Requirements Specification (SRS)
Project Squeaky Wheel

Authors: Team Autobots: Kevin Shreve – Project Manager, Seung-Min Kim – Project Facilitator, Andrew Crouch – Domain Expert/Customer Liaison, Cory Harter – Artifacts Manager

Customer: Continental Motors

Instructor: Dr. Betty HC Cheng

Table of Contents

Table of Contents

1.0 Introduction........................................................................................................... i
  1.1. Purpose.............................................................................................................. 1
  1.2. Scope............................................................................................................... 1
  1.3. Definitions, Acronyms, and Abbreviations-------------------------------------- 2
  1.4. Organization ................................................................................................... 3

2.0. Overall Description........................................................................................... 4
  2.1. Product Perspective....................................................................................... 4
  2.2. Product Functions.......................................................................................... 4
  2.3. User Characteristics...................................................................................... 4
  2.4. Constraints..................................................................................................... 5
  2.5. Assumptions and Dependencies.................................................................... 5
  2.6. Proportioning of Requirements...................................................................... 5

3.0. Specific Requirements ..................................................................................... 6
  3.1. System Use..................................................................................................... 6
  3.2. Steering Wheel Angle to Wheel Angle.......................................................... 6
  3.3. External Input................................................................................................ 6
  3.4. Load Torque.................................................................................................. 6
  3.5. Proportional Integral and Derivative Control................................................. 7
  3.6. Updating the Motor Drive............................................................................. 7

4.0. Modeling Requirements..................................................................................... 8
  4.1. Use Case Diagram.......................................................................................... 8
  4.2. Class Diagram................................................................................................ 11
  4.3. Sequence Diagram....................................................................................... 13
  4.4. State Diagrams.............................................................................................. 14

5.0 Prototype ........................................................................................................... 16
  5.1. How to Run Prototype................................................................................... 16
  5.2. Sample Scenarios.......................................................................................... 17

6.0 References ......................................................................................................... 19

7.0 Point of Contact.................................................................................................. 19

Template based on IEEE Std 830-1998 for SRS. Modifications (content and ordering of information) have been made by Betty H.C. Cheng, Michigan State University (chengb at chengb.cse.msu.edu)
1 Introduction

This document states and describes the requirements necessary to model and build a basic steer-by-wire system and also describes the use of a web-based prototype of the system.

1.1 Purpose

The purpose of this document is to present a detailed description of a steer-by-wire control system. It describes the purpose and features of the system, how the system behaves, what the system accomplishes, and the constraints under which the system must operate. This document is intended for the eventual developers and stakeholders of such a system, but first for Mr. David Agnew and Mr. Jeremy McClain of Continental Automotive.

1.2 Scope

This software system is a proportional-integral-derivative (PID) controller for use in a steer-by-wire system. Steer-by-wire is a system that removes the traditional physical link between steering wheel and road wheels, and instead uses a de-coupled wheel connected by wire to an actuator that turns the road wheels via an angle request. This system is designed to provide proper conversion of a steering angle request into a torque request for an actuator to provide proper steering of a vehicle. Steer-by-wire is a concept still in its early stages and this system is an early prototype to help confirm the viability of such systems in the future. A few benefits of such a system include: simplified interior design via removal of the steering column and reduced fuel consumption due to removal of hydraulic steering systems.
1.3 Definitions, acronyms, and abbreviations

A brief understanding of the following terms is recommended for comprehension of this document.

- **Actuator** - Generally anything that moves based on a command from a software system. In this case, it refers to the motor providing the torque necessary to turn the wheels of a vehicle to steer.
- **Angle Request** - The angle requested by an external source to be received by the PID.
- **Class Diagram** - Shows all the system’s classes, including their attributes, operations, and relationships (and their multiplicities).
- **Closed Loop system** - A feedback system that does not rely on outside sources to have a corrective course of action, all the necessary inputs are within the loop.
- **PID** - Proportional-Integral-Derivative controller. This is a system used to calculate an error adjusted request to an actuator to minimize error and maximize response time.
- **Sequence Diagram** - Presents a graphical representation of detailed interactions between objects in a specific scenario.
- **State Diagram** - Shows the states of a system and the events that occur during transition between states in a graphical format.
- **SBW** - Steer-by-wire
- **SWA** - Steering Wheel Angle
- **Torque Request** - The request for torque generated by the PID to be received by the actuator. Torque requests for this system are measured in Newton-meters (Nm)
• **UML** - Unified Modeling Language - used to model systems, and therefore glean a better understanding of the system than can be done from text alone.

• **Use Case Diagram** - Presents a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.

### 1.4 Organization

This document is comprised of seven sections: Introduction, Overall Description, Specific Requirements, Modeling Requirements, Prototype, References, and Point of Contact. The next section of this document, the Overall Description section, provides an overview of the functionality of the product. It describes its informal requirements and helps to create a context for the specific requirements presented in the third section.

The Specific Requirements section gives a detailed list of requirements software developers would need to consider when developing the system.

The Modeling Requirements section models the requirements in section three through a series of Unified Modeling Language (UML) diagrams, to further convey the form that the system should take.

The Prototype section covers the web-based prototype that our team has developed. It describes in detail how to use the prototype, where to access it, and what it conveys.

The References section contains a list of references used in the creation of this document and a reference to our team’s website.

The final section is a point of contact for questions and concerns regarding this document.
2 Overall Description
This section contains the overall description of the project. More specifically, it states the perspective of the system, the constraints of the system, and assumptions made about the system and its users.

2.1 Product Perspective
The system is an electronic steering controller. A de-coupled steering wheel provides an angle request to a PID controller that then provides an error-adjusted torque request to the actuator that turns the wheels of the vehicle.

2.2 Product Functions
The following points summarize the major functions that the software will perform.

- The steering angle request comes from an external source.
- Input values are used to calculate the necessary assisted torque value.
- Apply the needed amount of torque to the motor.
- Wheel angles are adjusted by the motor based on the torque value.

2.3 User Characteristics
The system is autonomous and no user input is required. Therefore, there is no learning curve to begin using this system; it is automatically engaged when the vehicle is in the speed range from -10 to 20 mph.
2.4 Constraints
The following points summarize properties that must be satisfied in order for the system to perform properly.

- The system is only active in the range of -10 to 20 miles per hour.
- System must recalculate values every 500 µs.
- The assisted torque is always within the range of -10 to 10 Nm.

2.5 Assumptions and Dependencies
This system is only for use on non-public roads. It is intended only to be a slow speed system. It controls all necessary parts of a motor vehicle based on external input that throughout requirements analysis we have assumed to be “clean” input: input with no required error checking.

2.6 Appropriation of Requirements
The original specifications required the use of watchdog timers in the system process. Since then, the client has confirmed that watchdog timers are outside of the scope of the project and can be disregarded.
3 Specific Requirements

This section enumerates the requirements of the system.

3.1 System Use

The system is intended to be used on non-public roads where there is external input that may be used to guide the car along a desired path. The expected vehicle speed range should lie between -10 and 20 miles per hour.

3.2 Steering Wheel Angle to Wheel Angle

The steering angle request to wheel angle is completed by a simple conversion. For each 720°, the wheels turn 45° in the direction of the turn. So by taking the amount that the steering wheel has changed and multiplying it by 45° / 720°, this maps the SWA to the wheel angle.

3.3 External Input

The system receives a steering request angle from an external source. The specific source is not within the scope of this document.

3.4 Load Torque

The load torque calculation is a function of the inertia, damping, spring and friction of the tires on the road and is as follows:

\[ \text{Load Torque} = \text{inertia} \times \dot{\theta} + \text{damping} \times \ddot{\theta} + \text{spring} \times \theta + \text{friction} \]
3.5 Proportional Integral and Derivative (PID) control

PID control is a system that calculates an error-adjusted torque request based on the current wheel angle and the steering request angle. This is necessary due to the imperfect nature of electric motor control.

3.6 Updating the Motor Drive

The motor drive command is updated from the calculations done by the PID and must take place every 500µs, to ensure proper control is maintained over the vehicle.
4 Modeling Requirements

This section specifies the bridge between the application domain and the machine domain using a variety of visual diagrams.

4.1 Use Case Diagram

Use Case Diagram - A use case diagram allows for each scenario to be displayed via a diagram, and followed by brief description of each use case. This shows the interaction between specified actors and the dependencies between those. Each individual use case is represented in a circle. The actors to the system are represented by stick figures. Use cases are connected to each other, each may be either dependent on another, extend the functionality, or be included. If a use case is included it means that this use case is used in several parts of the system and this is for code reuse.

Figure 4.1.1 – This diagram illustrates how the external source sends the steering request angle that is used in the torque calculation.
<table>
<thead>
<tr>
<th>Use Case:</th>
<th>Send Request Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td>External Source</td>
</tr>
<tr>
<td>Description:</td>
<td>Send the requested angle to the system, this is a combination</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary</td>
</tr>
<tr>
<td>Includes:</td>
<td>Calculate Torque</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>3.3</td>
</tr>
<tr>
<td>Dependent Use Cases:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case:</th>
<th>Measure Current Vehicle Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td>Retrieves the current angle of the wheels for the car, this enables the system to determine where the wheels need to be based on where they are currently.</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary</td>
</tr>
<tr>
<td>Includes:</td>
<td>None</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>3.1, 3.2</td>
</tr>
<tr>
<td>Dependent Use Cases:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Case:</th>
<th>Refresh Motor Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors:</td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td>Sends a signal based on a timer refresh of every 500 μs.</td>
</tr>
<tr>
<td>Type:</td>
<td>Secondary</td>
</tr>
<tr>
<td>Includes:</td>
<td>None</td>
</tr>
<tr>
<td>Extends:</td>
<td>Calculate Torque</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>3.6</td>
</tr>
<tr>
<td>Dependent Use Cases:</td>
<td></td>
</tr>
<tr>
<td>Use Case:</td>
<td>Calculate Torque</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Actors:</td>
<td>Calculates the necessary torque (bi-directionally) that the system must output in order for the steering to be adjusted to the desired angle. Also responds to the amount of friction the system is currently experiencing.</td>
</tr>
<tr>
<td>Type:</td>
<td>Primary</td>
</tr>
<tr>
<td>Includes:</td>
<td>Request Torque, Measure Current Vehicle Angle</td>
</tr>
<tr>
<td>Extends:</td>
<td>None</td>
</tr>
<tr>
<td>Cross-refs:</td>
<td>3.4, 3.5</td>
</tr>
<tr>
<td>Dependent Use Cases:</td>
<td></td>
</tr>
</tbody>
</table>
4.2 Class Diagram

Class Diagram – A class diagram shows exactly each object and all of the operations it can perform as well as the attributes. They also show the relationships it has with other objects in the system, as well as multiplicities. A class is represented as a rectangle and is divided into smaller sections. The First section is reserved for the class name, the middle is for attributes, and functions are placed in the third section. The classes are linked together and each link describes how the classes interact.

![Class Diagram](image)

**Figure 4.2.1** – This diagram illustrates how the objects in the system can interact with each other, and any values that are maintained in the system.

<table>
<thead>
<tr>
<th>Class name</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>This is the framework that all of the other classes utilize to make the system work. The system takes care of the calculations, and then sending a request to apply torque.</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>N/A</td>
</tr>
<tr>
<td>Attributes</td>
<td>N/A</td>
</tr>
<tr>
<td>Functions</td>
<td></td>
</tr>
<tr>
<td>-MeasureAngle()</td>
<td>Measures the current angle of the wheels.</td>
</tr>
<tr>
<td>-CalculateTorque()</td>
<td>Calculates an adjusted torque request value via a PID.</td>
</tr>
<tr>
<td>-Refresh()</td>
<td>Updates the torque request by running the calculations again.</td>
</tr>
<tr>
<td>Class name</td>
<td>External Input</td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Relationships</td>
<td>Provides external input to the system</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>N/A</td>
</tr>
<tr>
<td>Attributes</td>
<td>N/A</td>
</tr>
<tr>
<td>Functions</td>
<td></td>
</tr>
<tr>
<td>-RequestAngle()</td>
<td>Requests a new steering angle of the system.</td>
</tr>
<tr>
<td>-Refresh()</td>
<td>Input Refreshes until receiving new angle for the system.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class name</th>
<th>Actuator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationships</td>
<td>This is the actual class responsible for any movement of the wheels in response to the request from the system.</td>
</tr>
<tr>
<td>UML Extensions</td>
<td>N/A</td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
</tr>
<tr>
<td>Functions</td>
<td></td>
</tr>
<tr>
<td>-ApplyTorque()</td>
<td>Applies the torque request to the wheels.</td>
</tr>
<tr>
<td>-Off()</td>
<td>Actuator is turns off until receiving the next application of torque.</td>
</tr>
</tbody>
</table>
4.3 Sequence Diagram

Sequence Diagram – A sequence diagram shows detailed interactions between the objects in a scenario. Each Object has an Object “lifeline” that is represented with a square with a name inside, and then a dashed line. The interactions between any objects are represented by arrows that are functions that are being called on the receiving side.

Figure 4.3.1 – This diagram illustrates the flow of data between objects in the system. Specifically it shows one circuit of the control loop.
4.4 State Diagram

State Diagram – A state diagram shows the states of the system, and the events that must occur to transition from one state to the other. The initial state is shown with a filled in circle, and then each event is an arrow with the condition that satisfies the transition above. The states are rounded edge rectangles divided into two sections. The first section is the name of the section, and the second is any entry, or exit conditions that must be satisfied before transitioning, or an action list. An action list is a list of events that are triggered when the system reaches a certain state.

![State Diagram](image)

*Figure 4.4.1 – This diagram illustrates the states of the system class. The descriptions over the arrows represent the events that occur to move from one transition to the next.*

Template based on IEEE Std 830-1998 for SRS. Modifications (content and ordering of information) have been made by Betty H.C. Cheng, Michigan State University (chengb at chengb.cse.msu.edu)
Figure 4.4.2 – This diagram illustrates the states of the Actuator Class.

Figure 4.4.3 – This diagram illustrates the possible states of the External Input Class.
5 Prototype

The prototype shows the interaction between the driver of the steer-by-wire vehicle and steering system. It illustrates possible scenarios of using the system and provide information about all the values that pass through the system and cause these scenarios.

5.1 How to Run Prototype

Using the prototype, the steering wheel can be controlled in order to obtain feedback about the status of the system. As the steering wheel turns, all of the given values are recalculated dynamically and output to the user. Through use of the prototype, all situations can be simulated and tested for any type of error.

A JavaScript Enabled browser must be used because the prototype uses jQuery. jQuery is supported by the following browsers:

Firefox 3.6, 5.0.x, 6.0.x, Internet Explorer 6+, Safari 5.0.x, Opera Current - 1 version, Chrome Current - 1 version

The basic keys to control the car are w, a, s, d (accelerate, left, brake, right). The red car in the top view is the one that is driven by user. This is used to show the vehicle angle, the direction, and car's motion visually. The steering wheel is used to show the steering wheel's angle and the vehicle wheel is used to show the vehicle wheel's angle.

The prototype is located at:

http://www.cse.msu.edu/~kimseu16/game/driver.html.
5.2 Sample Scenarios

Scenario 1: The vehicle accelerates and the steering wheel angle is 53°. The vehicle desired wheel angle will be 6.625° by the equation. However, the vehicle actual wheel angle has error and the steer-by-wire system will fix this error.

Scenario 2: If there is a system fault detected, the value of fault detection will be set by 1. Then, the vehicle will try to stop.
Scenario 3: The vehicle has negative velocity. It means the vehicle is driven in reverse. The velocity will be limited within -10 to 20 mph in the prototype.
6 References


7 Point of Contact

For further information regarding this document and project, please contact Prof.

Betty H.C. Cheng at Michigan State University (chengb at cse.msu.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.