Software Requirements Specification

for

Low-Cost Torsional Testing Machine

Version 1.2 approved

Prepared by Jonas Cooper, Joshua McCoy, John Taylor

CSE 3223 (Section 01) Software Engineering Senior Project II

May 11, 2025

Revision History

Name	Date	Reason For Changes	Version
Jonas Cooper, Joshua McCoy	10/6/24	Initial Document	1.0
Jonas Cooper, Joshua McCoy	11/12/24	Added Additional Functionalities	1.1
Jonas Cooper, Joshua McCoy, John Taylor	05/11/25	Added Additional Functionalities	1.2

1. Introduction

This software system is for the control and data collection of a torsional testing machine within MSU's department of Agricultural and Biological Engineering.

1.1 Purpose

The software system that this document describes is the control and data collection system that we are developing for MSU's department of Agricultural and Biological Engineering's student-made torsional testing machine. This is the first general release of the software. This document covers the entire scope of the software, including functionality, user types and interactions, and operational environment.

1.2 Document Conventions

This document uses Times New Roman font for all text. Section headers use bold Times New Roman font of size 18. Subsection headers use bold Times New Roman font of size 14. All other text uses normal Times New Roman font of size 12.

1.3 Intended Audience and Reading Suggestions

This document is intended for the development team (consisting of Jonas Cooper, Joshua McCoy, and John Taylor), and for the faculty of the Software Engineering Senior Project course. Users may find this document useful if wishing to further understand the software system. For users, sections 2 and 4 of this document may be of particular use.

1.4 Product Scope

The software system is designed to be able to automate controls for the torsional testing machine given a control input and to store data gathered from the machine's sensors into a digital document. The system will be installed on the user(s) computer and will save documents locally.

1.5 References

Arduino Documentation: https://docs.arduino.cc/learn/programming/sketches/
Drawio: https://www.draw.io/
Drawio Documentation: https://support.draw.io/display/DO/Draw.io+Online+User+Manual
Tkinter Documentation: https://docs.python.org/3/library/tkinter.html

2. Overall Description

2.1 Product Perspective

The system is a follow-on member of the torsional testing machine. The machine is capable of controlling and measuring the torque and angle of twist applied to materials placed in the machine. Through the Arduino Uno, the software will allow users to connect to the machine, control, and measure the results of testing applied to materials.

2.2 Product Functions

The system must:

- Allow connection to the Arduino Uno via serial communication
- Ability to start and stop testing
- Allow control of variables:
  - Angle of twist
  - Direction of twist
  - Speed of rotation
- View test data through a 3 column table (Angle, Torque, Time)
- View test data through a graph
  - Angle vs Torque
  - Time vs Torque
- Save test results to a permanent .csv file stored on the user's device.

2.3 User Classes and Characteristics

End User

- Security Level: Standard access to GUI
- Functions Used: Machine control, data exports
- Characteristics: Require a simple, intuitive interface to interact with the machine.

2.4 Operating Environment

The system will operate:

- Hardware Platform: Desktops or Laptops
- Operating System: Windows 10 (64-bit) or above

2.5 Design and Implementation Constraints

The system requires an open serial connection to the Arduino through the serial communication port. Developers are required to understand the English language. The customer’s organization will be responsible for maintaining the delivered software.

2.6 User Documentation

The following will be delivered along with the software:

- User manual

2.7 Assumptions and Dependencies

The GUI depends on the Arduino Sketch used to program the Arduino that dictates the machine's functionality. If there are errors or changes in the Sketch, then the UI may become dysfunctional.

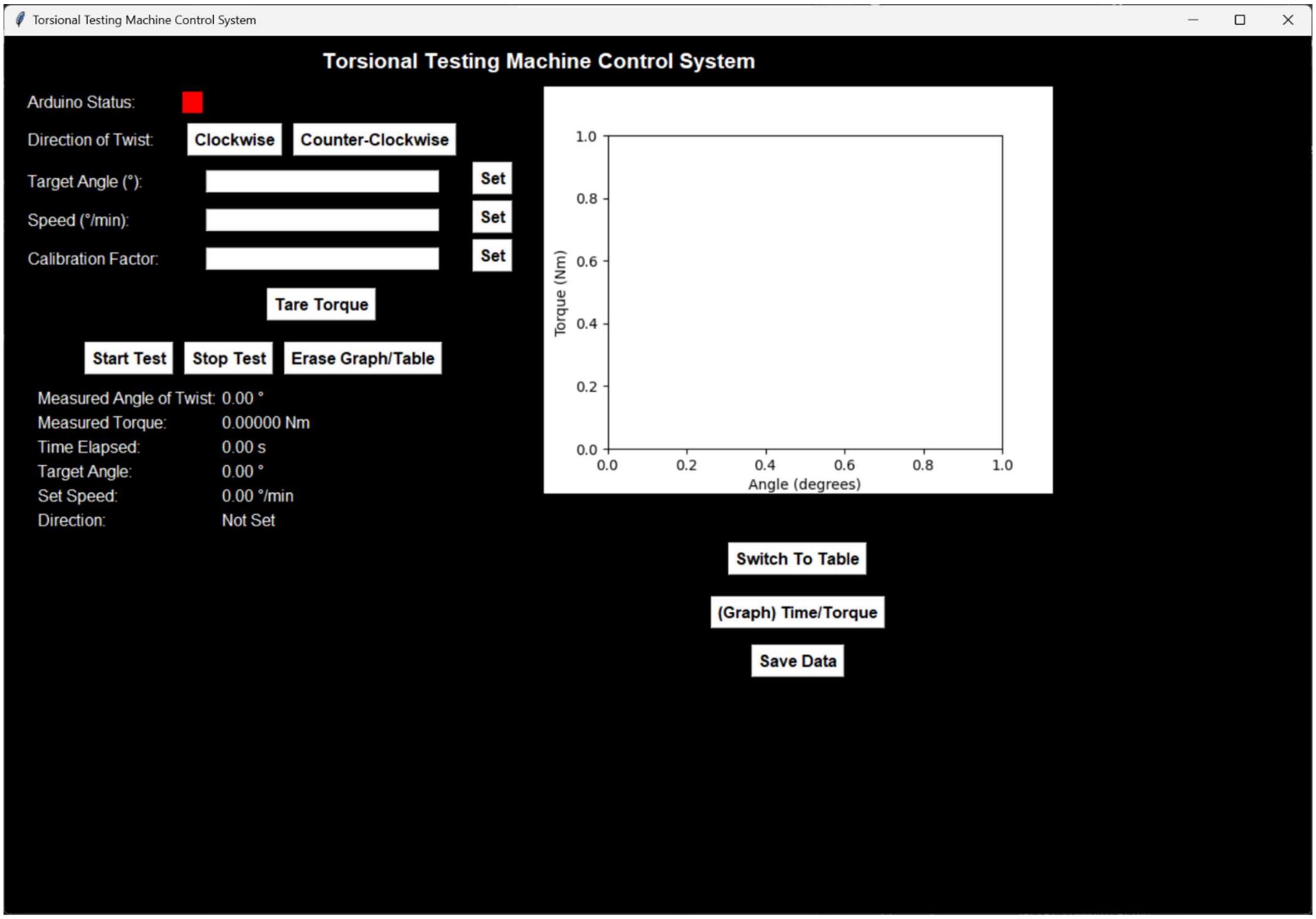
3. External Interface Requirements

3.1 User Interfaces

The control page will be the sole interface presented to the users of the software system. It will contain input buttons and text boxes interfacing with all functions of the system, including:

- Inputting control data to the machine (via prompt boxes)
- Sending control commands to the machine (via prompt button)
- Stopping test command to the machine (via prompt button)
- Receiving data from the machine (displayed to the user when received)
- Viewing test data (via table or graph)
- Saving test data to a .csv file (via prompt box)

Control Page:



Torsional Testing Machine Control System

Torsional Testing Machine Control System

Arduino Status: 🟥

Direction of Twist:

Target Angle (°):

Speed (°/min):

Calibration Factor:

Measured Angle of Twist:	0.00 °
Measured Torque:	0.00000 Nm
Time Elapsed:	0.00 s
Target Angle:	0.00 °
Set Speed:	0.00 °/min
Direction:	Not Set

- **Arduino Status Indicator:** This LED indicator displays serial connection status to the Arduino Uno. Green is successful. Red indicates an error.
- **Direction of Twist Toggle:** This button is for the user to specify whether the angle of twist should turn clockwise or counterclockwise.

- **Angle of Twist Prompt:** This prompt is for the user to input the angle of twist the machine should attempt to reach.
- **Speed of Rotation Prompt:** This prompt is for the user to input the speed (in degrees/min) that the motor should rotate.
- **Calibration Factor Prompt:** This prompt allows the user to manually calibrate the torque sensor.
- **Set Button:** These buttons confirm the set target angle, speed, and calibration factor that will be sent to the Arduino Uno.
- **Tare Torque Button:** This button automatically calibrates the torque sensor to 0.01 N*m.
- **Start Test Button:** This button sends a command to the machine with the testing variables provided by the user (if either parameter is missing, an error will display). It also starts the stopwatch and start the recording of data to the table and graphs.
- **Stop Test Button:** This button sends a command to the machine to stop cease all functionality. This button is expected to be pressed when a test has been completed.
- **Erase Graph/Table Button:** This button erases all data values in the table and graph. This action cannot be reverted once pressed.
- **Value of Measured Angle of Twist:** The machine returns its current angle of twist (provided by the angle encoder) to the program in near real time.
- **Value of Measured Torque:** The machine returns its current measure of torque (provided by the torque sensor) to the program in near real time.
- **Value of Time Elapsed:** After pressing the "Start Test" button, the timer begins to count in seconds and stops after the "Stop Test" button is pressed.
- **Target Angle Value:** After confirming the "Angle of Twist" value, it is displayed.
- **Set Speed Value:** After confirming the "Speed of Rotation" value, it is displayed.
- **Direction Value:** After confirming the "Direction of Twist" value, it is displayed.
- **Data Table:** This table will display the recorded angle of twist, torque values, and time every 200 milliseconds. The angle of twist, torque, and time will have their own respective columns.
- **Data Graph:** There are two graphs and they will display the recorded angle of twist, torque values, or time every 200 milliseconds. One graph will display the angle of twist (x-axis) and torque values (y-axis). The other graph will display the time (x-axis) and torque values (y-axis).
- **Switch to Table Button:** This button will swap the graph and table into the user's view.
- **(Graph) Time/Torque Button:** This button will allow the time vs. torque graph to be viewed.
- **Save Data Button:** This button allows the user to save the results of the current test in a .CSV file. Once pressed, the user can decide the name and location of the file.

3.2 Hardware Interfaces

Arduino Uno
The Arduino is a circuit board attached to the torsional machine that allows the control of the Cytron motor, rotary encoder, and the Qwiic scale that allows the accurate measure of torque applied by the torque cell. The software will communicate to the Arduino using serial communication via USB connection. The software will be used to control the Cytron motor, rotary encoder, and the torque cell. The software will also be able to display and record the data for each test and have the ability to export the data as a CSV file.

User Device
The user's device will connect to the Arduino via USB to USB-A connection and will run the software locally. The device will communicate with the Arduino using the UART protocol.

3.3 Software Interfaces

Microsoft Excel 2024 – Exported data from the product will be converted into a .CSV file that can be viewed in Excel.

3.4 Communications Interfaces

The only communications of concern are those between the computer and the torsional testing machine. It is not anticipated that this connection will be exposed, so we are not anticipating any security issues but there is still a chance of serial flooding if the system is utilized correctly.

4. System Features

4.1 Rotation Control

4.1.1 Description and Priority

Allow the control of the angle of rotation and the speed of rotation for the motor.

Priority: High
Benefit: 9
Penalty: 4
Cost: 1
Risk: 1

4.1.2 Stimulus/Response Sequences

- User input value for rotation angle, rotation speed, and rotation direction.
- System receives request and sends command to Arduino to facilitate the action.
- System displays current values for rotation angle and rotation speed.

4.1.3 Functional Requirements

REQ-1: Text box for integer input for Angle of Twist.
REQ-2: Text box for integer input for Speed of Rotation.
REQ-3: Button for directionality of Angle of Twist.
REQ-4: A "Send Command” button. If no input or an invalid input is in Angle of Twist or Speed of Rotation test box, display an error detailing to input a correct value.
REQ-5: A Test field for displaying the current Angle of Twist (deg).
REQ-6: A Test field for displaying the current Torque (Nm).
REQ-7: A Test field for displaying the set Speed of Rotation (deg/min).
REQ-8: A Test field for displaying the Direction of Rotation (cw/ccw).
REQ-9: A Test field for displaying Test Duration (seconds).

4.2 Data Export

4.2.1 Description and Priority

Allow the exportation of test data.

Priority: High
Benefit: 9
Penalty: 4
Cost: 1
Risk: 1

4.2.2 Stimulus/Response Sequences

- User selects "Save Data"
- User is given a file save dialogue tab to determine the name and location of the file
- System displays a confirmation message once the file has been created.

4.2.3 Functional Requirements

REQ-1: A "Save Data" button. If the data file does not get created due to error, display a message detailing the failure.

4.3 Data Visualization

4.2.1 Description and Priority

Allow the viewing of testing data during tests.

Priority: Medium
Benefit: 9
Penalty: 4
Cost: 1
Risk: 1

4.2.2 Stimulus/Response Sequences

- User submits data for angle of twist and speed of rotation.
- System displays data on table and graph.

4.2.3 Functional Requirements

REQ-1: Text box for integer input for Angle of Twist.
REQ-2: Text box for integer input for Speed of Rotation.
REQ-3: Text box for directionality of twist.
REQ-4: A toggle to view either the data table or graph.
REQ-5: A toggle to change the graph type (angle/torque or time/torque).

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must pass and receive information to and from the torsional testing machine in minimal time, ideally less than one second for communications between the software system and the machine. It must also perform calculations and save data to files in minimal time, ideally in under ten seconds for each operation.

5.2 Safety Requirements

The system must be designed so that when data is saved to a file, no damage is done to existing files or file systems on the computer that the software system is installed on.

5.3 Security Requirements

As there is only one user type in the system, there is no need for authentication to differentiate user types. The primary security concern is making sure that the software system does not access any files or file systems on its home machine that it doesn't require access to.

5.4 Software Quality Attributes

The significant quality characteristics of the software system are reliability, testability, and usability. The system is not intended to be a software system as a service, and is meant to only operate locally, so attributes such as availability and portability will be determined by the system machine Testability will primarily be of importance to the developers, while reliability and usability will be most important to the end user(s).

5.5 Business Rules

It is anticipated that the system is to be primarily accessed by Dr. Elder on his computer, but understood that access may be granted to other faculty or students authorized by Dr. Elder. It is the current understanding that in the case that individuals other than Dr. Elder are authorized to use the software system, they will be under the supervision and guidance of Dr. Elder.

6. Other Requirements

The software system will need to be capable of controlling the machine in the event of machine hardware being replaced and/or changed.

Appendix A: Glossary

Universal Asynchronous Receiver-Transmitter (UARP) - a communication protocol used for serial communication between devices.

Comma-Separated Value File (.CSV) - a plain text file used to store tabular data.