# Overview

## AVC systems are a critical element to automation of weld process. The weld arc creates current through air after breakdown voltage has been surpassed. The arc voltage is then a function of distance from the workpiece when used with a power supply in constant current mode. The AVC software, as defined in this document, will use Data Acquisition to measure arc voltage and a PID controller to maintain a constant or within tolerance arc length to ensure weld profile1 is maintained. This new SRS is an augmentation upon the original SRS and retains relevant information from the original SRS.

# Project Phases

## Phase I: Implementation of new Object-Oriented Programming (OOP) architecture and features

## Phase II: Test and Debug TigARC 2.0

## Phase III: Pilot Production with TigARC 2.0

## Phase IV: Full Remote Prototype – Dynasty Modbus

## Phase V: Volume Pilot Production

## Phase VI: Production Release

## Phase VII: End-of-life

# Project Goals

## Overhaul system architecture to Object-Oriented Programming (OOP)

## Daisy-chaining 2nd SmartMotor for positioner control

## Fix existing bugs and update existing features (sequence files, log files, GUI)

# Hardware

## NI Compact DAQ has been selected, but NI Compact RIO may be needed with “Real Time” processing to resolve latency issues

## NI Compact DAQ with supported modules:

### NI-9205 (AI)

### NI9861 (CAN Bus)

### NI-9263 (AO)

### NI-9482 (Relay)

### NI-9421 (DI)

### NI-9472 (DO)

## Servo Motors

### 2x Moog Animatics SmartMotor SM23165D-CDS7 option will be daisy-chained to control the linear actuator and positioner

### SmartMotor Interface (SMI) will be used to control motors, Thompson Engineering will produce/provide code for motor control in LabVIEW environment

### Motor moves to position driven by direction and distance command from PID controller is preferred, an absolute position mode with exclusion zones may also be considered in the future

## Linear Actuator Sensors

### 2x Panasonic GX-F8A to be used as limit switches and for homing command, needs implementation in firmware

## Emergency Off (EMO) Switch

### Omron A22E-M-03-EMO will be used

***Note: will physically disconnect arc contactor signal, and both SmartMotors’ power while simultaneously sending a trip signal to the DI module.***

# cDAQ Modules

## Analog Inputs

### Arc Voltage (Voltage Dyn): 0-10V, Voltage Feedback; +1VDC per 10V output

### Weld Current (Current Dyn): 0-10V, Current Feedback; +10VDC per 100% of peak current set on machine

***Note: set the Dynasty 400’s peak amp to 100A, therefore +10VDC per 100A. The 100A max default setting captures WPS process requirements in active production.***

## CAN Bus Communication (low priority)

### CAN Bus is required for full remote control of Dynasty 400 weld machine (Phase IV)

## Analog Outputs

### Current Out: 0-10V, defined in Sequence File, +10VDC per 100% of peak amp. Should correspond to Dynasty 400’s peak amp of 100A by default, similar to section 5.1

## Relays

### Arc Enable: arc contactor relay must be toggled closed-opened-closed with timing to initiate gas flow. The final close then strikes the arc to start the weld process. The relay must be closed during the weld process and opened at end of weld cycle.

***Note: closing this contact will strike an arc even if analog current is 0, machine strikes at min 5 amps. Therefore, it must be opened at the end of the weld cycle when the weld current <5 amps.***

## Digital Inputs

### EMO trigger signal

## Digital Outputs

### Reserve at least 3 digital outputs for indicator lights

# Motor Controls

## Motor operation should be implemented in two modes, Idle Mode and Weld Sequence Mode. In Idle Mode, the motor command buttons should be accessible and functional on the main GUI. During Weld Sequence Mode, the motor command buttons should be inoperable, and motors should be controlled by AVC/PID output and Positioner Speed (RPM) from Sequence File.

## Idle Mode motor commands for linear actuator:

### Home/Part Change – moves linear actuator to home position indicated by homing sensor. Home is also to be used as part change position for easy reload

### Set Torch Start Position – sets the position where welding starts. By default, the starting position would be where the linear actuator is located during startup

### Go to Torch Start Position – moves linear actuator to weld start position

### 4 Move buttons – move towards/away slowly/quickly. Use default motor speed defined in Config File for slow speed and 0.5”/s for fast speed

## Idle Mode motor commands for positioner:

### Set Positioner Start Position – sets the position where welding starts. By default, the starting position would be where the positioner is located during startup

### Go to Positioner Start Position – rotates positioner to weld start position

### 4 Job buttons – rotate CW/CCW slowly/quickly. Use default positioner speed defined in Config File for slow speed and 10RPM for fast speed

## Weld Sequence Mode motor commands:

### Implement AVC Lockout from Sequence File to lock linear actuator motor moves during arc start and downslope (arc taper to arc stop)

### Read linear actuator motor commands from PID controller output. PID output is recommended to be an integer where 1 = .001” of movement

### Read positioner values from Sequence File’s Positioner Speed (RPM)

# User Inputs

## Configuration File – a file to set and load software parameters at start up including but not limited to:

### Properties for Graphs

#### 0 to 100 amps full scale for current only graphs, and

#### -15 to 100 for voltage and current graph

### Software Sampling

#### Sampling Rate – Default: 2000 Hz

#### Samples per Acquisition – Default: 100 samples

### Motors

#### Linear Actuator Motor Speed – Default: 0.05”/s

#### Positioner Speed – Default: 1RPM

## Sequence File – a file for user to be able to load and adjust the weld sequence

### Header for defining single occurrence variables:

#### Motor Velocity

#### Voltage Set Point

#### Move SF

#### Integration Time

#### Pgain

#### Igain

#### Dgain

### Multiple steps with these parameters in each column:

#### Step Number

#### Step Time (s)

#### Current (A)

#### Ramping

#### AVC Lockout

#### Positioner Speed (RPM)

#### Voltage Set Point Offset (V)

## GUI Inputs

### Test description/job number

### Turntable serial number (should hold previously entered value)

### Operator name/ID (should hold previously entered value)

# System Outputs: Log Files

## Header for single occurrence variables:

### Name of Sequence File used

### Test description/job number

### Turntable serial number (should hold previously entered value)

### Operator name/ID (should hold previously entered value)

### Voltage Set Point

### PID & DAQ settings (Pgain, Igain, Dgain, Sampling Rate, Samples per Acquisition)

### Calibration Factor used (Turntable RPM/V)

### Process completed/sequence aborted flag

### New tungsten needed flag

## Weld data with these parameters in each column:

### Current sequence step number

### Voltage measured

### Voltage Set Point with Offsets calculated

### Voltage Error

### PID values

### ID error

### Total Gain (V)

### Total Move (mils)

### AVC Lockout

# Updates to the GUI

## Additional to the current ProdLite GUI, the following should be implemented:

### Replace right-clicking to load a sequence file with a button

### Add a running total weld count

### Display number of welds till tungsten change needed (value to be decided later)

### Clocked in the job yet or not, if not then do not allow weld to start

## Be mindful that the GUI should function as a touch screen without the use of a mouse or keyboard

# Safety and Software Requirements

## System Safety is of paramount concern. Effective safety interlocks and controls must be integrated with hardware. The following are requirements that outline integrated hardware and software for clarity. The software safety requirements must integrate with the hardware plan.

## Provide safety variables to establish criteria for Critical Software Errors defined as but not limited to the following:

### High voltage limit, >14 volts for >5 sec, configurable

### Low voltage limit, <7 volts for >5 sec, configurable

### Current setpoint error tolerance (e.g. |setpoint – measured| >10 amps)

### Other identified safety critical parameters as specified during development

***Note: must be tied to AVC Lockout variable where no CSE occurs during lockout.***

## Safety Interlock: provide alarms and error handling to open arc contactor relay when any Critical Software Error occurs

***Note: the relays are wired in series with arc contactor and EMO, so a user fault or software fault disables arc. This command circuit is the primary safety circuit.***

## Hardware EMO: User Emergency Off is a critical element to system safety and is implemented with a safety circuit to disable arc and motors in the event of an emergency or Critical Software Error.

## PID functionality (reference Table 1 and provided spreadsheet for control example)

### Control will be based on measured arc voltage error function defined in following section

***Note: the default setpoint voltage is 9.1V based on existing data sets and corresponds to an arc gap estimated at .060” for stainless steel, 13.8V for aluminum.***

### PID Variables: all variables to be calculated using compressed signals defined by sample rate and number of samples (per loop)

#### (Arc) Voltage Error. Defining the (Proportional) Error Function:

#### Voltage Error = (Measured Voltage) – (Set point Voltage) [Volts]

#### Derivative error – Derror. Defining rapid changes in voltage with a numerical derivative approximation by executing the following subtraction:

#### Derror = (Voltage Error) – (Shift register Voltage error) [Volts]

#### Integral error – Ierror. Defining bias voltage errors over time by executing the following average or equiv. average:

#### Ierror = , [Volts]

#### where n is number of loops evaluated, default is 20 loops at 10 Hz or 2 seconds and not to exceed 5 seconds of averaged voltage error

***PID Notes:***

***Control will be considered stable if Voltage Error is less that ±.5 Volts as read and recorded from DAQ compressed measurements.***

***Latency of less than 1 second is acceptable.***

### PID algorithm illustrated by the example in table 1 and formulas in reference spread sheets to be provided with this document. The control system is intended to be tuned by adjusting gains for each proportional, differential and integral gain set by the config file inputs and adjustable by the user in idle mode.

### The spread sheet includes:

#### Real system data with calculated control system response at each while loop execution

#### Ability to “tune” using PID scalars

#### math needed for calculating gains and PID integer output functions

### PID Error calculation:

#### PID Error [V] = (Voltage Error) x (PgainScalar) + (Ierror) x (IgainScalar) + (Derror) x (DgainScalar)

### Move calculation:

#### MoveScalar: based on experimentally determined .022V = 1 mil of travel (set in Config file)

#### Move Command Integer [mil] = (PID error [V] / MoveScalar [V/mil]) {rounded down to integer}

***Note: should be normalized so that the gain would be independent of the sampling rate.***

### PID Controller Design Plan

#### The PID controller is to output the Move Command as an integer intended to drive a motor command a direction and distance in thousands of an inch, or “mils”

***Note: the smallest motor step is .5 mil defined by motor resolution and lead screw.***

### Use of AVC lockout in sequence file with PID controller

#### The AVC lockout must be used to disable the controller during arc start and arc taper off

## Weld Current Profile

### Will be used to define and drive AO to specified weld current at any time in the weld process

### The software will provide a visual graph to specify the weld current profile before welding begins

### An additional graph will track the weld current progress in real time

## Weld Current Profile Execution

### Weld current data will be sent to Dynasty 400 weld machine via analog output (refer to 5.3) to control the weld current from the sequence file

### Arc Voltage Lockout shall be enabled during upslope and downslope, as defined in Sequence Files

## Acquisition and Control

### Software must:

#### Acquire Voltage and Current feedbacks from Dynasty 400 weld machine

#### Have the ability to view/acquire data of arc voltage to determine experimentally the relationship of arc gap to voltage

#### Have the ability to filter data for noise

#### Create an error function for feedback control

#### Create shift register on error function for PID numerical differential functions (refer to section 7.3)

#### Have logic to convert PID gain to position commands to ensure arc gap is within specified tolerance

***Note: arc gap is nominally .060”±.015” for stainless steel***

## Loop Speed

### Software must be able to process weld profile and send signals to weld equipment at recommended 10 Hz, and at a 4 Hz minimum (while loop speed)

# Features

## Upload feature to load weld sequence profiles, have a button instead of right-clicking

## Be able to manage the following:

### User inputs and property/parameter initialization

### Active weld program with sub states for each stage in weld profile, linked to graphical representation

### Data outputs to drive weld equipment

### Close tasks, actuate relays and reinitialize to default state

## A configuration file for key software parameters, default sequence files, and default variable values

## Be able to control 2x SM23165D-CDS7 option SmartMotors that are daisy-chained and still handle errors correctly

# Plans for future implementations

## Additional motors to be implemented as subsystems

## Control over 3 MFCs

## Absolute motor positioning mode with exclusion zones

## Have Touch-Retract functionality

## Seam tracking vision system

# References

## See Cal Weld procedures for weld profile definition. In general, the weld profile defines requirements for the cross-sectional shape of the weld to be achieved.

## Cable diagrams, System diagrams and Interconnect diagram are available upon request.

## Phase I: Leverage existing TigARC software.

## Software Platform: National Instruments LabVIEW and C++, C# code as needed to support Project in all Phases of the project requirements

## [Miller Dynasty 400 owner’s manual](https://www.millerwelds.com/files/owners-manuals/O275857A_MIL.pdf)

## Moog Animatics SmartMotor SM23165D Specifications

# Revision History

Rev 1, 4/29/2019, Initial release

Footnotes

1. Weld Profile is the collection of criteria that defines acceptable welds. See Ichor USW (Cal-Weld) Procedures for complete definition. The software term weld profile is the Weld Sequence Profile executed by the sequencer. Use of the term Weld Profile should be avoided where Weld Sequence Profile is preferred.

Table 1: Data Driven Control Example, refer to provided Excel spreadsheet.