Research Article

Fault-Tolerant Control of the Road Wheel Subsystem in a Steer-By-Wire System

Bing Zheng¹ and Sohel Anwar²

¹ GE Aviation, Cincinnati, OH 45246, USA

² Department of Mechanical Engineering, Purdue School of Engineering & Technology, IUPUI, Indianapolis, IN 46202, USA

Correspondence should be addressed to Sohel Anwar, soanwar@iupui.edu

Received 22 June 2007; Revised 10 December 2007; Accepted 22 January 2008

Recommended by Maode Ma

This paper describes a fault-tolerant steer-by-wire road wheel control system. With dual motor and dual microcontroller architecture, this system has the capability to tolerate single-point failures without degrading the control system performance. The arbitration bus, mechanical arrangement of motors, and the developed control algorithm allow the system to reconfigure itself automatically in the event of a single-point fault, and assure a smooth reconfiguration process. Both simulation and experimental results illustrate the effectiveness of the proposed fault-tolerant control system.

Copyright © 2008 B. Zheng and S. Anwar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

With electronically controlled steering systems such as steer-by-wire systems, there is a great potential for enhanced safety of the occupants in a vehicle so equipped, due to the fast and precise intervention that is in contrast to the limited reaction time of the driver [1]. At the same time, steer-by-wire systems can provide active steering functionality which can further improve driver's comfort, handling, and robustness of the driver-vehicle system with respect to uncertain operating conditions [2–4]. In active steering, the steer-by-wire control system can control the road wheel steering angle within a permissible range to generate yaw moment either to minimize the over/under steering conditions or to enhance the maneuverability of the vehicle.

In a steer-by-wire system, there is no mechanical connection between steering wheel and road wheel for steer-by-wire, active steering can be easily implemented by inserting an electronic control system between driver's input from steer wheel and road wheel drive system [5, 6]. This electronic control system employing sensors, actuators, communication channels, and microcontrollers ties the steering wheel and road wheels to each other. But due to the fact that there is no mechanical linkage between the steering

wheel and the road wheels for steer-by-wire system, a fault from a sensor, actuator, or microcontroller that form the control system may result in unwanted steering effects if not handled quickly in a fault-tolerant manner. Thus, a fault-tolerant control system is safety critical in steer-by-wire vehicles, requiring highly-dependable sensors and actuators, quick fault detection and identification algorithms, and a means for maintaining reliable vehicle control in the event of a fault.

To overcome the above-mentioned potential fail-safe issues, a dual motor, dual microcontroller control system architecture for steer-by-wire road wheel operation is proposed in this paper. With this architecture, each motor uses a smart motor controller (SMC) to form an inner motor torque control loop. This torque loop controls the motor torque output to track a given torque reference. The microcontroller forms an outer loop, performing road wheel position control. By using two microcontrollers, a master-slave structure is formed. The master carries out all the steer-by-wire road wheel control operation while slave provides the system redundancy. In the event that a fault occurs within the master controller, slave takes over the control operation. If a fault occurs at one local motor loop, the corresponding SMC will shut down the torque

FIGURE 1: Fault-tolerant SBW control system.

FIGURE 2: Master-slave microcontroller arbitration.

control loop. In this case, the steer-by-wire control system switches to single-motor operation automatically without any intervention from master or slave microcontrollers.

This paper is organized as follows. Section 2 discusses the system architecture. Section 3 describes the system dynamic and control system design. Control system implementation is outlined in Section 4. Section 5 illustrates the experimental results, and the conclusion of the paper is contained in Section 6.

2. System Architecture

The overall system architecture is shown in Figure 1. The actuator assembly uses two brushless motors, M1, and M2. The shafts of both motors are connected to a worm gear, which drives a dual-pinion rack-steering mechanism. The worm gear is connected with pinion 1, while pinion 2 is used as mount for an absolute angular sensor. The motor has a built-in resolver for motor-commutation purposes. Additionally, this resolver is used for relative pinion angle measurement. Each motor has its own SMC that carries out the local closed-loop torque control and motor-fault detection.

Two microcontrollers (μ Cs) are used to control the road wheels. Microcontrollers are connected to each other

FIGURE 3: Smart motor controller.

through an arbitration bus as shown in Figure 2. Each microcontroller sends a digital output signal *MySts* (HIGH or LOW) indicating its fault status which is received by the other microcontroller at the input port *StsMon*. When a microcontroller loses power, its *MySts* digital output turns LOW by design via external electrical hardware. If the slave microcontroller *StsMon* port receives an electrical LOW signal, it will reconfigure itself to take over the control functionality. The value for the *MySts* digital output signal from each microcontroller is also an electrical LOW during system initialization after power up or after reset.

3. Control System Design

The proposed fault-tolerant control system consists of two loops: an inner motor torque control loop and an outer pinion angle control loop. The purpose of the inner motor torque loop is to control motor to generate the required torque as specified by a torque reference, while the outer pinion control loop tracks the commanded steer angle with minimal error.

Figure 3 illustrates the block diagram of the inner motor torque control loop. The prefilter converts the desired torque T_d into the desired current components i_{dd} and i_{dq} in d - q coordinates. At the same time, the actual motor current components in d - q coordinates, i_d and i_q , are calculated using measured motor phase currents i_a and i_b as well as motor speed ω . A PI control algorithm is used to calculate the control commands based on the current error between i_{dd} , i_{dq} and i_d , i_q . The command in PWM form is then fed into power electronics (PE) that in turn drives the motor to generate the desired torque. Neglecting the transient response, the equivalent input-output relation of the local torque control loop is described by the following equation:

$$T_e = T_d, \quad (1)$$

where T_e is motor output torque. The above assumption is reasonable since an electrically actuated steering system has an order of magnitude faster response time than that of a hydraulically actuated steering system which is much shorter than the human reaction time.

It is assumed that the two motor control loops have the similar dynamic characteristics and that motor inertia and friction are negligible when compared with the road wheel

FIGURE 4: Equivalent control system.

assembly. The differential equation describing the pinion angle dynamics is then as follows [7]:

$$J\ddot{\theta} + b\dot{\theta} + F_c \operatorname{sgn} \dot{\theta} + k_a \tau_a = mu, \quad (2)$$

where θ is the pinion angle, J is the moment of inertia of the assembly, b is the viscous damping, F_c is the Coulomb friction, k_a is the scaling factor, τ_a is the tire self-aligning moment, u is the torque output for a single motor, and m is the number of motors.

In the above equation, the viscous damping term represents the internal friction within the tire material and other moving steering components whereas the Coulomb friction term represents the sliding friction at the road-tire interface. The scaling factor for the self aligning moment accounts for the portion of the moment that is transmitted via the steering mechanism. To design the outer control loop, the dynamic equation (2) is rearranged as follows:

$$J\ddot{\theta} + b\dot{\theta} = mu - \Delta(\theta), \quad (3)$$

where

$$\Delta(\theta) = F_c \operatorname{sgn} \dot{\theta} + k_a \tau_a, \quad (4)$$

$\Delta(\theta)$ is considered as a predictable disturbance.

Based on (3), the outer control loop is partitioned into two parts: a nominal feedback part and a feedforward part. The nominal feedback loop is designed based on the following equation:

$$J\ddot{\theta} + b\dot{\theta} = mu. \quad (5)$$

It can be easily seen that this is the pinion angle system dynamics model (3) with the disturbance portion truncated. A second-order lead-lag compensator as described by (6) is used for the nominal feedback loop. The compensator parameters a_1 , a_2 , b_0 , b_1 , b_2 , and g are selected based on the nominal model using bode plots. The compensated nominal system has a minimal phase margin of 45 degrees and 10 dB gain margin for both single motor and dual motor operations.

$$G_c(s) = g \frac{s^2 + a_1 s + a_2}{b_0 s^2 + b_1 s + b_2}. \quad (6)$$

The purpose of feedforward loop is to compensate the disturbance term as shown in (4). For the feedforward compensation, F_c is obtained through an experiment using system-identification techniques [8, 9]. The self-aligning

FIGURE 5: Control system implementation.

moment in (4) is directly approximated by the following equation as a function of tire-slip angle [10, 11]:

$$\tau_a = (t_p + t_m) F_{y,f}(\alpha_f) \approx (t_p + t_m) C_{y,f} \alpha_f, \quad (7)$$

where t_p is the pneumatic trail, t_m is the mechanical trail, $F_{y,f}$ is the front tire lateral force, $C_{y,f}$ is the front tire cornering coefficient, and α_f is the front tire slip angle. Using a single-track vehicle model or bicycle model [12], the above self-aligning moment can be related to the pinion angle, vehicle body side slip angle, and vehicle speed [11].

We define

$$f(t) = F_c \operatorname{sgn} \dot{\theta} + k_a \tau_a = \tau_f + k_a(t_p + t_m) C_{y,f} \alpha_f, \quad (8)$$

where τ_f is considered a disturbance torque. We also note that [11]

$$\alpha_f = \beta + \frac{ar}{V} - \theta, \quad (9)$$

where β is the vehicle body side slip angle, r the yaw rate, V vehicle speed, and a the distance of front axle from c.g. Vehicle speed is assumed to be constant here.

Then the overall pinion angle control signal can be represented by the following equation:

$$u = G_c(s)(\theta_d - \theta) + F(s)\theta_d, \quad (10)$$

where $F(s)\theta_d$ is designed such that it approximates the function $f(t)$ as defined in (8).

The block diagram of the overall closed-loop steer-by-wire road wheel angle control system is illustrated in Figure 4. In the diagram, P represents the plant dynamics of the road wheel assembly as well as the vehicle dynamics, θ_d the desired road wheel angle. It is noted that the motor inner control loop dynamics has been neglected since it has much faster response characteristics when compared with that of the outer angle control loop.

4. Fault-Tolerant Algorithm Design and Implementation

Figure 5 illustrates the implementation of the fault tolerant controller. The inner motor control loop is implemented in SMC and outer road wheel angle control loop is in microcontrollers. Both SMCs share the same program code and both microcontrollers share the same program code except that one has a longer initialization time than the other.

(a) SIMULINK model of the fault-tolerant controller

(b) Motor 1 current with fault in motor 2

(c) Motor 2 current with fault

(d) Road wheel angle after motor 2 current fault

FIGURE 6

FIGURE 7: Steer-by-wire system bench at Chassis Advanced Technology Lab.

4.1. Master-Slave Operation

After initialization, each microcontroller checks the *MySts* signal output line from the other through the *StsMon* digital input port as shown in Figure 2. The microcontroller that reads a LOW signal from others becomes the master. The master then pulls its *MySts* digital output line HIGH, and proceeds with control operation. The microcontroller that reads a HIGH signal becomes the slave. As a slave, the microcontroller keeps its *MySts* digital output LOW. The slave also calculates its own control command, and compares the calculated result with master's result received through the CAN communication network. Whenever the difference in the comparison of the two control commands is larger than a predefined threshold, the slave will use the control command from the master to reset its own calculation. At the same time, the slave controller continuously monitors the *MySts* digital output from the master. As soon as *MySts* line from the master is detected LOW for a specified time period, the slave will switch to master operation mode. As master, it will toggle its *MySts* digital output line HIGH and proceeds with outer angle control operation.

Each microcontroller is monitored by its own internal *WatchDog*. When a fault occurs within the microcontroller, and the fault is beyond the capability of the exception handler, the *WatchDog* will reset the μC , which will bring *MySts* to LOW.

4.2. CAN Bus Operation

The messages sent out from the SMCs include resolver values and local diagnostic information. The resolver values are sent with a 1 millisecond period for closed-loop control purpose. The diagnostic information is sent at 10 milliseconds period, and is used by microcontrollers to evaluate the health of the motor control loops.

Only the master sends control command to the SMCs through CAN buses during normal operation. Both SMCs receive the same control commands. The slave also receives the control command from the master and uses it to

(a) Phase current of motor 1

(b) Phase current of motor 2

(c) Pinion angle: actual versus desired

FIGURE 8

FIGURE 9

synchronize its own control calculation with master. If master becomes faulty, the slave takes over the control task by sending its control command to the SMCs through the CAN buses.

Whether the control signal is sent by master or slave, the CAN message ID for the control command is the same. There is no message ID conflict on the bus since master and slave never send control commands at the same time.

4.3. Sensor Operations

All three angular sensors described in the system architecture, one absolute, and two relative (resolver), are used to measure pinion angle. Upon power on, the absolute sensor value is captured. This captured value is then added to the resolver signal that has a zero output right after power on. With this added value, the resolver functions as an absolute

angular sensor whose output can be represented by the following equation:

$$\theta_{\text{cal}} = \theta_{\text{rel}} + \theta_{\text{abs0}}, \quad (11)$$

where θ_{rel} is a raw resolver signal value, θ_{abs0} is a captured absolute sensor value upon the power on, and θ_{cal} is calibrated resolver sensor value.

During normal operation, the readings from all three sensors are constantly compared among each other. A weighted averaging algorithm is used to process the readings as described by the following equation:

$$\theta_{\text{flt}} = \alpha\theta_{\text{cal1}} + \beta\theta_{\text{cal2}} + \gamma\theta_{\text{abs}}, \quad (12)$$

where θ_{flt} is the filtered pinion angle, θ_{cal1} is the calibrated pinion angle from resolver 1, θ_{cal2} is the calibrated pinion angle from resolver 2, θ_{abs} is the measured pinion angle from absolute sensor.

The value of α , β , and γ is between 0 and 1. At the same time, the values of α , β , and γ are constrained by the following equation:

$$\alpha + \beta + \gamma = 1. \quad (13)$$

In most cases, more weights are added to the calibrated resolver readings due to its higher resolution and linearity. When the difference among the sensor readings is larger than a specified threshold, the reading from the sensor that has the largest difference from the others will be omitted, that is, the corresponding coefficient from the above equation becomes zero. Nevertheless, this sensor reading will be used again for the weighted averaging during the next calculating cycle. But, if this large difference lasts longer than a specified period, the sensor will be considered faulty and the corresponding readings will be discarded for the rest of the operation.

5. Modeling and Simulation

The proposed system was modeled in MATLAB / SIMULINK environment (Figure 6(a)). Simulation run was performed with fault injected in motor 2 current to evaluate the fault-tolerant algorithm presented in this paper. The desired road wheel angle was a sinusoidal command with a frequency of 0.2 Hz. A fault was injected in motor 2 current by setting the motor input current to zero at 6 seconds (Figure 6(c)). The fault management algorithm detects the motor 2 fault and switches the control mode from dual motor to single motor mode by sending the full current command to motor 1. As a result, motor 1 current increased to ensure seamless performance of the steer by wire system, as illustrated in Figures 6(b) and 6(d).

6. Experimental Results

After establishing the effectiveness of the proposed fault-tolerant control algorithm in simulation, an experimental validation of the algorithm was performed. The control system developed was implemented in a steer-by-wire test bench at Visteon Chassis Advanced Technology department (Figure 7). The fault-tolerance test of the system was performed on the road wheel control subsystem and the test results are illustrated in Figures 8 and 9.

A reference pinion angle signal was applied to the control system reference input. This signal has a frequency of 0.115 Hz or 0.2 Hz, and amplitude of 100 or 125 degrees. At the same time, an active load of 8896 Newton is applied to the tie rod of the assembly. This active load has a waveform varying from a constant to a 5 Hz sinusoid shape or vice versa, depending on the stage of the test.

Figure 8 illustrates the test result for the motor loop fault-tolerance capability. Figures 8(a) and 8(b) show the phase currents of the motor, and Figure 8(c) shows the comparison between measured and desired pinion angle. During the test, fault was injected into the inner loop of motor 2. The corresponding SMC detects the fault via a diagnostic code and shuts the inner loop down. The overall control system switched automatically to the single motor control mode.

It can be noticed from Figure 8(a) that there was a phase current increase for the motor 1 (after about 7 seconds) during the switchover due to the shutdown of motor 2 which has zero phase current after switchover. But no pinion angle tracking degradation was observed in the captured pinion angle data which is illustrated in Figure 8(c). With this test, the active load is constant.

The result for microcontroller switchover is shown in Figure 9. As in Figure 8, the phase currents and the comparison between measured and desired pinion angles are shown in the corresponding Figures 9(a), 9(b), and 9(c). A fault is injected into master microcontroller during the test. The slave controller detects the *MySfs* change from master immediately, and switches into master operating mode right after the detection. Figure 9(d) shows the status change from the perspective of the slave microcontroller. Again, no pinion angle tracking degradation was observed in the captured pinion angle data which is illustrated in Figure 9(c). The high frequency oscillation appeared on the measured pinion angle curve reflects the fact that the active load is now in its 5 Hz sinusoid wave form.

7. Conclusion

The proposed steer-by-wire road wheel control architecture has shown the benefit of fault-tolerance capability. In the event of single point fault, either caused by one motor control loop, or by one microcontroller, the system continues to provide the road wheel steering functions without degradation of the command-tracking performance, as evidenced by the simulation and experimental results.

Since the total power needed for steering the vehicle can be contributed by both motors, the maximum power output of each motor can be reduced by half, which in turn allows the usage of the power electronics with lower power consumption. This will reduce the complexity of the power electronics design with respect to the noise, EMI/EMC, as well as thermal release.

Finally, since both microcontrollers share the same program code, the cost related with the software development, debug, and maintenance can be greatly reduced.

Acknowledgment

The authors wish to acknowledge Mr. Matt Mikesell, senior manager of Chassis Advanced Technology at Visteon Corporation, for his review comments and suggestions on the paper.

References

- [1] J. Ackermann and T. Bunte, "Automatic car steering control bridges over the driver reaction time," *Kybernetika*, vol. 33, no. 1, pp. 61–74, 1997.
- [2] J. Kasselmann and T. Keranen, "Adaptive steering," *Bendix Technical Journal*, vol. 2, pp. 26–35, 1969.
- [3] J. Ackermann, "Yaw disturbance attenuation by robust decoupling of car steering," in *Proceedings of the 13th World IFAC Congress (IFAC '96)*, San Francisco, Calif, USA, July 1996.

- [4] K. Huh and J. Kim, "Active steering control based on the estimated tire forces," *Journal of Dynamic Systems, Measurement and Control*, vol. 123, no. 3, pp. 505–511, 2001.
- [5] M. Segawa, K. Nishizaki, and S. Nakano, "A Study of vehicle stability control by steer-by-wire system," in *Proceeding of the 5th International Symposium on Advanced Vehicle Control (AVEC '00)*, Ann Arbor, Mich, USA, August 2000.
- [6] B. Zheng, P. Oh, and B. Lenart, "Active steering control with front wheel steering," in *Proceedings of the American Control Conference (ACC '04)*, vol. 2, pp. 1475–1480, Boston, Mass, USA, June-July 2004.
- [7] B. Zheng, C. Altemare, and S. Anwar, "Fault tolerant steer-by-wire road wheel control system," in *Proceedings of the American Control Conference (ACC '05)*, vol. 3, pp. 1619–1624, Portland, Ore, USA, June 2005.
- [8] B. Zheng and N.-S. Hsu, "Single-input-single-output system identification via block-pulse functions," *International Journal of Systems Science*, vol. 13, no. 6, pp. 697–702, 1982.
- [9] B. Zheng and N.-S. Hsu, "Analysis and parameter estimation of bilinear system via block-pulse function," *International Journal of Control*, vol. 36, no. 1, pp. 53–65, 1982.
- [10] E. Bakker, L. Nyborg, and H. Pacejka, "Tyre modelling for use in vehicle dynamics studies," Tech. Rep. 870421, Society of Automotive Engineers, Warrendale, Pa, USA, 1987.
- [11] S. Anwar and L. Chen, "An analytical redundancy-based fault detection and isolation algorithm for a road-wheel control subsystem in a steer-by-wire system," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, part 2, pp. 2859–2869, 2007.
- [12] P. Riekert and T. E. Schunck, "Zur Fahrmechanik des gum-mibereiften Kraftfahrzeugs," *Ingenieur Archiv*, vol. 11, no. 3, pp. 210–224, 1940.