Electronically Controlled Steering

Customer: Mr. David Agnew, Continental Automotive Systems, Advanced Engineering

Purpose: Develop software for electronically controlled steering control module.

System overview:

This steering system is a modified version of “rack and pinion” steering. (See books on automotive mechanics for a description of “rack and pinion” steering.) A DC motor provides additional assisted torque to the steering rack through a pinion mounted on the motor shaft. The driver’s steering commands are input into an electronic control module via two independent torque sensors that are on mounted steering column. The input torque is verified and used to establish the required “assisted torque” value. This is done by mapping the desired input torque and vehicle speed to the required assisted torque. A pulse-width-modulated signal is used to drive the DC motor which supplies the assisted torque for steering. A direction command signal is utilized to determine clockwise or counter clockwise torque assist.

Competitive Product information:
TRW, Delphi Automotive, and Mitsubishi Electric. See web sites for associated information.

Requirements:

Torque control:
Torque is equal to the torque constant of the motor times the current of the motor.

A Proportional Integral and Derivative (PID) control shall be implemented in software to control the current of the motor.

The motor current shall be read from a current mirror on the electrical power drivers to the motor.

A current no greater than 60 amperes shall be allowed through the DC motor.

The torque control shall not be utilized if the battery voltage is less than 11 volts.

Convert the input torque to assisted torque by using interpolation between a set of “U” shaped curves. One curve exists for each of the following: 0, 20, 40, 60 80, 100, and 120 miles per hour. For speeds up to 160 miles per hour use the 120 mph curve.
The curve at zero miles per hour provides the most assist. At higher speeds the road to wheel friction is lower and less assisted torque is required for turning. At lower speeds the dead band is small i.e. near zero input torque produce some torque assist. At higher speeds the dead band is wider i.e. the x axis around zero requires more input torque before providing assisted torque.

The PWM signal to the DC motor must run at a rate of 15 kilohertz.

*Update Rate:*
Refresh the torque motor drive commands data from updated calculations at a rate no slower than every 500 microseconds. (This is required for proper control)

*Sensor Verification*
Verify that the torque sensors values correspond within 5% of each other. Perform this test at a rate not less than every 10 milliseconds. Use the minimum value of the two torque sensors for the calculation.

*Fault handling*
Should a failure condition be detected, measure the existing motor current value then ramp the current back to zero over a 2 second period. (Failures to do this will cause the steering wheel to be harshly snapped from the driver's hand and could result in injury.)

An internal “watch dog” timer, an external watch dog counter and interrupt counter shall be used to insure all control loops and background software are operational.

Once a failure has occurred the system shall remain in the failure mode until the system is powered down and re-initialized.

Operational diagnostics shall be repeated at a rate no less then every 10 milliseconds. (A failure to do so within this time could be sufficient to cause undesired vehicle lane changes and uncontrolled steering. This could result in a deadly situation.)

Where possible, diagnostics shall be tested in real time to show that sub-systems are functional. At a minimum the following are required:

- external watch dog verification
- torque value verification
- ram memory verification
- flash memory (program and calibrations) verification
If a failure occurs a Malfunction Indicator Light (MIL) shall be illuminated. The MIL light will also be illuminated for 3 seconds on power up.

**Serial Communications**

A fault status shall be sent on a CAN communications link once every 60ms:

- **Message ID 110**
  - byte 1 - Functional Group of failure
  - byte 2 - Failure Mode Identifier
  - byte 3 - duration of failure
  - bytes 7 to 4 - running timer values (Low Significant Byte is 4)
  - bytes 8-15 – reserved (set to a value of zero)

The message shall be sent to with all zeros if the steering system is totally functional. Identification for the functional groups and failure modes shall be set by the developing software team.

**Re-programming:**

The module shall provide re-programming ability over the CAN data link. The format, protocol and security shall be defined by the software developers.

**Scenarios**

**Scenario 1:** The system is fully working as described previously. Vehicle is idling in a parking lot at 0mph, and the driver is currently depressing the brake pedal with the gearshift in drive. As the driver is depressing the pedal, the steering wheel starts being pulled in a counter-clockwise fashion of its own accord. There must be some mechanism to disengage the electronically controlled steering in the case of a severe error.

**Scenario 2:** The system is fully working as described previously. Vehicle is travelling at 75mph on the freeway. A fault occurs in the wiring, and the system is unable to provide the correct amount of torque to the steering wheel.

**Scenario 3:** Since the CAN bus is essentially a free-for-all, it is assumed that data may sometimes become corrupted. Some mechanism must be introduced to deal with the fact that received data may be incorrect.