Automating Script Implementation of Motor Parameterization Testing Using Novel Coding Techniques

Name: Matthew Mustapic
Supervisor: Dr. Narayan Kar
Program: Electrical Engineering
Level of studies: MASc

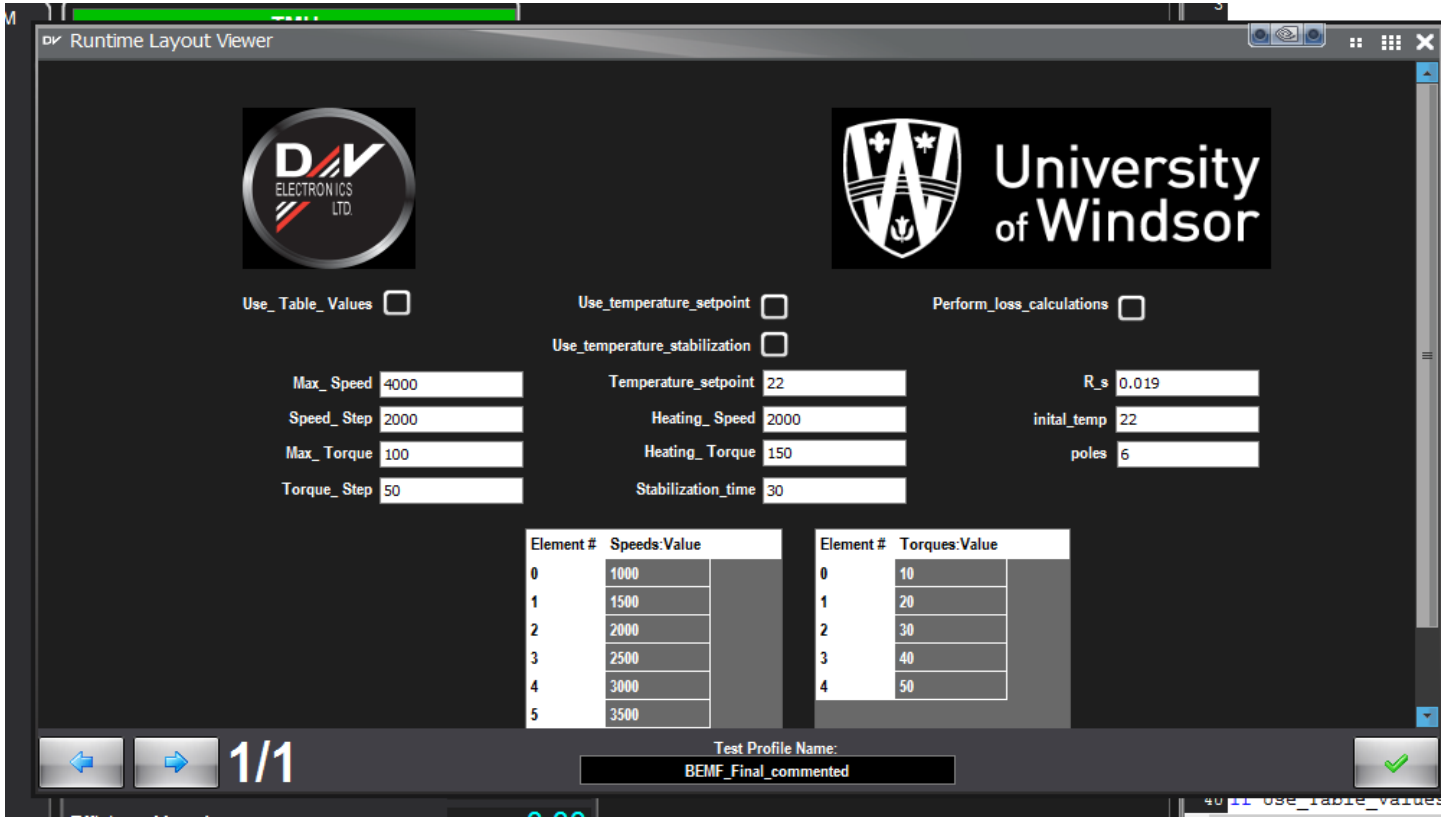
Project Description

- Enables testing of both high speed and high torque in one system
- D&V's testers primarily use components from Beckhoff Automation, programmed in Visual Studio with C#, downloaded into the PLC
- Implementation of new test functions in EPT-150
- Scripts which automate testing using the EPT system

EPT 150 Dynamometer & Power Supply

Project Description

Graphic User Interface for Test Condition and DUT Motor Definition



Objective(s)

- To implement current and novel testing methods into C++ scripts compiled onto TwinCAT PLC file
- Develop automated process to transfer **Simulink** models into accessible programs on the machine

Tasks/Plan

- Review current study cases in which code is already running and working on machine
- Get familiar with product's specific scripting language (similar to C++)
- Create scripting shell able to automate process of Simulink model to compiled code implementation
- Test functionality of converter tool, written with TCL

Expected Outcome & Deliverables

- Expand utility of current in-lab testing machinery
- Develop a code generator to enable Simulink models to be converted into C++ code for use in other projects
- Streamline implementation of other test from other projects

Progress Report

- Created example scripts using C++
- ex: loss_models.dvs
- Able to compile these scripts
- Developing methodology of code conversion from .slx to .dll file type
- Automated script using TCL will be used to make this a “1-click” process

Ex: Loss Model Code

Function definition for copper loss calculation with temperature effect

```
Function CopperLoss(Current_A As Double, Current_B As Double, Current_C As Double,
    Initial_Resistance As Double, Ambient_Temperature As Double, Winding_Temperature As Double) As Double
  dim R_measured as double
  dim Copper_Loss_A as double
  dim Copper_Loss_B as double
  dim Copper_Loss_C as double
  dim Copper_Loss as double

  R_measured = Initial_Resistance * (1 + (0.004041 * (Winding_Temperature - Ambient_Temperature)))

  Copper_Loss_A = Current_A*Current_A*R_measured
  Copper_Loss_B = Current_B*Current_B*R_measured
  Copper_Loss_C = Current_C*Current_C*R_measured

  Copper_Loss = Copper_Loss_A + Copper_Loss_B + Copper_Loss_C

  Return Copper_Loss
End Function
```

Loss Factors Determined

1. Copper loss with temperature
2. Rotational loss

Copper Loss