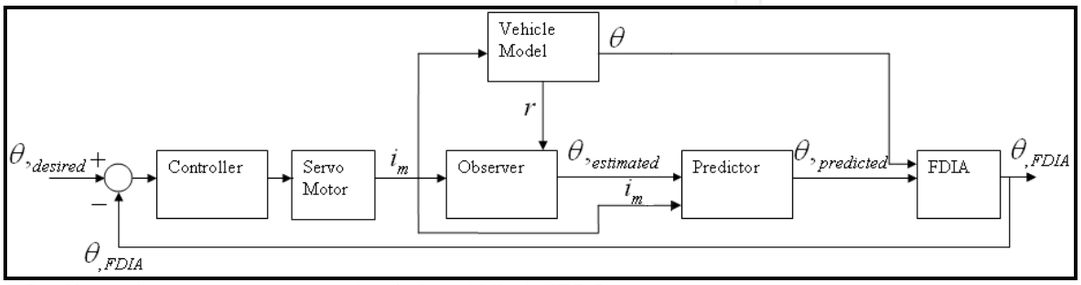

Fig. 3. Complete control model of simplified SBW system

3.2 SBW Hardware-In-Loop Bench Design and Construction

In building the loop bench, there are several components and parts (Table 1) used in machine constructing. The overall road wheel portion (including rack and pinion) of the HIL bench is shown in Figure 4.

Name of Component/Part	Quantity	Description
Servo Motor	1	Parker BE342KJ-K10N, with a gear box coupled with the rack.
Drive	1	Parker OEM770T, selected for the servo motor.
Rack	1	Volkswagen SI-9281-9-2, with a pinion on top
Angular Sensor	3	Model: MH22B, coupled with the pinion
Steering Wheel System	1	The system has been already built before, with angular signal as output.
Controller	1	DSpace MicroAutoBox 1401

Table 1. List of components/parts for building the loop bench

For the design and construction of mechanical components of the bench, each component was been modelled by using Pro/E Wildfire 3.0 and tested by using ANSYS 14.0 to check the maximum deformation within the material (steel).

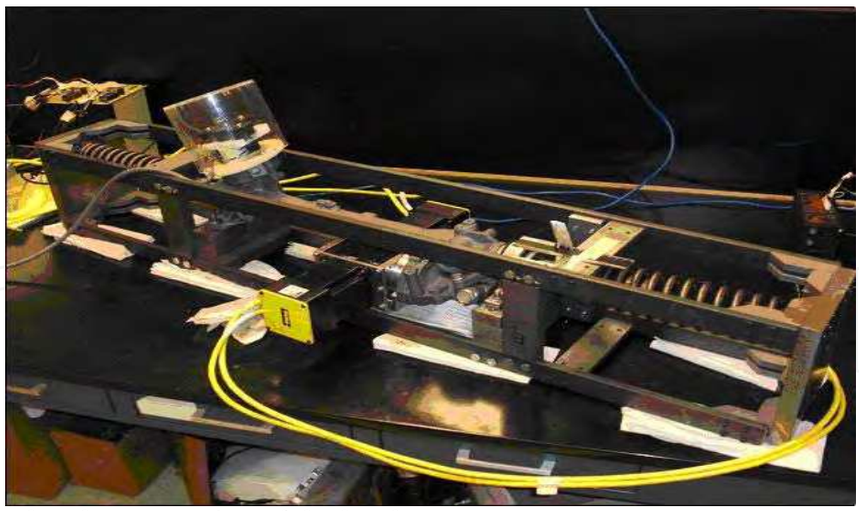

Fig. 4. Completed construction of the hardware

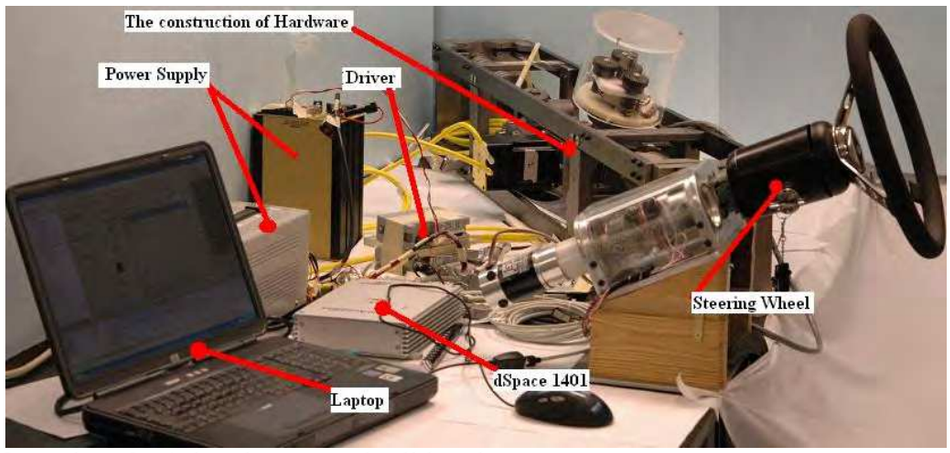

Fig. 5. Complete SBW hardware with all hardware components

Once the construction of hardware was finished, all the electrical and electronic components of the HIL bench (listed in Table 1) were then connected using the designed wiring diagram. The complete SBW system with the mechanical components, driver, controller, laptop computer, and power supply is represented in Figure 5.

4. Observer, Predictor, and FDIA Algorithm

A majority of the theoretical developments for analytical redundancy has been described in references (Anwar & Chen, 2006; Hasan & Anwar, 2008). A summary of the observer, predictor, and FDIA algorithms are presented below.
The overall vehicle and steering system model are given by (Anwar, 2005):

$$\dot{x} = Ax + Bi_m$$
$$y = Cx$$
$$x = \begin{bmatrix} \beta \\ r \\ \theta \\ \dot{\theta} \end{bmatrix}; A = \begin{bmatrix} \frac{-C_{\alpha,f} - C_{\alpha,r}}{mV} & -1 + \frac{C_{\alpha,r}b - C_{\alpha,f}a}{mV^2} & \frac{C_{\alpha,f}}{mV} & 0 \\ \frac{C_{\alpha,r}b - C_{\alpha,f}a}{I_z} & \frac{-C_{\alpha,f}a^2 - C_{\alpha,r}b^2}{I_z V} & \frac{C_{\alpha,f}a}{I_z} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{(t_p + t_m)C_{\alpha,f}}{J} & \frac{a(t_p + t_m)C_{\alpha,f}}{JV} & \frac{-(t_p + t_m)C_{\alpha,f}}{J} & \frac{-b}{J} \end{bmatrix} \quad (1)$$
$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_m}{J} \end{bmatrix}; E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{J} \end{bmatrix}; C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

Whrere, β is the vehicle body side slip angle, r is the vehicle yaw rate at the center of gravity, θ is the steering angle at road wheel, $C_{\alpha,f}$ & $C_{\alpha,r}$ are cornering coefficients for the front & rear wheels respectively, a & b are distance between front & rear axles to the vehicle center of gravity respectively, m is vehicle mass, V is vehicle nominal velociy, I_z is vehicle inertia in yaw direction, J is wheel ineria, t_m & t_p are mechanical and pneumatic trails repectively, k_m is the motor torque constant. The details of the above model equations can be found in (Hasan, 2007) and is based on simplified single track vehicle model (or bicycle model) as illustrated in Figure 6.

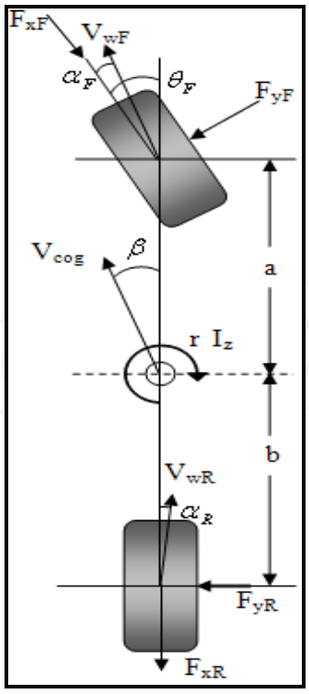

Fig. 6. Linear vehicle model simplified as bicycle model

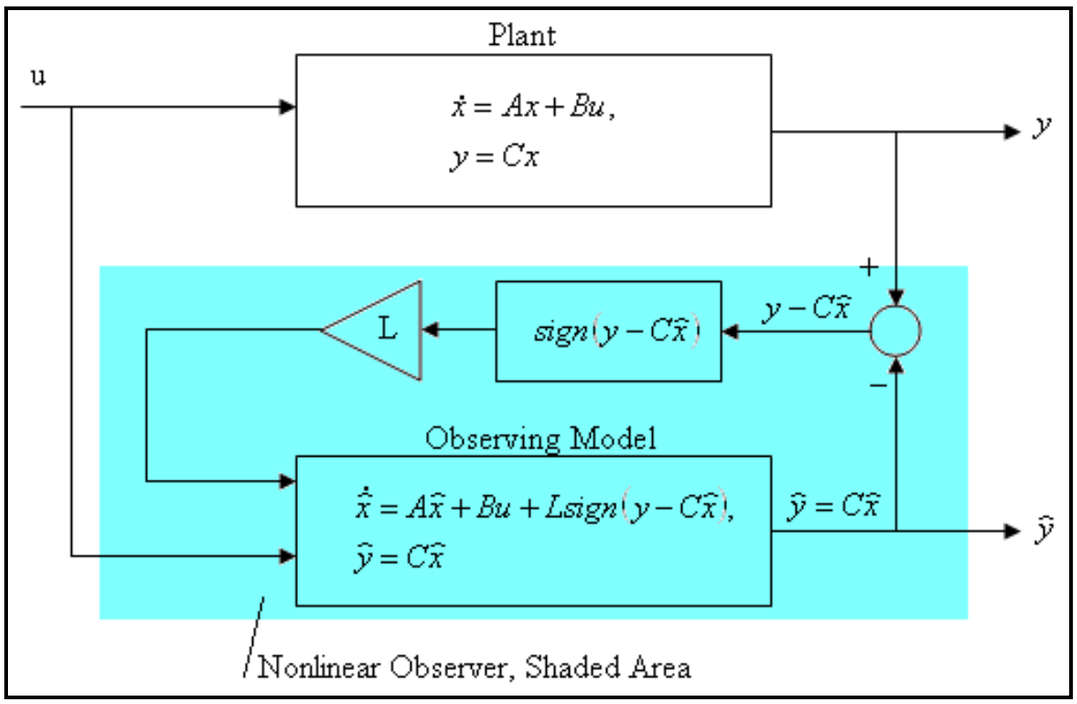

Fig. 7. Nonlinear observer (sliding mode) in state space form

A sliding mode observer was designed based on the above vehicle model. The details of the sliding mode observer developed are described in (Hasan, 2007; Utkin et al, 1999; Stotsky, & Hu, 1997) and hence skipped here. A schematic of this observer is given in Figure 7.

A long range prediction algorithm was designed based on Diophantine equation (Clarke et al, 1987). This algorithm was then used to predict the future output (e.g. steering angle at road wheel) which is then used in the FDIA algorithm. This objective of this predictor is to reduce the latency in fault detection due to computational delays. A detailed account of this prediction algorithm can be found in (Hasan & Anwar, 2008). Only prediction equation is stated here for reference.

$$\theta(t+j) = E_j(z^{-1})A(z^{-1})\Delta U(t+j-1) + F_j(z^{-1})\theta(t) \quad (2)$$

Where, t is current time, z^{-1} is the backward shift operator in z-domain, E_j and F_j are polynomials in Diophantine equation which are uniquely defined given the system characteristic polynomial $A(z^{-1})$, ΔU is the system input.

The fault detection, isolation, and accommodation algorithms (FDIA) are based on a majority voting scheme and are given in details in (Hasan, 2007; Niu, 2009). Only the fundamental algorithm in flow chart format is given here for brevity (Figures 8 and 9).

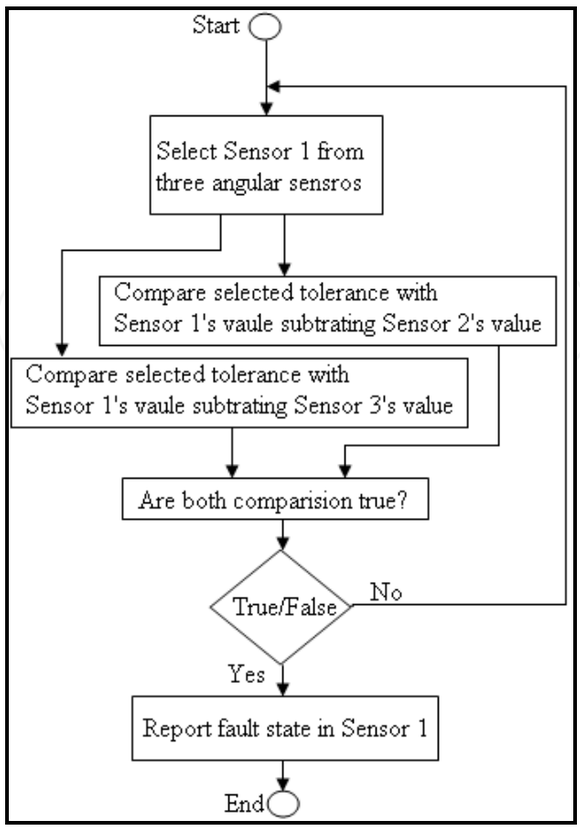

Fig. 8. Single-sensor FDIA logic

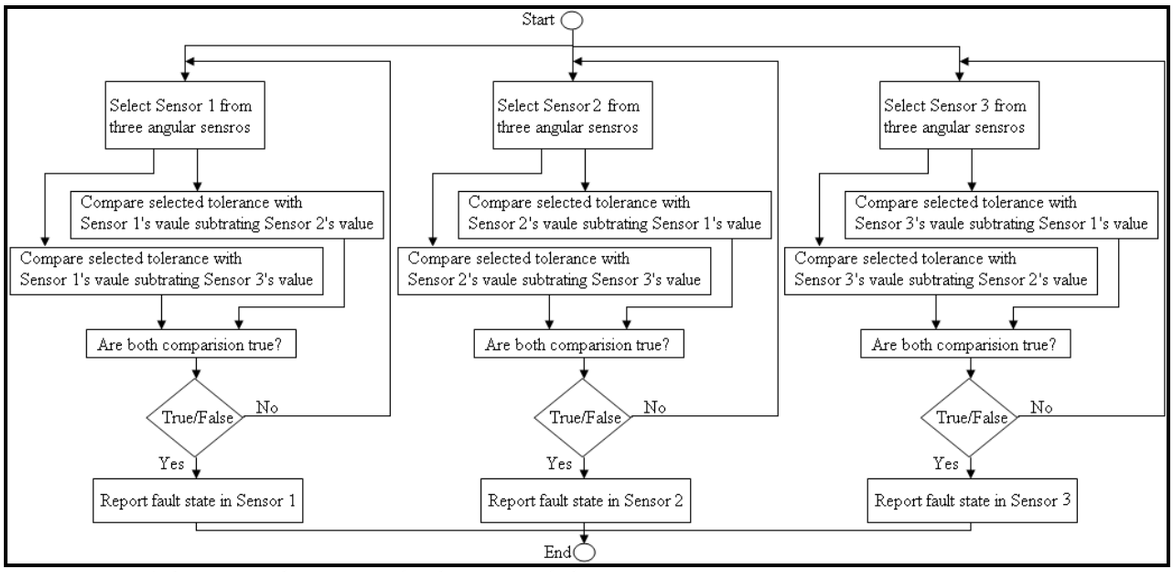

Fig. 9. FDIA logic in detecting fault state of three sensors

5. Experimental Results

Before we illustrate the FDIA algorithms used in this work, it is necessary to describe the types of Faults which are tested in this research. Commonly, faults can be classified in three main types: transient fault (Figure 10), hard fault (Figure 11), and soft fault (Figures 12 – 13).

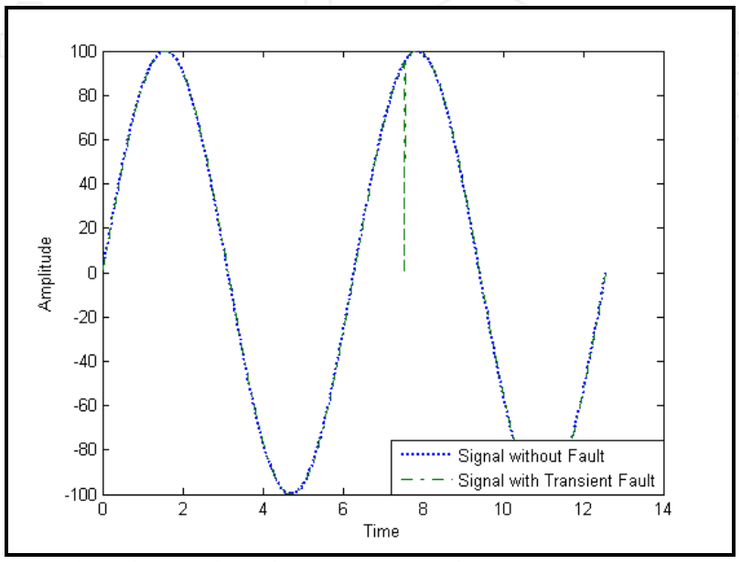

Fig. 10. Normal signal and signal with transient fault

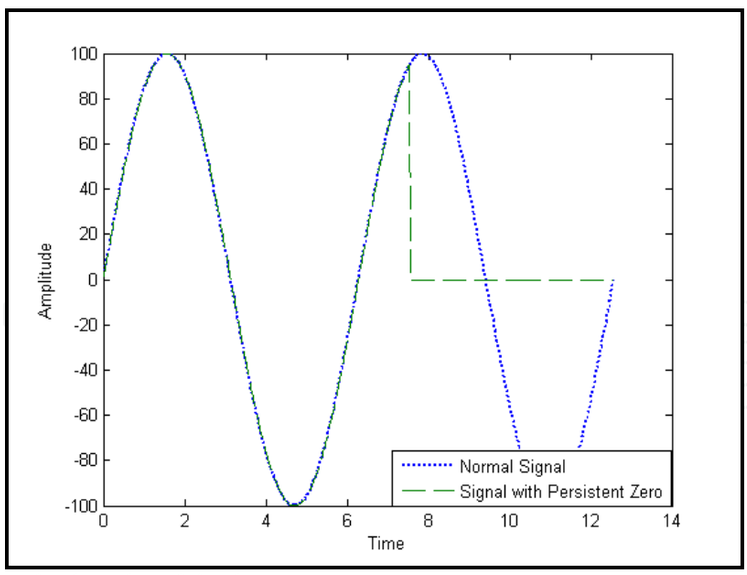

Fig. 11. Persistent fault, a hard fault

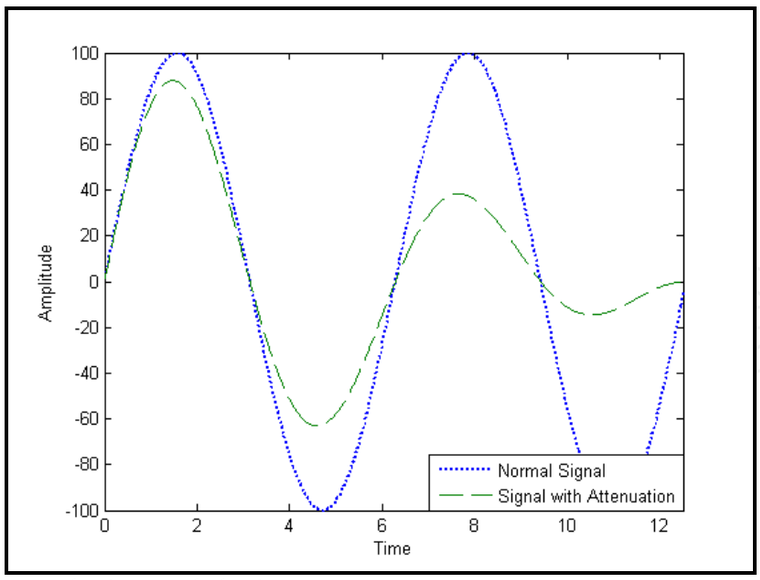

Fig. 12. Attenuating fault, a soft fault

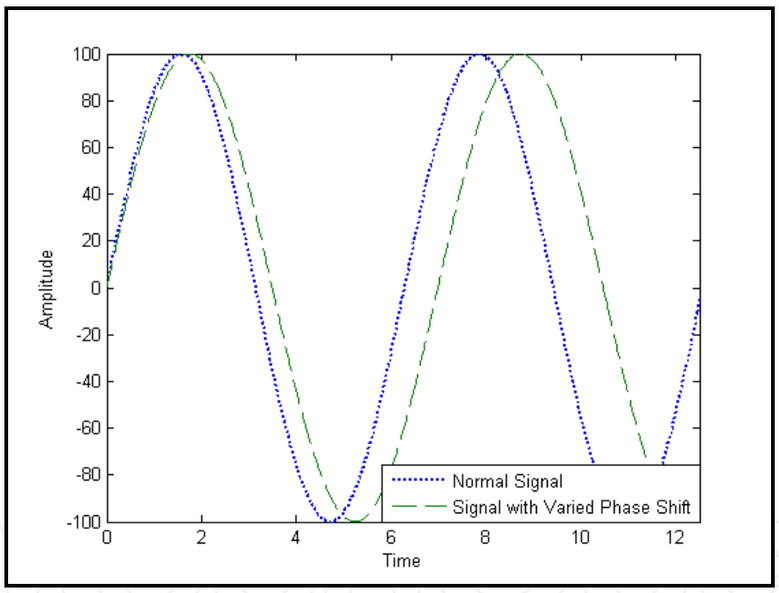

Fig. 13. Variable phase shift, a soft fault

5.1 Servo Motor Control Modeling by Using Matlab/Simulink

In order to build the control program for the experiment with the SBW system, it is first necessary to develop the basic functions to control the angular rotation of motor via feedback information from steering wheel's angular sensors. Servo motor control function (Figure 14) is the combination of the controller block and the servo motor block in Figure 3, and involves a counter, a PID controller, and a command block.

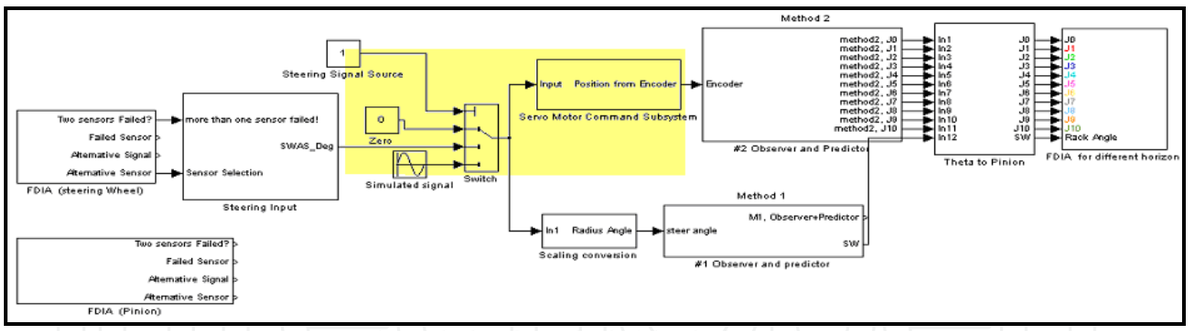

Fig. 14. Motor rotation control made by Simulink (shaded area)

5.2 Vehicle Parameters

The vehicle parameters used in a previous work (Hasan & Anwar, 2008) has been adopted for the modified vehicle model here and is modeled in Matlab/Simulink. These vehicle parameters are given in Table 2.

Parameter	Value
$C_{a,f}$	22000 (N/rad)
$C_{a,r}$	55000 (N/rad)
m	1400 (kg)
V	15 (m/s)
b	1.5 (m)
a	1 (m)
I_z	4000 (kg-m2)
t_p	0.025 (m)
t_m	0.03 (m)
J	20 Kg-m2

Table 2. Parameter values used for the modified vehicle model

A validated sliding mode observer model was developed using the HIL bench. The overall observer model along with the predictor is shon in Figure 15 will all the input/output relationships. It is noted that if the steering model in Figure 15 is $G(s)=\theta(s)/i_m(s)$, then the expression for motor current reading is be given by $i_m(s) = G^{-1}(s)\theta(s)$.

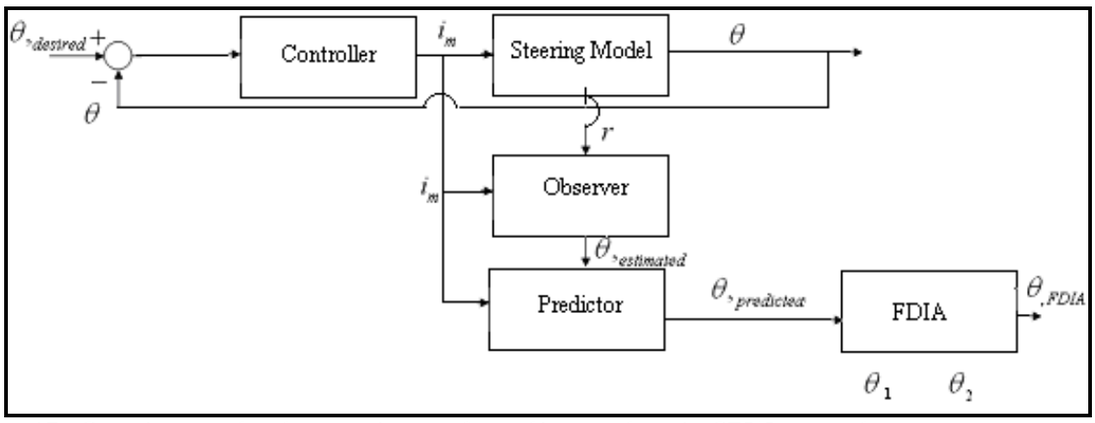

Fig. 15. Steering angle observation and predictor taken in SBW control system

Usage of ControlDesk program provides a convenient connection between modelling control program using Matlab/Simulink and using dSpace D1401 as a prototyping controller. After the control model built by using Matlab/Simulink is downloaded into D1401, ControlDesk program can help users to view and record various model parameters and inputs/output data. Additionally, other functions such as testing hardware with any control model, reporting state, etc. can also be performed. A number of interfaces were built by using ControlDesk to test steering function to move the rack, to test FDIA logic with different fault types, and to test the detection speed in detecting fault of predicted with different prediction horizon.

The purpose of adding a predictor in a control system is to reduce the delay between input and system's response. If we remove the FDIA and Predictor, then the system's feedback in a close-loop control system becomes $\theta_{estimated}$. Similarly, if the predictor hasn't been removed, then $\theta_{predicted}$ can be equal to $\theta_{estimated}$ when selecting the horizon as zero ($J = 0$).

Now, Figure 16 and Figure 17 represent the delay between $\theta_{desired}$ and $\theta_{estimated}$. In Figure 15, it is the same as the delay between $\theta_{desired}$ and $\theta_{predicted}$ for J equal to zero because $\theta_{estimated}$ is equal to $\theta_{predicted}$ for $J = 0$.

From these figures, it is noted that the controller with servo motor in the SBW System causes a 40 millisecond delay between $\theta_{desired}$ and $\theta_{estimated}$.

When the SBW System with Predictor (Figure 3) is used with different number of J, it is noted that selecting a higher number of J makes $\theta_{predicted}$ closer to $\theta_{desired}$ (Figure 17). Thus, using long range predictor with higher number of horizon can shorten the delay time between steering angle input and predicted steering angle output.

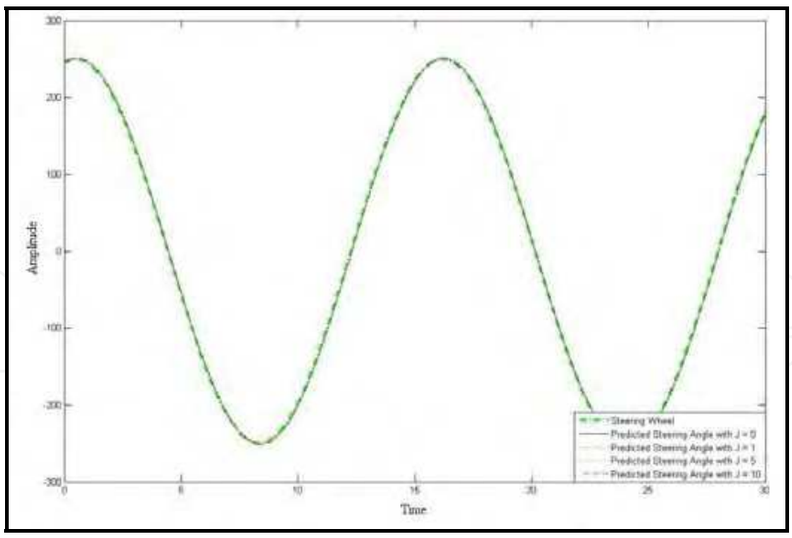

Fig. 16. Steering angle and estimated steering angle

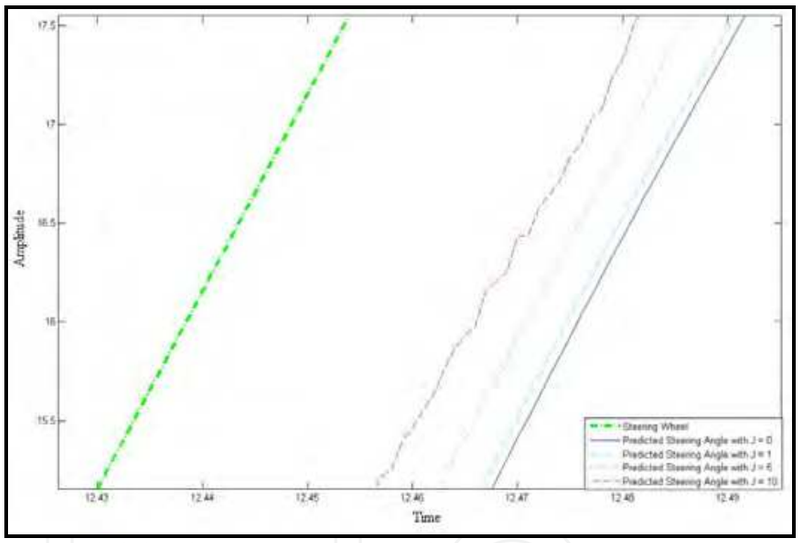

Fig. 17. A 0.04-second delay between $\theta_{desired}$ and $\theta_{estimated}$

In this paper, a number of tests are designed to test whether the FDIA is able to detect transient, hard, and soft faults with the sensors attached with pinion, not $\theta_{predicted}$ with different horizon.

In this section, the FDIA is tested with transient fault, hard fault (persistent zero and constant phase shift), and soft fault (increased amplitude, attenuated amplitude and variable phase shift). With appropriate designed FDIA using Matlab/Simulink and using Control Desk program, the FDIA interface built with Control Desk program is able to report the times of transient fault, the fallen sensor, and the state of morn than one sensor fallen.

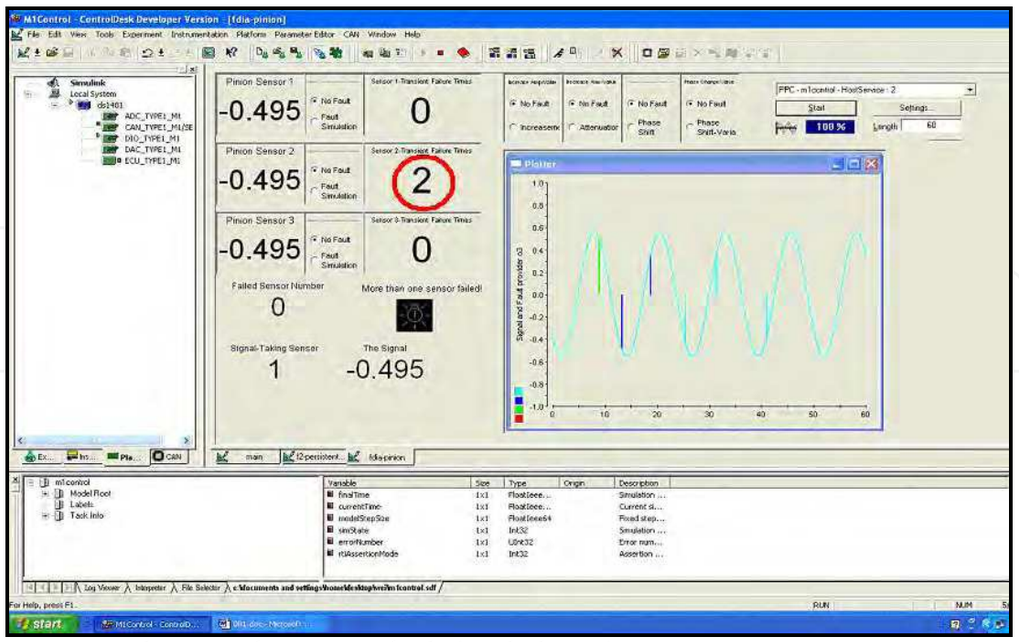

Fig. 18. Reported times of transient fault (circled area)

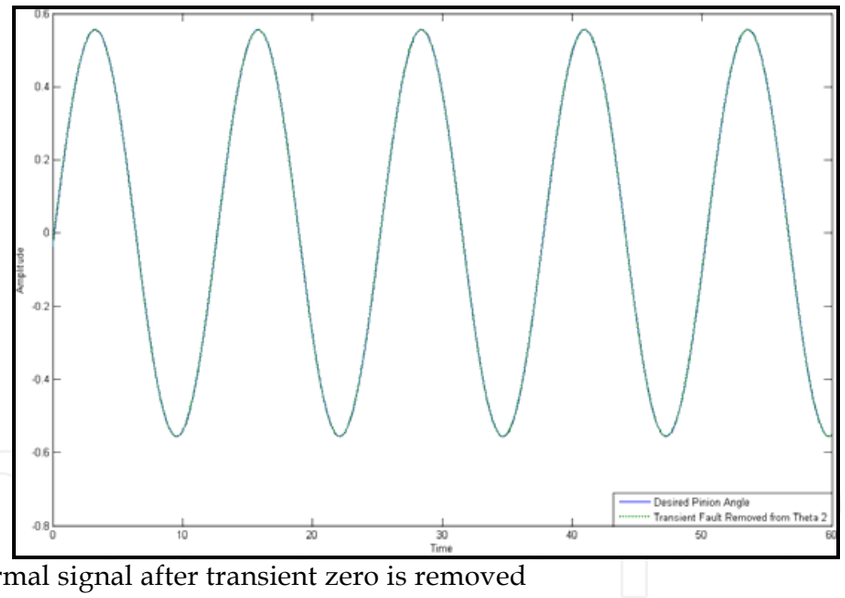

Fig. 19. Normal signal after transient zero is removed

To test transient fault and hard fault for the SBW System with mechanism, three manually operated switched are wired between the angular sensors coupled with pinion and D1401. Also, to test soft fault, Simulink blocks with signal-increasing, signal-attenuating and phase-shifting functions are added into the SBW System.

After manually operated switches and Simulink blocks are made, the ability of detecting fault using FDIA algorithm with transient, hard, and soft faults respectively was tested.

In testing transient fault, the transient-fault signal is made by quickly turning off and then turning on a manually operated switch with a selected sensor coupled with pinion. Thus,

a transient signal can be created for testing purpose. After doing experiment, the FDIA interface (Figure 18) represents the detection of transient fault with θ_2; it shows that the FDIA interface is able to record the times of transient fault. Certainly, the interface keeps the times of transient fault (Figure 18) after transient fault is removed (Figure 19).
Next, we test persistent zero with FDIA interface. The persistent-zero signal is created by turning off the manually operated switch with a selected sensor on pinion. Then, the interface immediately reports a fault state of the selected sensor (Figure 20). Later, once the switch is turned on (Figure 20), the fault state is updated as fault-out (Figure 21).

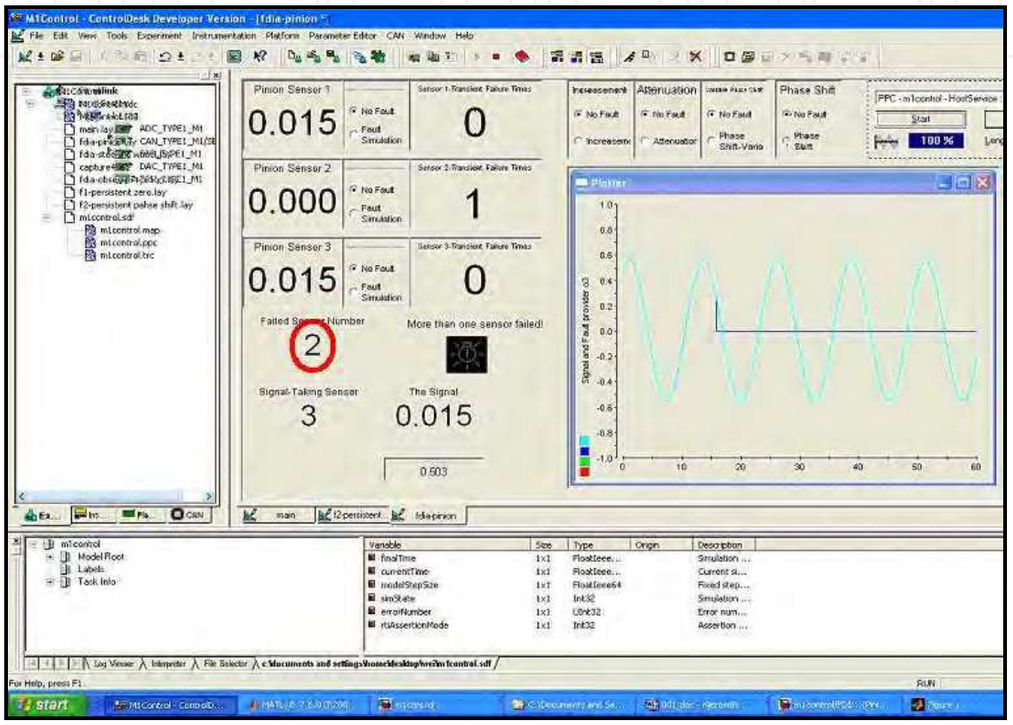

Fig. 20. Fault state reported by FDIA interface for persistent zero with θ_2(circled area)

To test the attenuating-amplitude fault with FDIA interface, the ControlDesk interface creates a decreasing-amplitude fault with a selected θ, and fault detection from FDIA interface is then observed (Figure 22). After the test was completed, it was noted that attenuating-amplitude fault is removed, and then it was observed that the FDIA interface updated the fault state (Figure 23).

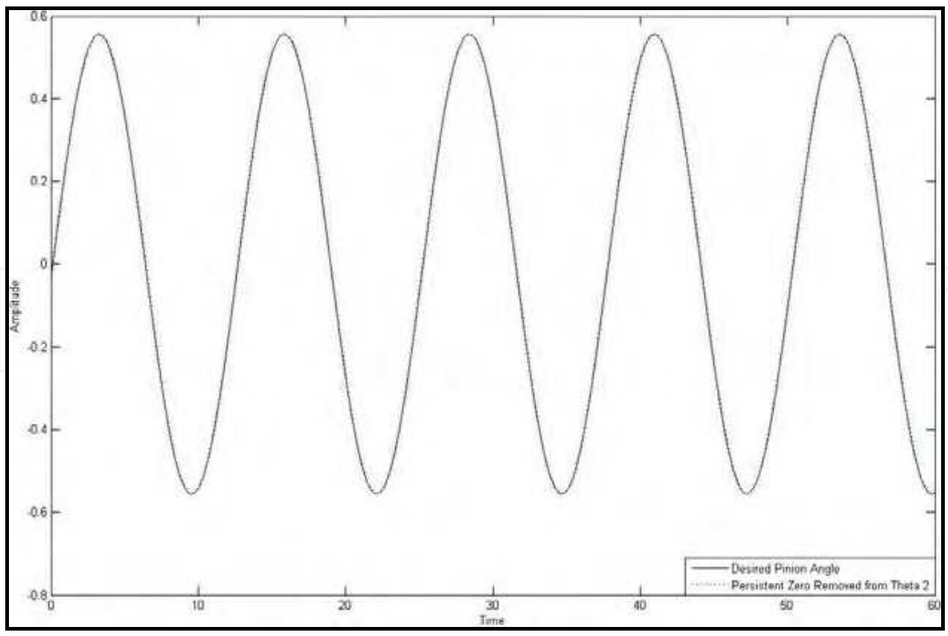

Fig. 21. Normal signal after persistent zero is removed

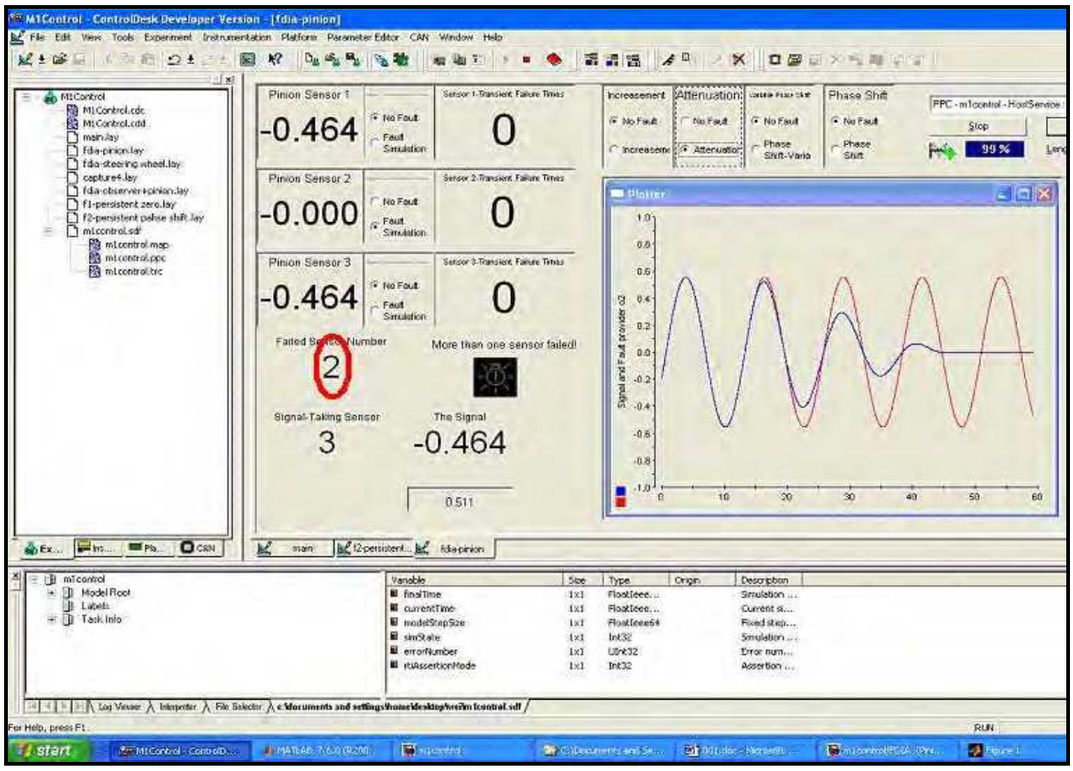

Fig. 22. Fault state reported by FDIA interface for attenuating amplitude with θ_2 (circled area)

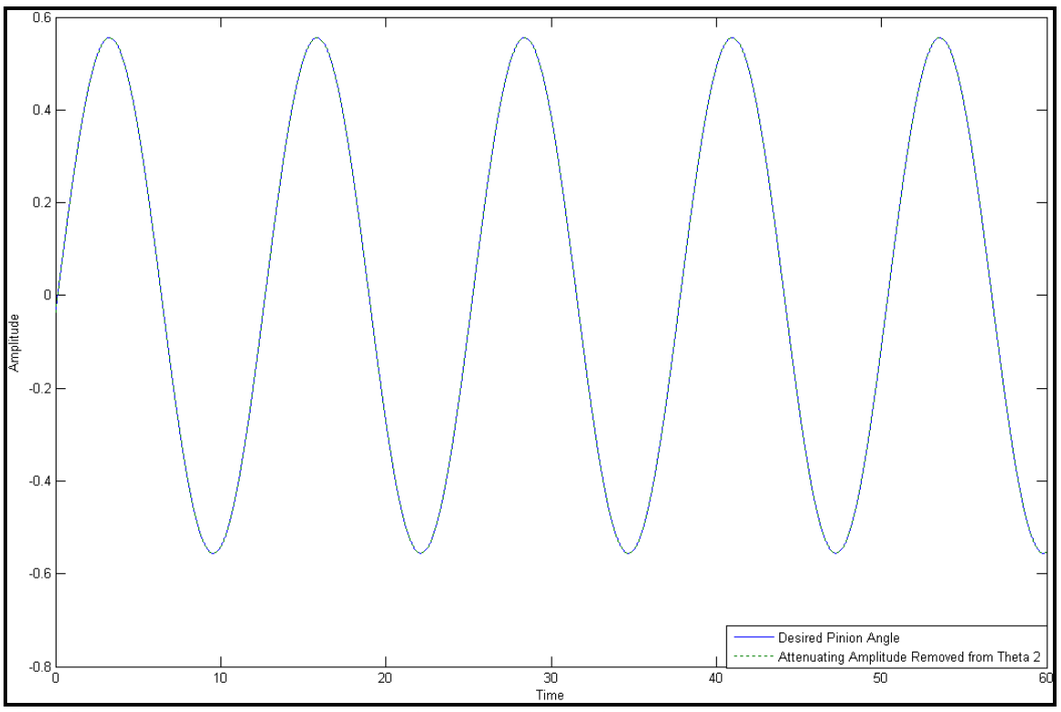

Fig. 23. Normal signal after attenuating amplitude is removed

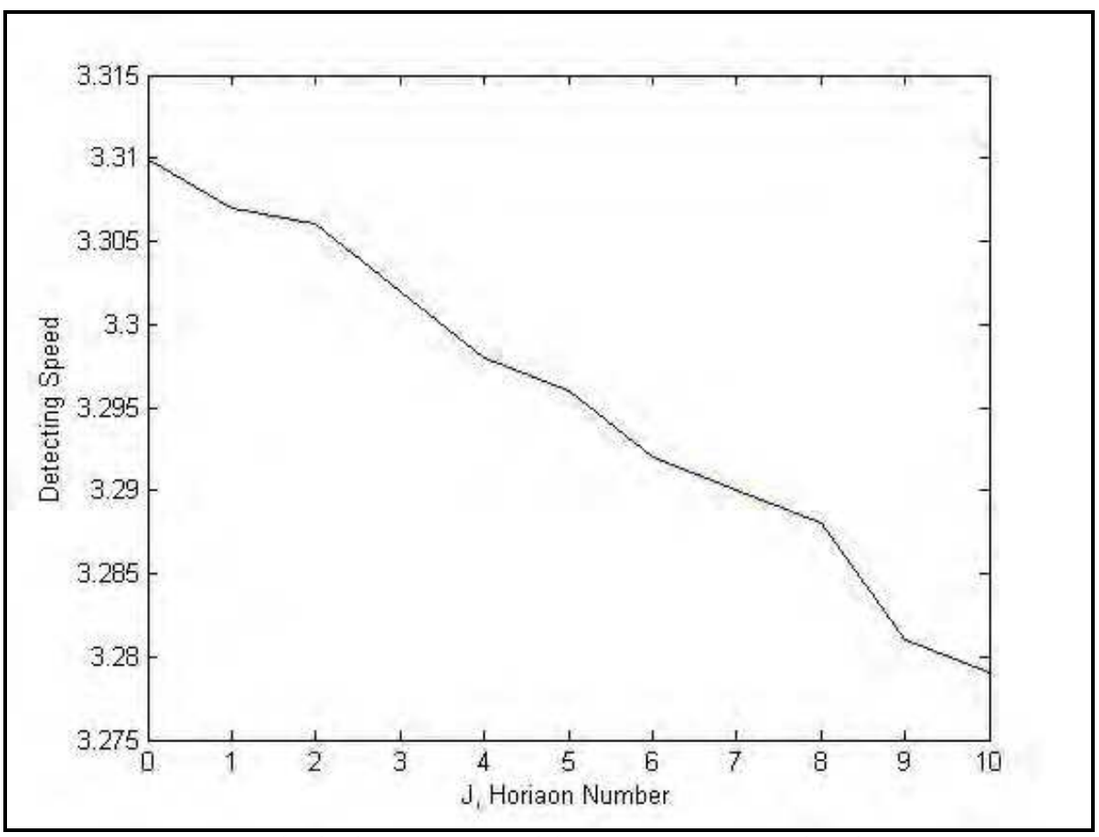

Fig. 24. Horizon versus fault-detecting speed for attenuating amplitude

For this fault case, the impact of the predictor with various prediction horizons is studied. Here the fault detection times are recorded for each prediction horizon selection. After completing the experiment with all selected prediction horizon, the result for attenuating-type fault is shown in Figure 24. It is clear that selecting higher number of horizon makes the FDIA report the detecting speed faster. Again, this result is similar to the ones tested in software-simulation experiment (Hasan & Anwar, 2008).

Similarly, to test variable-phase-shift fault with FDIA interface, the interface creates a variable-phase-shift fault with a selected θ, and fault detection from FDIA interface is then observed (Figure 25). After the test was completed, it is noted that the fault is removed, and then it was observed that the FDIA interface updated fault state. (Figure 26).

Finally, in verifying the performance with varying-phase-shifting-type fault, the result shows that selecting a higher number for prediction horizon makes the FDIA report the detecting speed slower (Figure 27). However, this result is different that the once as tested in (Hasan, 2007) using software-simulation.

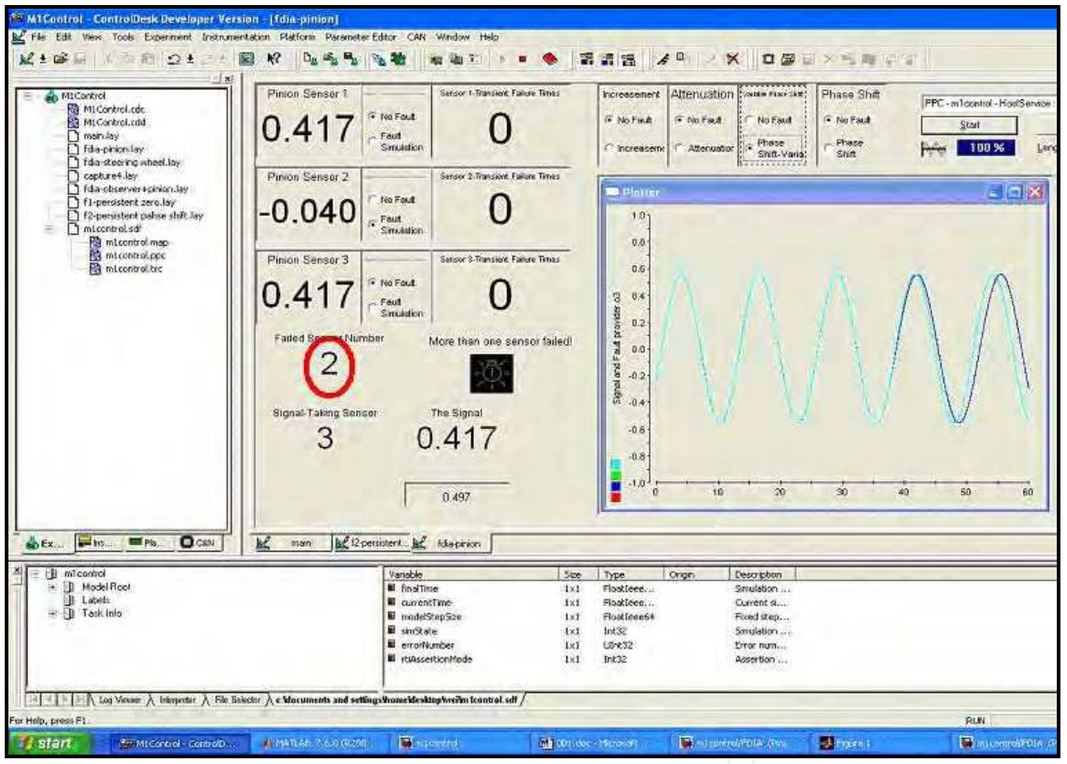

Fig. 25. Fault state reported by FDIA interface for variable phase shift with θ_2 (circled area)

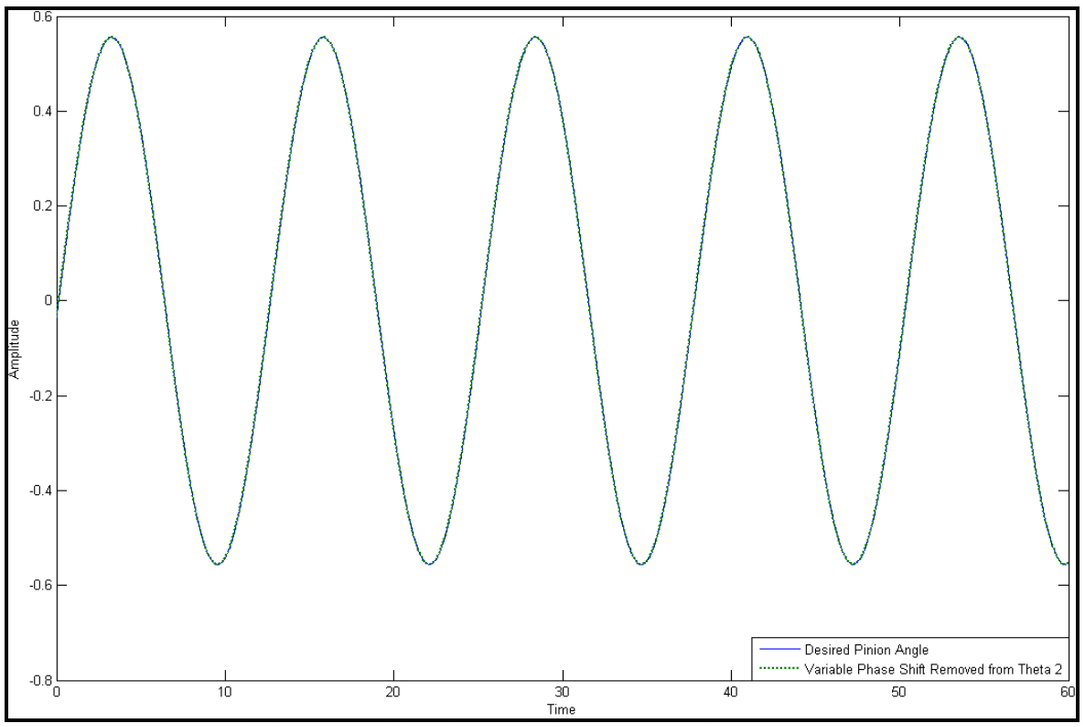

Fig. 26. Normal signal after variable phase shift is removed

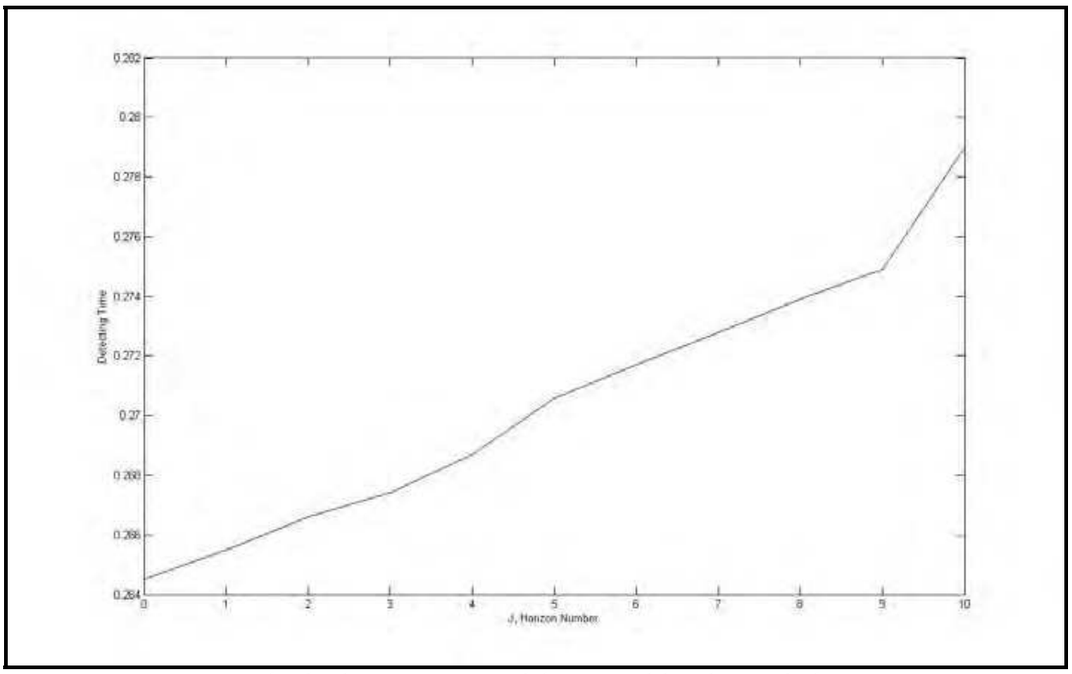

Fig. 27. Horizon versus fault-detecting speed for variable phase shift.

6. Summary and Conclusion

Although automobiles with Steer-by-Wire Control applications are still limited to only prototypes without any near-term commercialization scenario, the potential has been proven by recent research work based on simulation and prototype-based experiments. In this paper, we developed analytical redundancy algorithms using a sliding mode observer and long-range predictor that have been validated on a steer by wire hardware in loop bench. It has also been shown that overall robustness of the SBW system is not sacrificed through the usage of analytical redundancy for sensors along with the designed FDIA algorithm. It is also shown that the faults can be detected faster using the developed analytical redundancy based algorithms for amplifying-type and attenuating-type faults as shown in results section.

However, the fault detection speed for different prediction horizons for the varying phase-shift type faults, the developed analytical redundancy based FDIA algorithms developed for this paper doesn't work as good as in software-simulation. A close look at the result of detecting varying-phase-shift faults reveals that selecting longer prediction horizon for the predictor would reduce the delay time between input and system's output. But, this also means that if there is a fault with gradually changing phase, the system with longer horizon w\would not be able to react or isolate the fault quickly. The reason for the inability of detecting varying-phase-shifting fault faster is that phase of system's response is changed when the input's phase is changed (Figure 28).

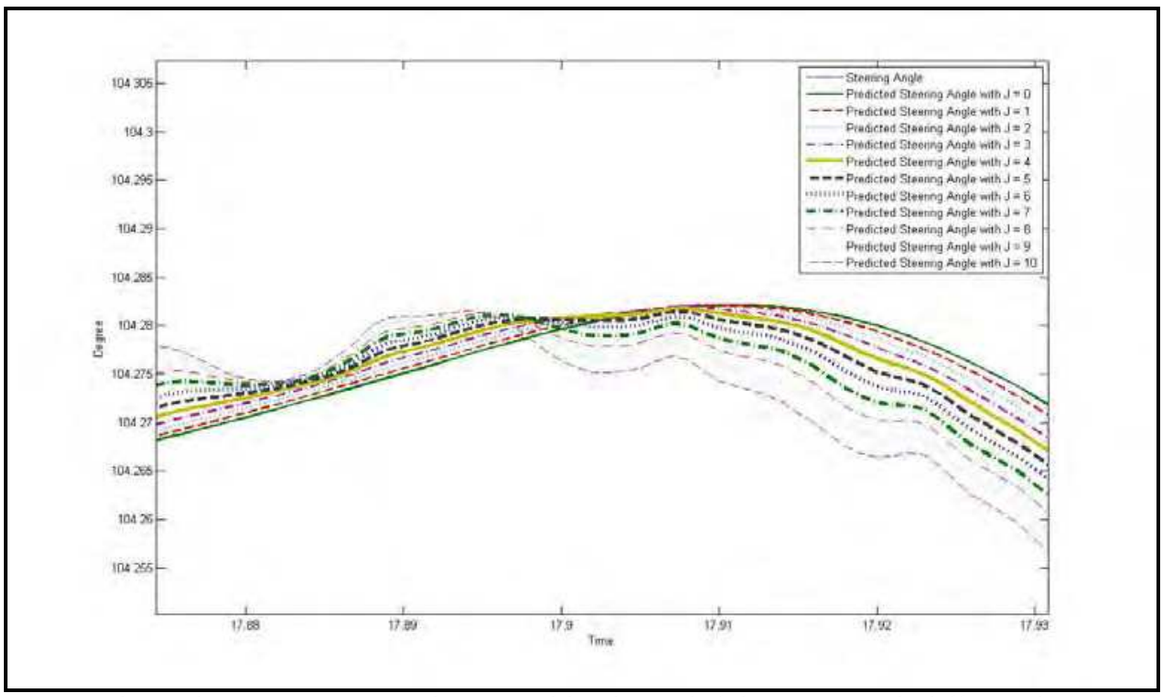

Fig. 28. Phase change occurring when variable phase shift happens with different horizon

Therefore, in order to make possible improvements in solving this problem, an enhanced FDIA algorithm is needed such as tracking the phase for certain duration, etc.

7. References

Anwar, S. (2005). Generalized Predictive Control of Yaw Dynamics of a Hybrid Brake-By-Wire Equipped Vehicle. *Mechatronics*, Vol. 15, November, 2005, pp. 1089-1108.

Anwar, S. & Chen, L. (2006). Analytical Redundancy Based Fault Tolerant Control of a Steer-By-Wire System, *American Society of Mechanical Engineers (ASME), Dynamic System and Control Division (Publication) DSC*, November, 2006, Chicago, IL.

Clarke, D. W., Mohtadi, C., & Tuffs, P. S. (1987). Generalized Predictive Control – Part I: The Basic Algorithm, *Automatica*, Vol. 23, No. 2, 1987, pp. 137-148.

Dong, Y. & Hongyue, Z. (1996). Optimal Design of Robust Analytical Redundancy for a Redundant Strapdown Inertial Navigation System, *Control Engineering Practice*, Vol. 4, No. 12, 1996, pp. 1747-1752.

Gertler, J. (1992). Analytical Redundancy methods in Fault Detection and Isolation, *IFAC Fault Detection, Supervision and Safety for Technical Processes*, Vol. 06, pp. 09-21, 1992, Baden – Baden, Germany.

Hasan, M. S. (2007). Sliding Mode Observer and Long Range Predictive Based Fault Tolerant Control of a Steer-By-Wire Equipped Vehicle, *M.S. Thesis,* Indiana University Purdue University Indianapolis, 2007, pp. 16-39.

Hasan, M. S. & Anwar, S. (2008), Sliding Mode Observer and Long Range Predictive Based Fault Tolerant Control of a Steer-By-Wire Equipped Vehicle, *SAE World Congress and Exposition, Paper No.* 2008-01-0903, April, 2008.

Isermann, R., Schwarz, R. & Stolzl, S. (2002). Fault-Tolerant Drive-By-Wire Systems, *IEEE Control Systems Magazine*, Vol. 22, No. 5, October 2002, pp. 64-81.

Nise, N. S. (1994). *Control Systems Engineering*, 2nd Edition, Addison-Wesley, 1994.

Niu, W. (2009). Fault Tolerant Control Of A Steer By Wire System Using Nonlinear Observer And Predictive Method On Hardware In Loop Bench, *M.S. Thesis,* Indiana University Purdue University Indianapolis, 2009.

Shibahata, Y. (2005). Progress and Future Direction of Chassis Control Technology, *Annual Reviews in Control*, Vol. 29, No. 1, 2005, pp. 151-158.

Stotsky, A. & Hu, X. (1997). Control of Car-like Robots using Sliding Mode Observers for Steering Angle Estimation, *Proceedings of the 36th Conference on Decision & Control*, December, 1997, San Diego, CA, USA.

Suzuki, H., Kawahara, T., Matsumoto, S., Ikeda, Y., Nakagawa, H., & Matsuda, R. (1999). Fault Diagnosis of Space Vehicle Guidance and Control Systems Using Analytical Redundancy, *Space Technology*, Vol. 19, No. 3-4, 1999, pp. 173-178.

Utkin, V., Guldner, J., & Shi, J. (1999). *Sliding Mode Control in Electromechanical Systems*, Taylor & Francis, 1999.

Venkateswaran, N., Siva, M. S., & Goel, P. S. (2002). Analytical Redundancy Based Fault Detection of Gyroscopes in Spacecraft Applications, *Acta Astronautica,* Vol. 50, No. 9, 2002, pp. 535-545.

Fault Detection

Edited by Wei Zhang

ISBN 978-953-307-037-7
Hard cover, 504 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

In this book, a number of innovative fault diagnosis algorithms in recently years are introduced. These methods can detect failures of various types of system effectively, and with a relatively high significance.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen