CRANFIELD UNIVERSITY

AGOSTINHO GIL TEIXEIRA LOPES

ARC-BASED SENSING IN NARROW GROOVE PIPE WELDING

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE

PhD THESIS

CRANFIELD UNIVERSITY

SCHOOL OF INDUSTRIAL AND MANUFACTURING SCIENCE

PhD THESIS

Academic Year 2005-2006

Agostinho Gil Teixeira Lopes

Arc-Based Sensing In Narrow Groove Pipe Welding

Supervisor David Yapp

August 2006

Abstract

Big gains in productivity are found in tandem and dual tandem pipeline welding but require highly skilled operators who have to control the position of the torch very accurately for long periods. This leads to high demands on the skills and stamina of the operators of mechanised pipeline welding systems. There is a very strong motivation to fully automate the welding process in order to reduce the required skills and to improve consistency. This project focuses on the use of through-the-arc sensing for seam following and contact-tip-workpiece-distance (CTWD) control. A review of literature reveals very little development work on arc sensing for Pulsed Gas Metal Arc Welding (GMAW-P) in narrow grooves. GMAW-P is often used to achieve optimum properties in weld quality and fusion characteristics and also positional welding capability, all of which are important factors for pipeline welding.

The use of through-the-arc sensing for narrow groove pipe welding applications poses specific challenges due to the steep groove sidewalls and the use of short arc lengths, producing very different behaviour compared to V-groove arc sensing techniques. Tandem welding is also quite different from single wire techniques with both wires working in close proximity producing mutual interferences in arc signals.

An investigation was conducted in order to assess GMAW-P arc signals and it was found that improved consistency, higher sensitivity and less noise was present in voltages in the peak current period (peak voltages) used for torch position control. As a result of this investigation, a CTWD and cross-seam control system was developed and tested for single and tandem GMAW-P, using a 5º narrow groove. The test results have revealed accuracies for both controls of better than 0.2 mm. CTWD control was developed by following the existent welding procedure voltage average and cross-seam control by peak voltage comparison between maximum torch excursions.

Experiments were also performed to evaluate the influence of torch oscillation frequency on arc voltage behaviour and sensitivity, along with weld bead characteristics and fusion profiles. The resultant arc signal sensitivity was consistent with the results found in the literature for conventional GMAW. For GMAW-P, although no data was available from the literature for comparison, the results have shown no increase in sensitivity with the increase of oscillation frequency with the welding setup used.

Bead profile analysis performed at different sidewall proximities indicated that optimum wire to sidewall proximities can be found between 0 mm and +0.2 mm, measured from the outer edge of the wire to the sidewall corner. Accurate control is required since +1 mm proximity produced poor sidewall fusion and no signal differentiation for control recognition of groove width. This work showed that negative proximities or wire proximity beyond the sidewall produce wire burn back and hence very long arc lengths, resulting in poor depths of penetration and shallower beads, with major undercut defects.

In addition, this work has also shown the importance of torch oscillation width control, in order to produce accurate cross-seam control. A method is proposed to achieve torch oscillation width control by a continuous peak voltage comparison between centre and sidewall torch positions, using the optimum values of wire to sidewall proximity found and the resultant peak voltage value. This control will also provide a clear indication of actual groove width. Clearly this data can also be used to implement a system which adapts welding parameters to groove width.

Acknowledgments

I would like to express my deep appreciation to my supervisor David Yapp for all his support in the development of this work. I would also like to thank Stephen Blackman and Professor Stewart Williams for the opportunity to pursue this work in the Welding Engineering Research Centre and for providing the laboratory facilities required for the experimental part of this project.

My appreciation also goes to my fellow colleagues and friends Anthony Varughese, Harry Liratzis and Patricia Marmelo for their help, support and encouragement necessary to accomplish this research work. Thanks are also due to the technical staff, John Savill and Brian Brooks, for their creativity and support with the solutions and means to undertake the experimental work.

To my friends Fernando Ribeiro for guidance and encouragement and Ricardo Cruz for the long hours of fruitful discussions, I wish to express my deepest gratitude. To my new colleague Nuno Pepe for his prompt help, and for those not mentioned but have also contributed with their knowledge and wisdom for the accomplishment of this work, I also wish to thank them all.

I also wish to acknowledge the financial support from BP Exploration and the Pipeline Research Council International (PRCI) in funding this research project.

Finally, I would like to thank my family for all their support and I deeply wish to thank my wife Teresa for the love, patience, perseverance and encouragement that helped me to undertake and complete this work.

To my wife, Teresa

Contents

List of Figures

List of Tables

Nomenclature

A	cross-sectional area of electrode (mm^2)
a	cross sectional area of the wire (mm^2)
c_1	constants from process variables
c_2	constants from process variables
$CTWD$	contact tip to workpiece distance (mm)
D	torch rotational diameter
d_{ce}	torch-to-work distance for consumable electrode process (mm)
E_a	length dependency of arc voltage (V/mm)
f	pulse frequency (Hz)
H_{max}	CTWD maximum height
H_{min}	CTWD minimum height
I	welding current (A)
I_0	desired arc current (A) ; equation (2.14)
I_0	arc current when torch is at x_0 (A) ; equation (2.17)
I_b	background current (A)
I_c	arc current from centre torch position (A)
I_l	arc current from left-hand torch position (A)
I_{l1}	arc current from left-hand at consecutive torch oscillations (A)
I_{l2}	arc current from left-hand at consecutive torch oscillations (A)
I_m	GMAW-P mean current (A)
I_p	peak current (A)
I_p	profile arc current (A)
I_{peak}	detected arc peak current (A)
I_r	arc current from right-hand torch position (A)
I_{r1}	arc current from right-hand at consecutive torch oscillations (A)
I_{r2}	arc current from right-hand at consecutive torch oscillations (A)
$I_{sampled}$	sampled arc current (A)
I_t	template arc current (A)
$I_{threshold}$	calculated arc current threshold (A)

I_v	desired arc current when torch is located directly above joint centreline (A)
k_1	empirical constants for given materials and sizes
k_2	empirical constants for given materials and sizes
k_3	burn-off factors
k_4	burn-off factors
k_5	constant determined by the joint geometry
k_a	parameters of arc characteristics (V/mm)
k_p	parameters of arc characteristics (V/A)
L	inductance of welding circuit (mH) ; equation (2.1)
L	electrode length (mm) ; equation (2.3)
l	arc length (mm)
L_a	arc length (mm)
L_{a1}	long arc length
L_{a2}	short arc length
L_s	electrode length or stick-out (mm)
l_t	wire tip distance to bottom workpiece (mm)
MR	melting rate (mm/s)
R	resistance of welding circuit (mΩ)
R_a	current dependency of arc voltage (V/A)
R_L	resistance (Ω)
T	temperature (K)
t_b	background time (s)
t_o	time of a complete torch oscillation cycle (ms)
t_p	peak time (s)
t_r	time rate of recovery (ms/mm)
U_a	arc voltage (V)
U_c	parameters of arc characteristics (mV)
U_p	output voltage of power supply (V)
V_0	constant in arc voltage drop equation (V)
V_{arc}	arc voltage drop (V)
V_{avg}	arc average voltage (V)

V_{CAN}	arc voltage reference (V)
V_L	voltage drop along electrode extension (V)
V_{peak}	detected arc peak voltage (V)
V_r	voltage rate of recovery (V/mm)
$V_{sampled}$	sampled arc voltage (V)
V_{sum}	arc voltage accumulation variable (V)
W_f	wire feed rate (mm/s)
x	torch actual position
x_0	actual cross-seam centre of oscillation
y	length from the centre of both bearings
z	length from the wire tip to the centre of the fixed bearing
α	constants from wire characteristics ($mmA^{-1}s^{-1}$) ; equation (2.5)
α	averaging window or smoothing factor - the number of samples to be used on average calculation ; Figure 4.6
β	constants from wire characteristics ($A^{-2}s^{-1}$)
Δb	groove bottom width
Δl_t	variation of wire tip distance to bottom workpiece (mm)
Δp	wire proximity to the sidewall
ΔV	peak voltage recovery (V)
Δw	oscillation width error (mm) ; equation (2.14)
Δw	oscillation width of the wire tip ; Figure 3.9
Δw	oscillation width ; Figure 3.12 ; Figure 4.1
Δx	cross-seam off-centre error (mm) ; equation (2.12)
Δx	oscillation at the moving bearing ; Figure 3.9
Δx	torch cross seam position ; Figure 4.1
Δy	CTWD error (mm) ; equation (2.13)
Δz	torch height position ; Figure 4.1
Φ	resistivity of wire (Ωmm)
Φ	V-Groove angle

Abbreviations

ADC	Analog to Digital Converter
ANN	Artificial Neural Network
BASIC	Beginner's All purpose Symbolic Instruction Code
BHK	Babcock-Hitachi K. Type
CAN	Controller Area Network
CC	Constant Current
CTWD	Contact Tip to Workpiece Distance
CUA	Common User Access
CV	Constant Voltage
DLL	Dynamic Link Library
DSP	Digital Signal Processor
EIA	Energy Information Administration
FFT	Fast Fourier Transform
FLC	Fuzzy Logic Controller
GMAW	Gas Metal Arc Welding
GMAW-P	Pulsed Gas Metal Arc Welding
HSLA	High Strength Low Alloy
ID	Identification
IDE	Integrated Development Environment
JPVRC	Japan Pressure Vessel Research Council
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LH	Left Hand
LVDT	Linear Variable Differential Transformer
NGW	Narrow Groove Welding
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PID	Proportional, Integral and Differential
PSD	Power Spectral Densities

RH	Right Hand
RX	Receive
T@C	Torch at Centre
T@S	Torch at Sidewall
TX	Transmit

1 Introduction

World gas consumption is growing world wide at a record rate and is expected to double in size on the next 20 years (Figure 1.1). Investments of $20b/year are expected in new gas pipelines (Energy Information Administration – EIA [1]).

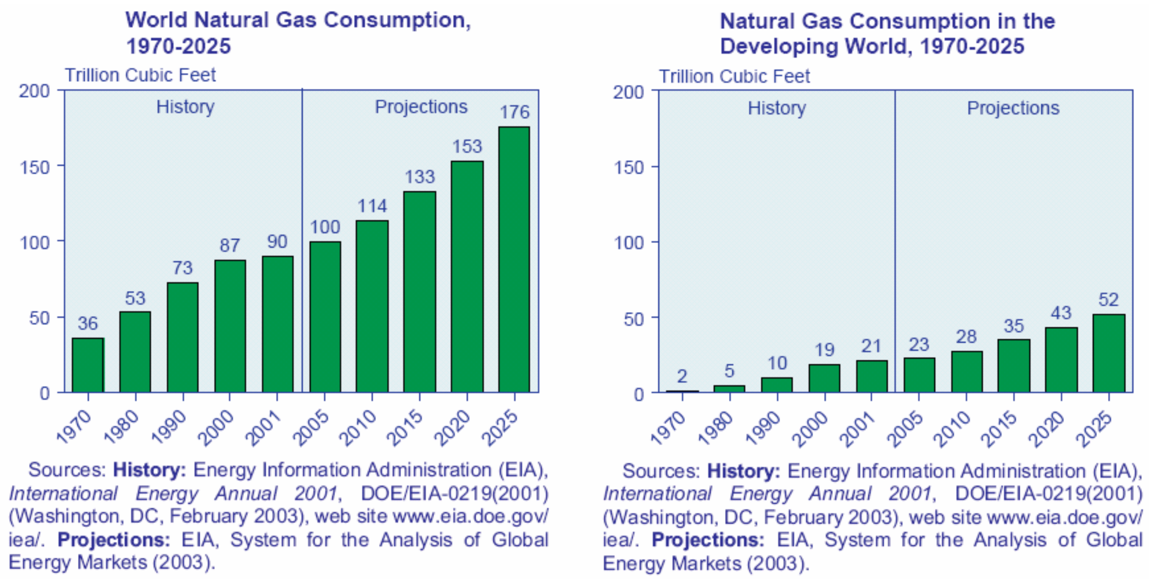


Figure 1.1 – World Natural Gas Consumption evolution [1]

Today's homes are being supplied by natural gas, pumped from places often thousands of miles away from them. Whenever possible, gas is transported through gas pipelines. With the increase in consumption, more and larger diameter pipelines are demanded. The new proposed Alaska gas pipeline for instance, covers 5,700 km starting in the Alaska, crossing Canada and ending in Chicago, USA. Each pipe has a diameter of 1.32 m (52 inches) and is 12 m long with 23 mm wall thickness of High Strength Low Alloy (HSLA) steel.

Girth pipeline welding is usually performed by the Gas Metal Arc Welding (GMAW) process in orbital multi-pass welds. Because of its improved characteristics and enhanced toughness properties with HSLA steels, Pulsed Gas Metal Arc Welding (GMAW-P) is increasingly used. This welding process can also produce improvements in weld quality and fusion characteristics, as well as positional capability, lower spatter generation and higher deposition rates when used in tandem torch systems.

Pipe welding mechanisation started 50 years ago for pipeline girth welding, using a welding head holding and moving the welding torch, attached to a band and moving around the pipe. Typical travel speeds were 0.3 m/min using the GMAW process. Current systems are still based on the same welding head/band principle although a second torch has been introduced with the ability for single and tandem wires in both torches. This increases by four times the metal deposition using the same travel speed rates. Figure 1.2 shows a welding head system with dual tandem torches, capable of travel speeds of up to 1.5 m/min and torch oscillation frequencies of up to 10 Hz, using GMAW-P. It is almost 5 times faster with 4 times higher metal deposition rates when compared with single wire GMAW systems.

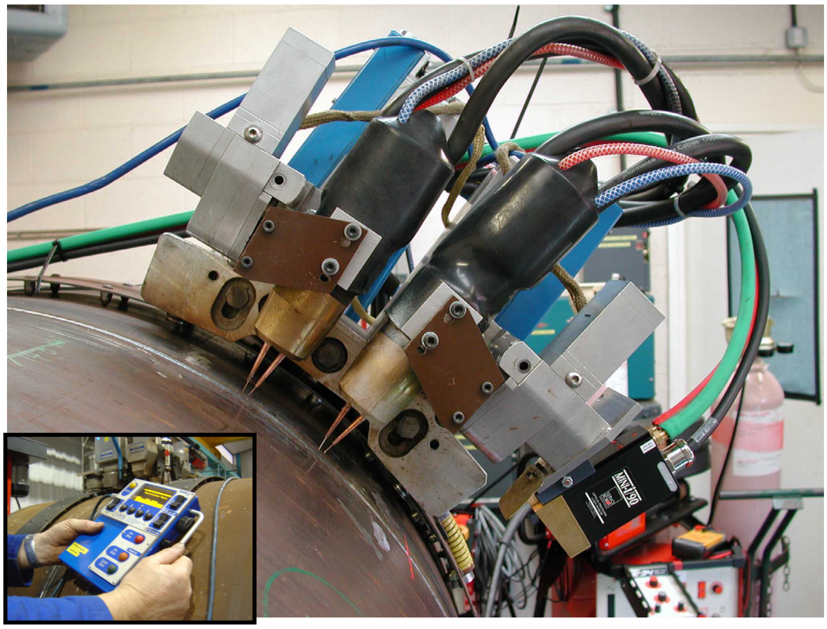


Figure 1.2 – Dual tandem welding head and pipe band with pendant control

Although remotely controlled by a pendant and possessing elaborate network connections, actual systems are still mechanised with no automatic control of process conditions. The work is still highly dependant on the monitoring, control and guidance of a skilled welder. British Standard BS EN287 [2] sets the requirements for manual welding skills and British Standard BS EN1418 [3] applies to users of mechanized welding systems. The user in the BS EN1418 standard is called an operator and not a welder and has a theoretical test to verify his knowledge of the welding station functioning. A skilled operator is then someone who has to be well qualified, who sets the appropriate welding parameters, who recognizes when the parameters need to be adjusted and controls the overall process. “This calls for an experienced manual welder with expert knowledge” [4]. Rapid growth of mechanised pipe welding onshore has been constrained by the difficulty and expense of finding sufficiently skilled pipeline welders [5, 6].

In order to increase productivity and to reduce the amount of deposited metal, bevel angles have been changed from 30° to 15° and lately to 5° for mechanised pipe welding, as shown in Figure 1.3. Pipe wall thickness may vary from 6 mm to 23 mm. Torch manual guidance for this type of groove is more difficult to perform to avoid contact-tip collision with the sidewall in the initial welding passes. However, a high contact-tip to sidewall distance creates low sidewall fusion leading to a decrease of welding quality and possible welding defects.

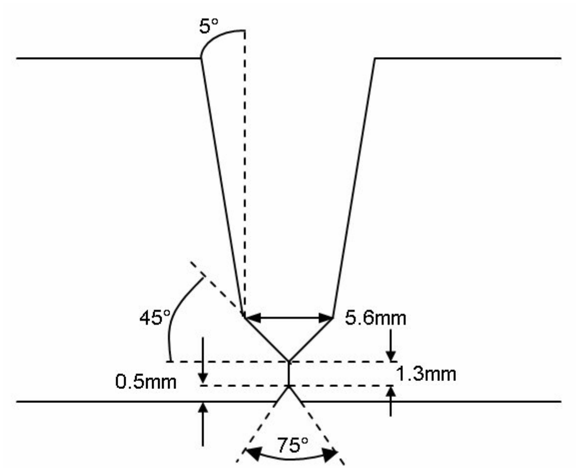


Figure 1.3 – Typical 5° bevel narrow groove

The application of newly developed high speed welding systems could be slowed by the limitation of the human being. The welding operator has to perform the tasks of control and guidance of the high speed travel multi-wire multi-torch welding head device and in general, to monitor the whole welding process condition. The adoption of HSLA steels, 5° narrow grooves and GMAW-P process has shown low tolerance for errors and a requirement for tight control of the whole welding process. The demand for semiautomatic or fully automated systems is high and only possible by developing sensor supported intelligent devices to achieve it.

1.1 Project aims and objectives

Lack of consistency, welding defects, inaccuracy, high productivity demands, lack of highly skilled and experienced welders are some of the problems in the welding industry today [6]. Fully automated adaptive systems for GMAW-P offer significant potential to overcome these problems. To achieve this, the whole process has to be investigated along with the application of sensors to enable control.

The main objective of this project was to study and develop a system where less control is needed from the welder and more is given to the machine. The system should be arc sensing based and robust but flexible enough to avoid recalibrations or reprogramming. As a result it is expected to be capable of operating at higher welding speeds, supplying better quality welds and reduced welding defects.

Specifically, the aims were defined as:

- to investigate the use of the arc as a sensor and to determine its sensitivity to variations in welding parameters for narrow groove pipe welding

- to develop algorithms for torch positioning to achieve control of contact-tip to workpiece distance, cross seam and torch oscillation width in the 5º narrow groove pulsed gas metal arc pipe welding
- to investigate the use of high speed torch oscillation and its effects on bead geometry and arc signals
- to develop off-line analysis software for experimental data acquisition with synchronised high speed filming of the welding arc
- to understand the arc behaviour in narrow groove pipe welding

2 Literature review

Welding automation is evolving and new methods and techniques are being developed every day to fulfil the demanding requirements of the emerging technologies. This chapter presents the state-of-the-art of the Gas Metal Arc Welding automation, process modelling, torch self-guidance technologies, narrow groove welding and high speed oscillation systems.

2.1 Gas Metal Arc Welding process

Gas Metal Arc Welding (GMAW) is a welding process where a wire is fed into an arc that is sustained between the wire tip and a base plate (Figure 2.1). The wire passes through a contact-tip that supplies the power to the arc. This is the last part where the wire touches any active component before melting in the arc. A potential differential is created between the contact-tip and the base metal. The initial arc ignition is produced by short-circuit with the wire touching the base metal. A constant flow of an inert (Argon based) or active (CO_2 based) gas allows an ionisation path to be created and shields the weld pool. The process is maintained with the continuous supply of power, wire and gas. A balance of several process parameters determines its working stability.

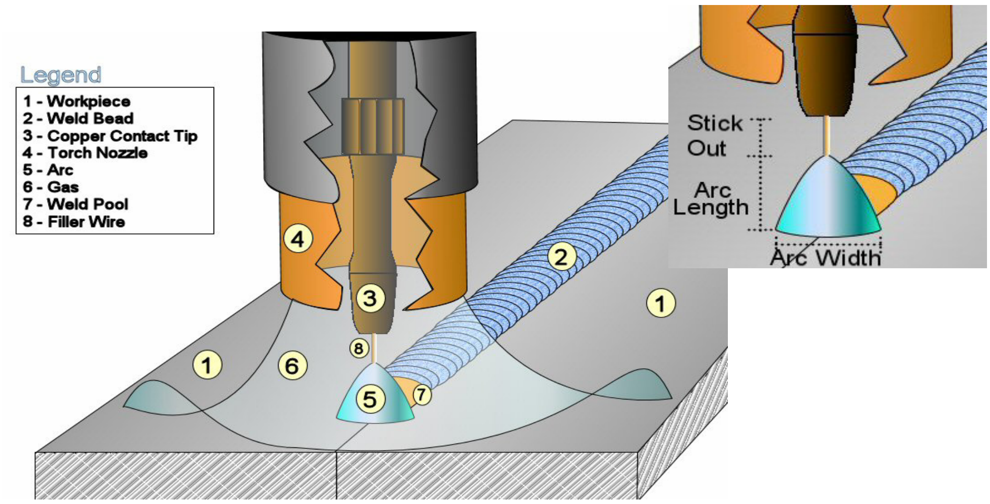


Figure 2.1 – The GMAW and GMAW-P processes

2.1.1 Voltage models

Circuit voltage drop models of the GMAW process were found from several different authors. Pan [7] and Ushio et al [8, 9] have shown that the output voltage of the power supply is equal to the sum of three voltages: the inductive, the resistive and the arc voltages as shown in equation (2.1).

$$U_p = L\frac{di}{dt} + Ri + U_a \tag{2.1}$$

where

U_p	output voltage of power supply (V)
L	inductance of welding circuit (mH)
R	resistance of welding circuit (mΩ)
i	instantaneous welding current (A)
U_a	arc voltage (V)

The authors also represented arc voltage as the sum of three terms as shown in equation (2.2).

$$U_a = k_a L_a + k_p I + U_c \tag{2.2}$$

where

U_a	arc voltage (V)
L_a	arc length (mm)
I	effective value of arc current (A)
K_a, k_p, U_c	parameters of arc characteristics (V/mm, V/A, mV)

Bingul et al [10-13] represented the voltage at the terminals of the power supply by the sum of electrode voltage and arc voltage. The electrode voltage model is based on a temperature dependent resistance of the wire (electrode) extension beyond the contact tip, as shown in equation (2.3).

$$V_L = IR_L, \quad R_L = \frac{\Phi(T)}{A} L \tag{2.3}$$

where

V_L	voltage drop along electrode extension (V)
I	current (A)
R_L	resistance (Ω)
Φ	resistivity of wire (Ωmm)
T	temperature (K)
L	electrode length (mm)
A	cross-sectional area of electrode (mm^2)

The Bingul et al [10-13] arc voltage model is shown in equation (2.4) and it is similar to the previous arc voltage model shown in equation (2.2).

$$V_{arc} = V_0 + R_a I + E_a l \tag{2.4}$$

where

V_{arc}	arc voltage drop (V)
V_0	constant in arc voltage drop equation (V)
R_a	current dependency of arc voltage (V/A)
I	current (A)
E_a	length dependency of arc voltage (V/mm)
l	arc length (mm)

With the increase of the welding current, the relationship between voltage and current becomes approximately linear [14]. From the previous equations (2.2) and (2.4), it is clear that with a constant welding current, arc voltage variations are directly related to arc length variations and thus to the welding voltage variations at power supply terminals.

2.1.2 Melting rate

Another important parameter in GMAW welding process is melting rate. It is considered the most important factor for the welding process productivity assessment [15]. It is defined as the mass of filler wire melted in a unit of time and can be represented as shown in equation (2.5) [15-17]:

$$MR = \alpha I + \frac{\beta l I^2}{a} \tag{2.5}$$

where

MR	melting rate (mm/s)
I	current (A)
l	stick-out (mm)
a	cross sectional area of the wire (mm^2)
α , β	constants from wire characteristics (mmA^{-1}s^{-1}, A^{-2}s^{-1})

GMAW is considered to be in steady-state when melting rate is equal to wire feed rate (also known as wire feed speed). The differential between rates is reproduced in the arc length as shown in equation (2.6) [7, 16].

$$\frac{dL_a}{dt} = W_f - MR \tag{2.6}$$

where

L_a	arc length (mm)
W_f	wire feed rate (mm/s)
MR	melting rate (mm/s)

2.1.3 Arc length

Considering a constant wire feed speed, changes in arc length produce changes in melting rate. The system self-balances its variables to achieve a new steady-state. The dynamic state of the process is usually a result of changes in contact-tip to workpiece distance or CTWD. Although in mechanised systems it is a predetermined value, this value suffers small changes caused by variations in preparation, thermal distortion and weld pool behaviour. CTWD is the sum of the observed arc length and wire stick-out or wire extension as shown in equation (2.7).

$$CTWD = L_a + L_s \tag{2.7}$$

where

$CTWD$ contact tip to workpiece distance (mm)

L_a arc length (mm)

L_s stick-out (mm)

It should be noted that stick-out voltage is small compared to arc voltage [16] and so, voltage variations are more evident as a result of arc length variations than of stick-out variations. Figure 2.2 shows a relationship between current, voltage and arc length for Constant Voltage (CV) GMAW power supply [17].

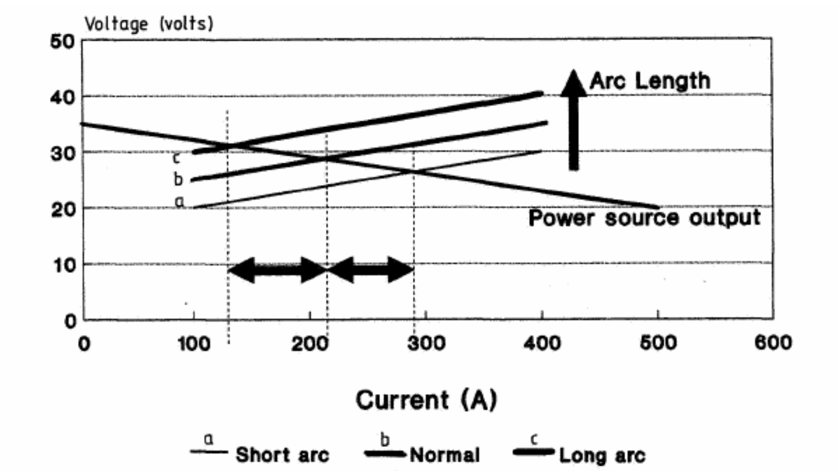


Figure 2.2 - Relationship between current, voltage and arc length, with CV GMAW [17]

There is no fixed ideal value for arc length and each procedure may have its own value. In general, arc length should provide a balance between arc stability, bead shape and weld metal penetration or depth of fusion, low spatter and good weld mechanical properties. A very short arc may result in excessive spatter and arc instabilities while a short or normal arc results in a better weld pool control and bead shape, higher travel speeds (lower heat input), less fume generation, and less susceptibility for porosity and cracking [18]. For pipeline welding, arc length is usually short in order to create smaller weld pools thus reducing the effects of the gravitational forces when welding in over-head positions. A short arc is used for narrow-groove pipeline welding since a long arc would create problems with arc deflections to the sidewalls and hence undercut defects and low root penetration.

2.1.4 The self-regulation mechanism

The self-regulation mechanism is an important characteristic of the GMAW process. It is an automatic mechanism that recovers system balance should any instability in the welding process occur caused essentially by variations of CTWD, inside certain limits. When process stability is achieved ($W_f = MR$), the system is known to be in its steady-state or "dynamic equilibrium" with arc length and stick-out values steady.

In a CV GMAW process, a reduction of CTWD initially produces a reduction in arc length and hence a reduction in voltage. The CV power supply responds to this variation and then by increasing the welding current to re-establish the voltage levels, increasing therefore melting rate and burning back the wire and hence reducing stick-out. Reducing stick-out increases the arc length and a new equilibrium point is attained. This natural self-regulation mechanism has created a demand for the use of CV GMAW power supplies mainly for manual welding situations [17]. For CC GMAW power supplies, the process is more complex.

In practice, self-regulation does not occur instantaneously and it is dependent on the characteristics of the power supply. Ushio et al [8] has developed a system where a torch is oscillated up and down changing cyclically *CTWD* values at frequencies that vary from 0 Hz to 10 Hz. Frequency analysis has revealed that from 0 Hz to 0.6 Hz equilibrium was completely achieved, decaying rapidly up to 6 Hz and never achieving equilibrium for frequencies above 6 Hz. Figure 2.3 shows the results [8] of the frequency response (x-axis) against the variation ratio in amplitude between wire extension and welding current (y-axis).

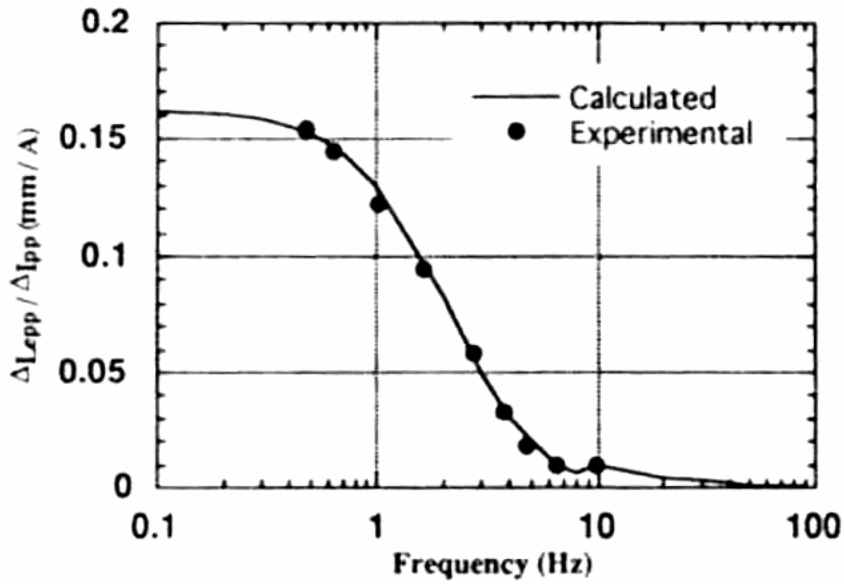


Figure 2.3 – Comparison in frequency-domain between the calculated and experimental results of the variation ratio in amplitude between wire extension and welding current [8]

According to these results, approximately 150 ms were necessary for the complete equilibrium, with the welding setup used and induced CTWD variations. Pan [19] also developed an empirical model based on experimentation where a rotating torch was inclined 45º to the baseplate in order to produce controlled variations of CTWD with frequencies from 0 Hz to 50Hz (Figure 2.4).

Where

H_{max}	CTWD maximum height
H_{min}	CTWD minimum height
D	torch rotational diameter

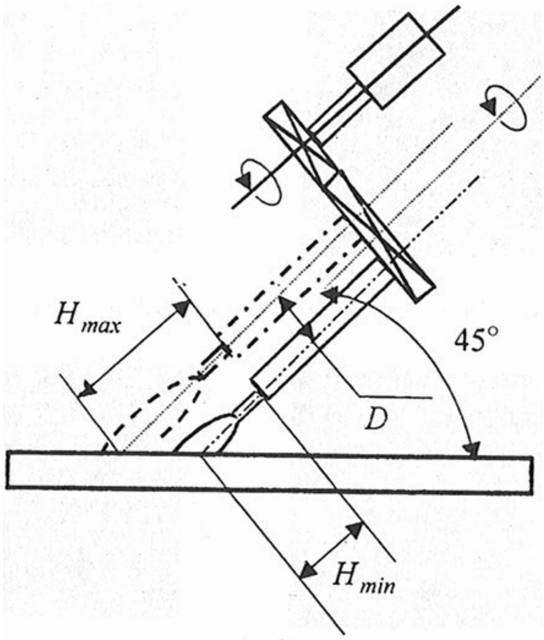


Figure 2.4 – Rotating torch tilted 45º to perform CTWD variations from Pan [19]

The results were very similar to those from Ushio et al [8] where changes in the variation ratio are only apparent in the frequency range between 2 Hz and 10Hz. From the experiments, Pan also concluded that the level of variation is different when using different power supply output characteristics (the voltage/current drooping characteristics), although maintaining the same pattern at the same frequencies. These changes were sensed not in the frequency domain but in the amount of variation between wire extension and welding current. It is more accentuated for power supplies with a lower voltage/current drooping ratio.

2.1.5 Torch oscillation

Mechanical torch oscillation or weaving is a widely used method for spreading the arc improving heat distribution and sidewall fusion in GMAW welding [20] in mechanical welding. The effect of torch oscillation on V-grooves is similar to the 45º tilted torch from Pan's [19] experiments, producing fixed changes of CTWD on each torch oscillation cycle. Many authors have reported arc signal sensitivity improvements when using high speed oscillation/rotation torches for

GMAW welding [19, 21-53]. The self-regulation mechanism produces arc signal variations in accordance to CTWD changes.

For CV GMAW, CTWD variations are more detectable in the welding current and for CC GMAW are more detectable in the welding voltage. The amount of signal variation is related to the amount of CTWD variation. Depending on the torch oscillation frequency used, the same CTWD might result in different values of arc signal variation. Figure 2.5 illustrates the process.

For low oscillation frequencies equilibrium is achieved with arc length and stick-out finding a new balance point. For high oscillation frequencies, the equilibrium is never achieved and thus, stick-out does not have time to change and CTWD difference is equal to the change in arc length. In this way, arc signal sensitivity is higher for higher oscillation frequencies: $(L_{a3} - L_{a4}) > (L_{a1} - L_{a2})$.

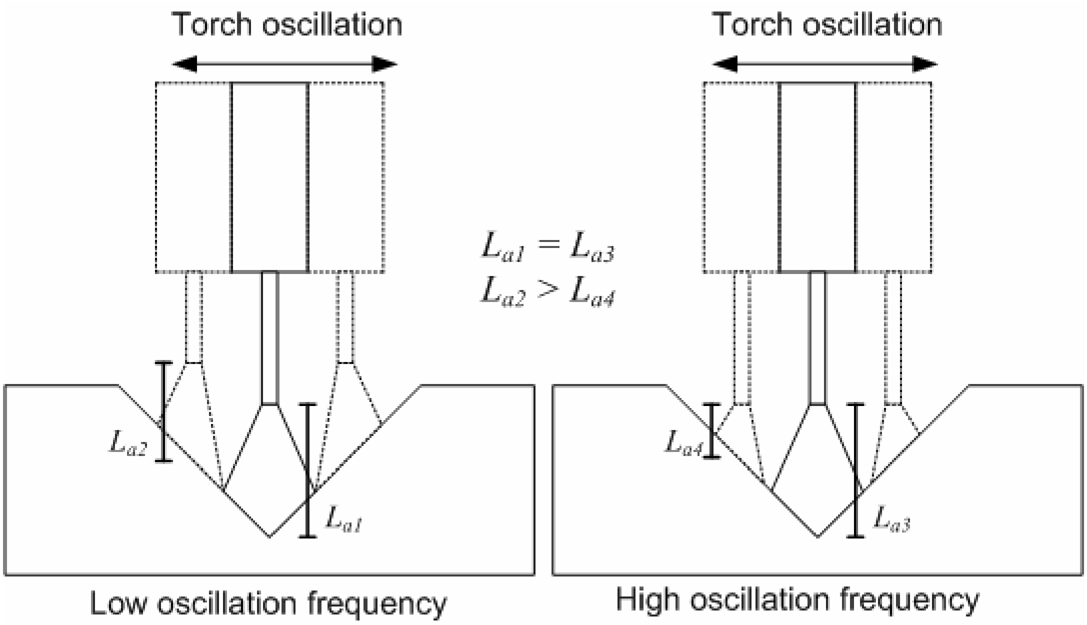


Figure 2.5 – High and low torch oscillation frequency differences in arc length

2.1.6 Pulsed Gas Metal Arc Welding

Pulsed Gas Metal Arc Welding (GMAW-P) is a special metal transfer mode of GMAW based on maintaining a welding current value in the spray metal transfer region long enough to melt, detach and transfer a molten droplet of metal, switching then to a relatively low current value to sustain a small arc until the process repeats, as shown in Figure 2.6. In this way, controlled drop spray transfer can be achieved with the average welding current lower than the transition current for spray transfer. Positional welding can thus be performed at low currents, but with controlled drop spray transfer instead [54].

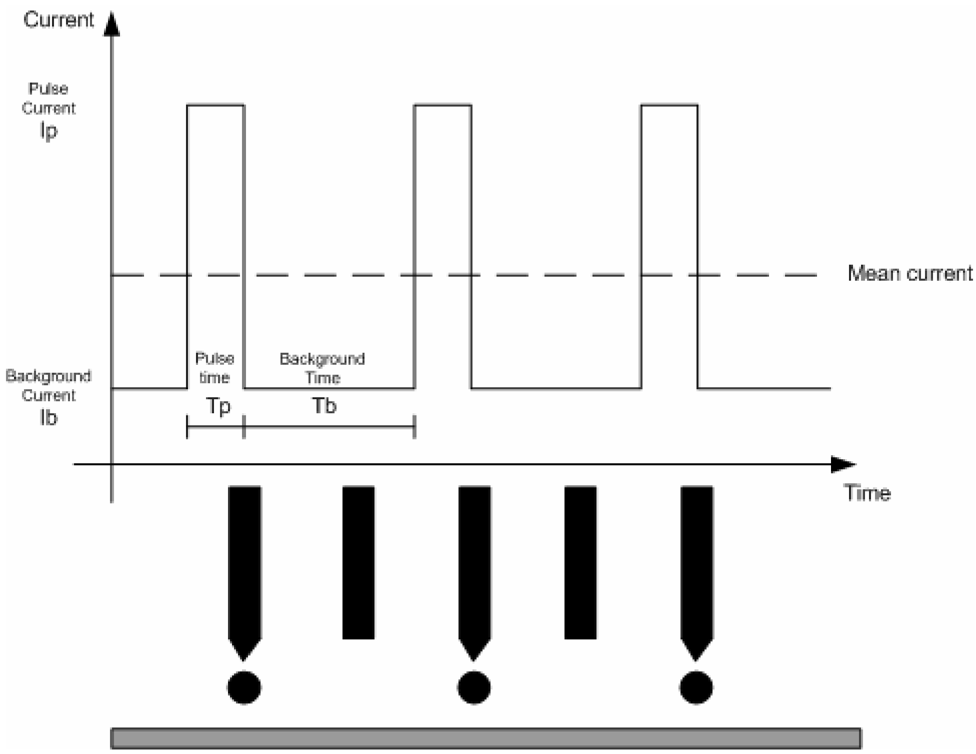


Figure 2.6 - Droplet transfer in pulsed GMAW [17]

The introduction of transistorised power supplies in the 1970's enabled the development of GMAW-P. However, the process was very demanding due to the complexity of adjusting many process variables (peak time, peak current, background time, background current, wire feed speed) for a stable process. Due to its complexity and cost this technique was not widely used [54, 55].

In late 1970's GMAW-P was simplified by the development of Synergic GMAW-P at The Welding Institute – Cambridge – UK, by Amin et al [56]. The complex task of setting process variables was substituted by changing a single parameter: Wire Feed Speed. All other parameters are then changed accordingly. The set of developed relationships are recorded inside the power supply and only one control knob is available to the welder.

The mean value of arc current in GMAW-P is expressed as [16, 17]:

$$I_m = \frac{I_p t_p + I_b t_b}{t_p + t_b} \tag{2.8}$$

where

I_m	GMAW-P mean current (A)
I_p	peak current (A)
t_p	peak time (s)
I_b	background current (A)
t_b	background time (s)

Melting rate for GMAW-P can then be calculated as [16]:

$$MR = k_1 I_m + k_2 I_m^2 l_s \tag{2.9}$$

where

MR	melting rate (mm/s)
I_m	GMAW-P mean current (A)
l_s	stick-out (mm)
k_1, k_2	empirical constants for given materials and sizes

A more detailed model is shown in equation (2.10) considering the proportions of high and low currents of the pulse and their respective times [57].

$$MR = k_3 I_m + k_4 l_s (I_p^2 t_p + I_b^2 t_b) f \tag{2.10}$$

where

MR	melting rate (mm/s)
I_m	GMAW-P mean current (A)
l_s	stick-out (mm)
I_p	peak current (A)
t_p	peak time (s)
I_b	background current (A)
t_b	background time (s)
f	pulse frequency (Hz)
k_3, k_4	burn-off factors

The Joule heating in the second term of equation (2.10) does not use the average current as in equation (2.9) but the influence of each current level of the pulse in the melting rate process.

2.1.6.1 GMAW-P dynamic process

GMAW using CV and CC power supplies is a steady-state process in spray transfer mode, the dynamic state changes when arc length disturbances occur produced mainly by CTWD variations. GMAW-P on the other hand is in a constant dynamically changing state produced by the cyclical changes in the welding current. The high and low values of the welding current create constant rebalance of arc length and stick-out around two different equilibrium points. In this way, GMAW-P is considered a non-equilibrium process [12, 58]. Dynamic models for GMAW-P are the same for GMAW.

2.1.6.2 Adaptive versus non adaptive pulse control mode

GMAW-P has two control modes of operation: adaptive and non adaptive. They refer to ability of the power supply to respond to external changes and disturbances that interfere with the process quality and stability. Each power supply manufacturer has their own proprietary way of implementing this control but in general the control is attained by changes in pulse parameters [17]. For instance, Fronius International GmbH changes pulse frequency and amplitude with changes in CTWD [59]. It should be noted that the power supply adaptive mode may conflict with attempts to make voltage measurements for control purposes [60].

2.2 Through-the-arc sensing

Through-the-arc sensing is a well established technique ideal for the welding environment. It is based on measurements of the welding arc parameters such as current and voltage. The widespread application of this technique in welding is based on the increased research in areas of arc welding such as process quality and automatic torch guidance [8, 26, 61]. The application for pipeline welding has also been reported [62-64]. Table 2.1 summarises the advantages and disadvantages of arc sensors.

Table 2.1 – Advantages and disadvantages of arc sensors

	Advantages	**Disadvantages**
Arc based	No extra sensors No need for registration Not susceptible to damage	Complex data analysis
Add-on sensors	Simple data analysis	Cost of extra sensors Need registration Wire deflections reduce accuracy Susceptible to damage

The use of arc based sensors for torch control of cross-seam torch position (seam tracking or seam following in some literature) or of torch height position (CTWD control), is based on the principle of measuring arc signal changes that occur as a result of changes in arc length and hence in CTWD values. The following sections review the application of through-the-arc sensing technologies for torch guidance applications.

2.2.1 GMAW open groove

Open groove is a wide angle groove formed between two walls, typically 90º for fillet welds or two 30º-45º bevel angles for V-groove welds. First reports of GMAW welding torch guidance by arc sensing techniques were developed for open grooves [65].

2.2.1.1 Cross-seam control principle

As shown previously, changes in CTWD produce variations in arc signals. This variation can be sensed and used for torch cross-seam control. King et al [65] (later reviewed by Pan [19]), suggested two ways of implementing a GMAW cross-seam control: twin-electrode and oscillatory wire (Figure 2.7). In GMAW with twin-electrode scheme, both electrodes are side by side along the path. Both arcs have the same length when the torch is in the groove centreline and different when the torch deviates from the centreline.

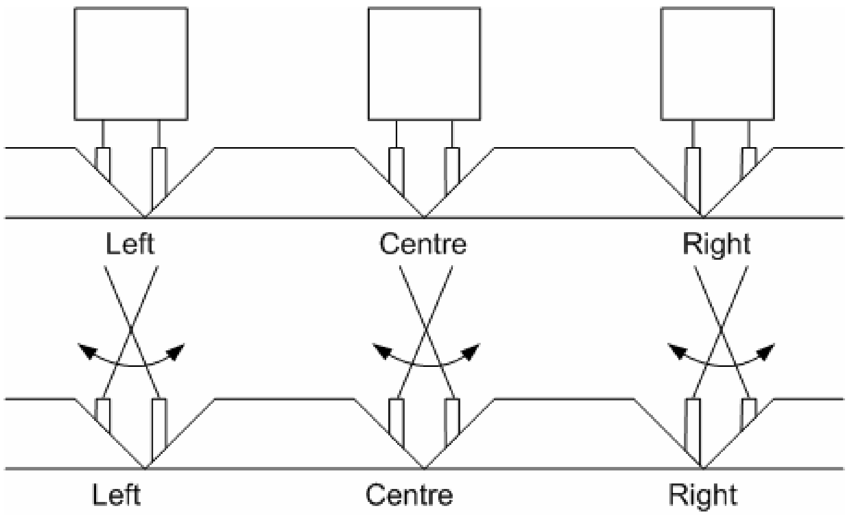


Figure 2.7 – Cross-seam control schemes implemented by King et al [65] – twin-electrode (top) and oscillatory single wire (bottom)

In the oscillatory single wire scheme, arc length is equal at the maximum extremes of oscillation when the torch is aligned with the groove centreline and un-equal when it is off-centre. In both twin-wire and oscillatory wire systems, it is possible to detect this disparity by the use of arc signals. For CV power supplies the change is more pronounced in the welding current and for CC power supplies in the welding voltage. Since CV it is the most common process used in GMAW because of the natural self-regulation mechanism, most of the reviewed literature refers to the use of welding current to detect CTWD variations.

In general, for single wire or tandem welding (where one wire is in front of the other along the path) methods, arc signal changes are used for torch cross-seam control if the torch produces a cyclical motion across the seam to allow the signal comparison at torch maximum excursions. This motion can be oscillatory, pendular or rotational. The measurements of arc signals at different positions of each cycle of the torch movement make possible the detection of discrepancies in the torch path.

The cyclical torch oscillation in an open groove using CV GMAW produces a variation in welding current as shown in the graph of Figure 2.8. The welding current is higher at the extreme torch excursion (points A and C) caused by the shorter arc length and hence an increase in melting rate whilst a lower welding current if found at the centreline (point B) from the opposite reason. Jieyu et al [66] stated that the higher current values at extreme torch positions are better for cross-seam control since they are more robust and less influenced by noise when compared to the lower current values of the centreline used by some researchers to achieve control.

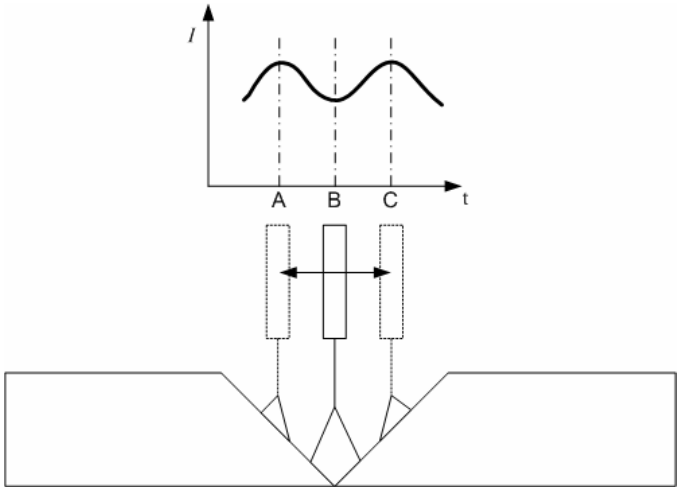


Figure 2.8 – Arc current variation with torch position in the groove

2.2.1.2 Electronic control

The control mechanism developed by King et [65] was based in discrete electronic components (transistors and/or operational amplifiers) working as welding current comparators and driving the torch cross-seam motor when a difference was found to reduce the difference to zero. For the oscillatory single wire system, it is important to detect when the weaving motion of the torch reaches both extremes. This can be implemented by positional sensors such as conventional switches to detect when the torch reaches each maximum excursion. When a switch is triggered, the welding current is measured and compared to the stored value measured at the opposite side. Their difference indicates the amount of torch off-centre that needs to be corrected. The same principle can be achieved by rotational torch movement. The rotational movement is measured by an encoder detecting the contact tip position when reaching the maximum excursions of the seam [52].

Patents by Suzuki [67] and Sugitani [68] describe cross-seam control using oscillatory torch systems. An electronic circuit was used to calculate the mean current values from both torch oscillation extremes, and a short-circuit detection scheme was operated to disable the measurements in the occurrence of short-circuits. When short-circuiting occurs, arc current rises higher than the mean value altering the mean value calculation, thus supplying incorrect information to the cross-seam comparator. This system is limited to short arc applications with frequent short-circuiting or applications based on dip transfer.

For dip transfer applications, Philpott [20] developed a solution for cross-seam control and CTWD estimation for dip metal transfer applications. Basically, an electronic circuit with operational amplifiers work as a divider for resistance calculation. Dividing voltage by current it is possible to establish the relationship between CTWD and resistance, since arc length is zero in the dip period and CTWD is approximately equal to stick-out. CTWD measurements had a real-time accuracy of +/- 2 mm being “certainly better than is normally achieved by a manual welder”, according to the author. This author considered that for cross-seam control the accepted tolerance was within +/- 5mm. It should be noted that the actual groove width for root pass of a 5º narrow groove pipeline welding is 5.6 mm, and a much lower tolerance on torch position is required (Figure 1.3 – p. 3).

2.2.1.3 Computational control

Different computation methods were used by researchers to achieve control over torch position, based on arc signals acquisition and processing, such as open and closed loop control systems with PID controllers, fuzzy logic controllers (FLC) and artificial neural networks (ANN). Some researchers also used mixed controllers. PID is a linear controller that generates an output value from a single input value based on three predefined parameters: Proportional, Integral and Derivative [69]. This controller cannot be applied to non-linear situations. Fuzzy controllers are based on a set of predefined rules applied to the input(s) to generate the intended output(s) [70]. New situations are not managed by these controllers unless new rules are defined. ANN controllers are knowledge base networks programmed to work as an interconnected neurons to interpret the inputs and to generate the outputs [70]. This controller needs the necessary information from training to attain the desired results. For new situations the network has to be retrained.

Jieyu et al [66] proposed a solution based on a closed loop control system where arc signals are acquired, filtered and averaged by computer algorithms. Welding current was the main signal used for control by comparing the values between torch maximum excursions. The difference is output to a PID controller that calculates the right amount of motion needed to centre the torch until a zero difference in the comparison is achieved by the closed loop. Figure 2.9 shows a functional diagram of the system.

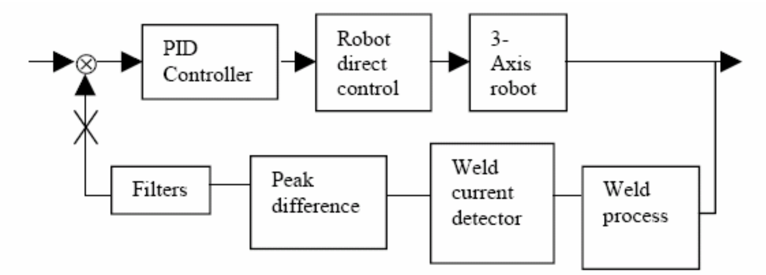


Figure 2.9 – Functional diagram of a closed-loop seam tracking system by Jieyu et al [66]

Murakami et al [71, 72] tested a FLC by developing three experiments based on FLC with a fuzzy filter, FLC without the filter and a PI control. The authors concluded that the FLC with the fuzzy filter produced better results than the FLC without the filter and in turn better that PI controller. From the comparison with the PI controller, the authors stated that the PI bead was a little irregular in the end part of the work and moved a little in zigzag direction along the path. No explanation was given by the authors for this PI controller behaviour. Excessive overshoot is usually caused by a low Integral value [69] and may have caused the poor performance.

To determine the control variables and rules definition for fuzzy logic controllers based on the experience of the welder, a great amount of experimentation is required [73]. To overcome this problem, Kim Y.S. et al [73] proposed a PI type fuzzy controller using Murakami et al [71, 72] and Kim J.W. et al [74] models but considering the noise that was previously filtered using the fuzzy filter. According to the author, the previous noise filtering made the system insensitive to errors. Simulation and experimental data was used to tune the PI parameters for the fuzzy rules.

An application of a FLC and ANN for seam tracking was proposed by Eguchi et al [75]. The groove angle used was 60º with open gap and no backing-plate. A trained ANN controller was used to estimate arc length and wire extension based on 50 ms readings of current, voltage and wire feeding rate at a sample rate of 1 kHz. Every 5 ms, the FLC processes the actual torch position with the arc length and wire extension calculated by the ANN controller and generates outputs to the wire feeder, power source and torch manipulator for seam tracking. The combination of an ANN controller and FLC revealed good results on the interaction.

ANN is widely used for pattern matching or recognition. Wang et al [76] used a taught ANN controller with experimental results of deviation patterns in the welding current. The resulting algorithm could detect the torch displacement by similar experimented patterns and correct the torch path applying what was learnt by the system in each particular case. Another application of an ANN controller was proposed by Ohshima et al [77] using both the welding current and voltage signals with CV GMAW process. From the experiments, the authors showed that although on a minor scale compared to welding current variations (≈50 A variation), voltage variations (<1 V variation) could be sensed and used successfully for seam tracking. To achieve the desired performance, much training data is required. To avoid this problem, the authors relied on numerical simulations from partial differential equations on heat conduction.

Although very reliable, consistent and flexible, FLC and ANN controllers have to be trained to perform well. Training is the method of filling in a database of hypotheses and respective output behaviours accordingly. The more data is used in the training, the more reliable and accurate is the system. This learning process is inconvenient when the system needs to be adapted to a new situation due to the fact that it is a time and resource consuming retraining process. As a way of simplifying this process, Kim et al [74] developed a self-organising FLC where basic common sense rules were the initial defined rules. The system uses two fuzzy variables obtained from the error between the welding current reference and a curve-fitted reference, with the integral of the error. A learning algorithm then improves the initial set of rules based on these new calculated values. For a consistent angular error of the torch trajectory the system has shown good results. For new torch trajectories with changes in the angular error, the process needs to restart. In other words, the application of this system for pipeline welding would show lower cross-seam control accuracy for the initial welding passes with improvements in the preceding ones. Unfortunately, it is not possible to perform an experimental weld for system training every time the welding head is shifted to a new pipe joint.

2.2.1.4 Process modelling

Wells [78] proposed two mathematical models for cross-seam and groove width control on the wide 90º V-groove:

1) differential model - equations (2.11) to (2.14)
2) template matching model - equations (2.15) to (2.17)

The differential model is based on sampling arc current or voltage, depending on whether a CC or CV power supply is used, at three positions of the torch excursion (left dwell, centre and right dwell). The developed models were based on the arc steady-state models from Halmoy [79] and Lesnewich [80]. Left and right dwells are used for the cross-seam control mechanism while the centre values are for CTWD adjustments. Signal acquisition is only performed when the torch passes over the three positions. Torch oscillation width control was also performed.

$$d_{ce} = -c_1 I + c_2 \tag{2.11}$$

$$\Delta x = -\frac{c_1}{2}\tan\left(\frac{\phi}{2}\right)(I_r - I_l) \tag{2.12}$$

$$\Delta y = c_1(I_0 - I_c) \tag{2.13}$$

$$\Delta w = -c_1 \tan\left(\frac{\phi}{2}\right)\left[(I_{r2} + I_{l2}) - (I_{r1} + I_{l1})\right] \tag{2.14}$$

where

d_{ce} torch-to-work distance for consumable electrode process (mm)

I steady-state arc current (A)

c_1, c_2 constants from process variables

Δx cross-seam off-centre error (mm)

Δy CTWD error (mm)

Δw oscillation width error (mm)

Φ V-Groove angle

I_r arc current from right-hand torch position (A)

I_l arc current from left-hand torch position (A)

I_c arc current from centre torch position (A)

I_0 desired arc current (A)

I_{r1}, I_{l1}, I_{r2}, I_{l2} arc current from right-hand and left-hand torch positions at consecutive torch oscillations (A)

The template matching model is achieved by continuously sampling the arc signals and by comparing them to a mathematical template based on a parabolic equation. The template matching model assumes that a parabola shape is made by the joint and weld pool (Figure 2.10). A difference curve is defined as the difference between the template and the profile, expressed by equation (2.17). The parabolic template fits a concave groove shape or previous bead. Convex shapes may not work properly with the proposed model.

$$I_t(x) = k_5 x^2 + I_v \quad (2.15)$$

$$I_p(x) = k_5 x^2 - 2k_5 x_0 x + k_5 x_0^2 + I_0 \quad (2.16)$$

$$I_t(x) - I_p(x) = 2k_5 x_0 x + I_v - I_0 - k_5 x_0^2 \quad (2.17)$$

where

I_t	template arc current (A)
I_p	profile arc current (A)
x	torch actual position
x_0	actual cross-seam centre of oscillation
I_0	arc current when torch is at x_0 (A)
I_v	desired arc current when torch is located directly above joint centreline (A)
k_5	constant determined by the joint geometry

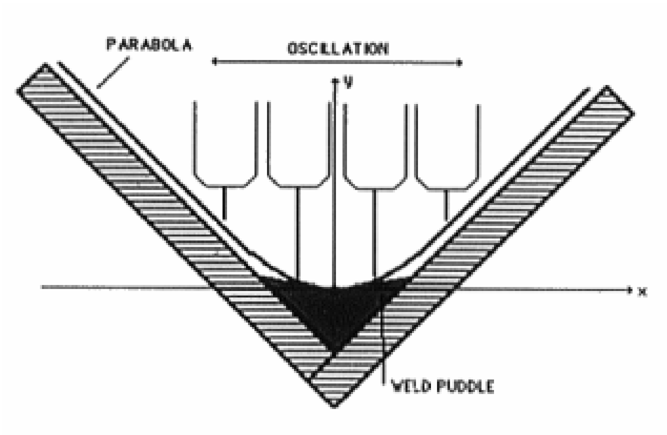


Figure 2.10 – Template-matching model using a parabola by Wells [78].

The developed system based on the models averages arc signals using very low pass cut-off filters (< 10 Hz). Experimental torch oscillation speeds were below 2 Hz. The control approach made by this author has revealed good potential and can be summarised as:

1) cross-seam control by arc signal comparison between torch maximum excursions
2) CTWD control by arc signal comparison with a predefined value when torch is at centre of oscillation

3) torch oscillation width control by arc signal consecutive comparisons at torch maximum excursions between complete oscillations

2.2.1.5 High-Speed rotational torch

For high speed rotational torch seam tracking systems, several developments have been described [19, 48, 52, 81-83]. These systems work similarly to torch weaving methods described previously with the use of an encoder to detect when the contact-tip passes at each extreme position across the seam. Due to the fixed rotation width, this system is not able to perform the control for changes in groove width. FLC was also used similarly for torch guidance control [25, 37].

2.2.1.6 Summary

As a summary, many systems were developed for CV and CC GMAW open groove welding as described, and more can be found in the published literature based on the methods presented. The state-of-the-art of torch guidance on open groove welding was reviewed showing the strong and weak points as well as limitations of each type of technique used for control.

2.2.2 Pulsed GMAW open groove

Through-the-arc sensing with GMAW-P for torch guidance control, in contrast to its mature predecessor CV and CC GMAW, is still being studied by researchers. Basically, the approach performed so far is based on the same principle of torch oscillations across the seam to produce arc length changes and therefore arc signal changes. As referred previously, GMAW-P is a CC based process and arc voltage is commonly used for control.

Barnett et al [60] developed a solution using a data acquisition system sampling the welding signals plus a torch axial displacement transducer at 3 kHz sampling rate. In order to reduce computational time, the data was reduced by a factor of 30 to an equivalent 100 Hz sampling rate. The experiments were conducted in the 90º V-groove and lap welds. Two methods were then employed:

1) Finding the difference between peak and background values of the pulsed voltage at torch maximum excursions
2) Integration of the welding voltage during the torch travel from centreline to maximum excursion

The authors stated that the sampled pulse background voltage (minimum background voltage of every pulse) has less noise content and better signal regularity than the pulse peak voltage and it is more appropriate as an input for a seam-tracking control system. From the published data by the authors, it is apparent that the voltage signal from the experiments is high with the absence short-circuiting, reflecting the use of a long arc length procedure.

A different approach was proposed by Di Pietro et al [59], using the pulsed welding current filtered by a Butterworth low pass filter and analysed by power spectral densities (PSD). These authors found a linear relation between the resultant waveform and the torch off-centring amount. When aligned with the centreline, the resultant peak frequency from the PSD coincides with the torch oscillation frequency. Introducing a known torch off-centring amount reduced the dominant frequency from the PSD analysis to half of the torch oscillation frequency. No specific cross-seam control technique was proposed by the authors to implement the above described method.

Based on a sidewall-matching algorithm patented by Cook et al [84], Rashid et al [85] used a FLC and a set of rules to implement cross-seam control for GMAW-P. The sidewall-matching algorithm is based on averaging the arc voltage signal and comparing the averaged values of both torch oscillation maximum excursions, as proposed by the author for CV GMAW [84]. The same technique is proposed by Rashid et al [85] for GMAW-P. The resultant voltage difference from the comparison is then supplied to the FLC that computes the necessary amount of torch path correction to be passed to the motion system. The experiments were conducted on fillet welds.

Similar proposals for cross-seam control were found for GMAW and GMAW-P for the open groove and fillet welds although only one commercial solution for GMAW-P was found to date [86] for pipeline welding, and operational details were not disclosed. The wide angle between sidewalls of the open grooves produces a clear variation in the arc length and hence in arc signals, optimum to achieve good cross-seam control.

2.2.3 Narrow groove welding technique

Narrow groove welding (NGW) is a technique commonly used in arc welding to weld thick walled sections of metal with the sidewalls almost parallel [87, 88]. The filling process is based on placing one bead on top of the previous one until the gap is filled. This technique is preferred to the conventional V-groove since the reduction in deposited material makes the welding process faster and cheaper.

In Japan during the eighties, NGW was reviewed by the Sub-committee on Weld Metal & Welding Procedures of JPVRC as a technique to weld thick plates of more than 30 mm thickness [89]. A good review can be found by Modenesi [90]. This technique is also being used in smaller plate thicknesses [87, 88, 91, 92] and especially in the pipeline industry, typically using a 5° bevel angle with pipe wall thickness of 19 mm and above.

A major problem in utilising NGW is lack-of-sidewall-fusion defects. This is due to low or no angle between sidewalls limiting the side metal penetration. Several different methods were developed in an attempt to eliminate this problem. These methods include:

- Wire methods
  - Twist-arc (Figure 2.11)
  - Babcock-Hitachi K. Type (BHK) (Figure 2.12)
  - Corrugated wire (left of Figure 2.13)
  - “Rotating arc” (right of Figure 2.13)
  - Loop nap (Figure 2.14)
  - Bent contact tip (Figure 2.15)
- Other methods
  - High Speed Rotating arc (Figure 2.16)
  - Electro-magnetic arc oscillation (Figure 2.17)
  - Mechanical torch oscillation (Figure 2.18)

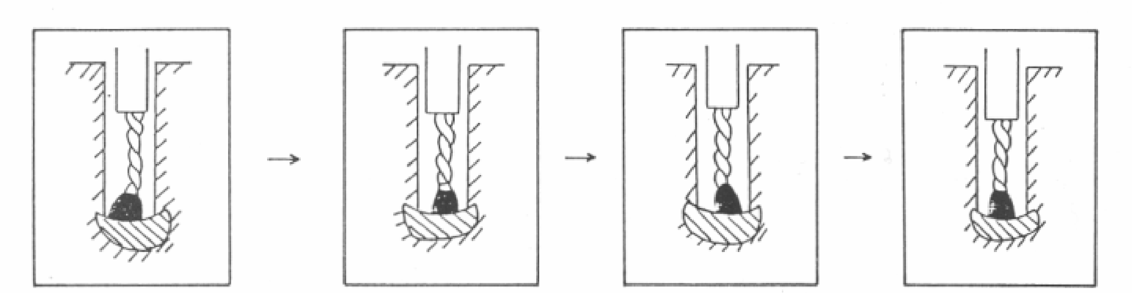


Figure 2.11 – Rotational movement of welding arc produced by Twist-arc [93]

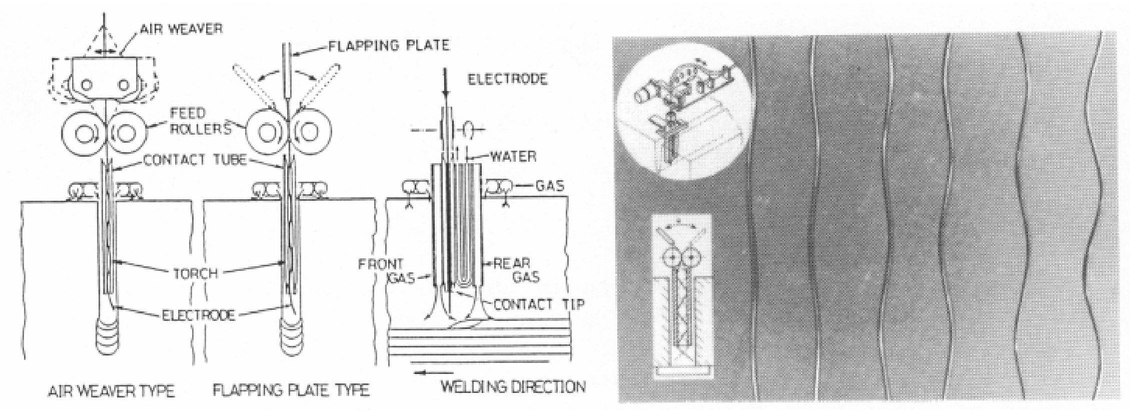


Figure 2.12 – Wire waving mechanism (left) and the waved wires (right) [94]

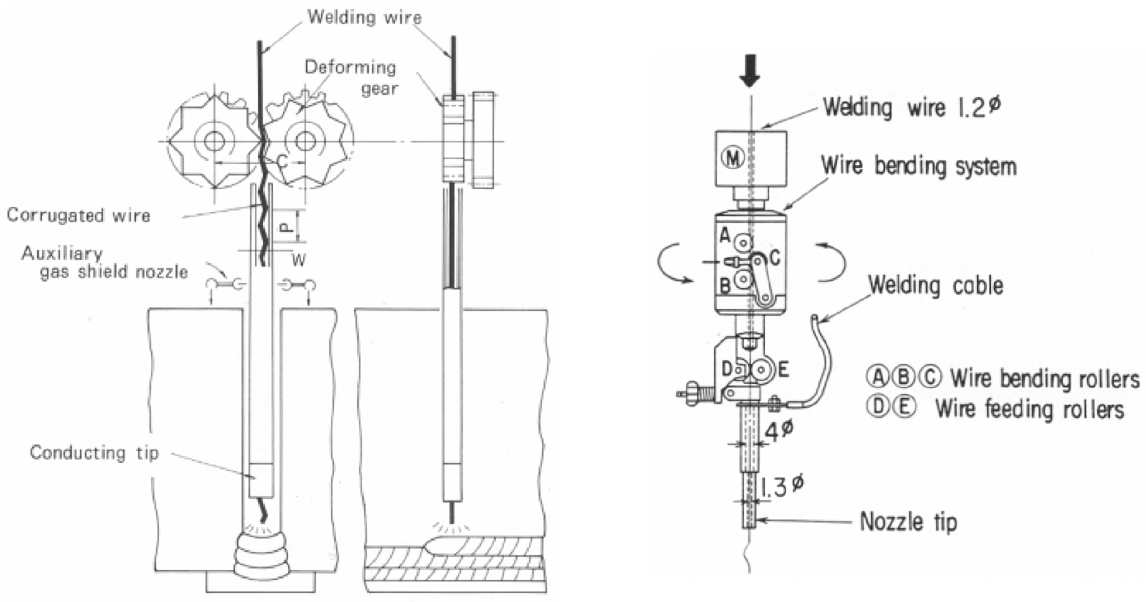


Figure 2.13 – Corrugated wire (left) [95] and "Rotating Arc" (right) [51]

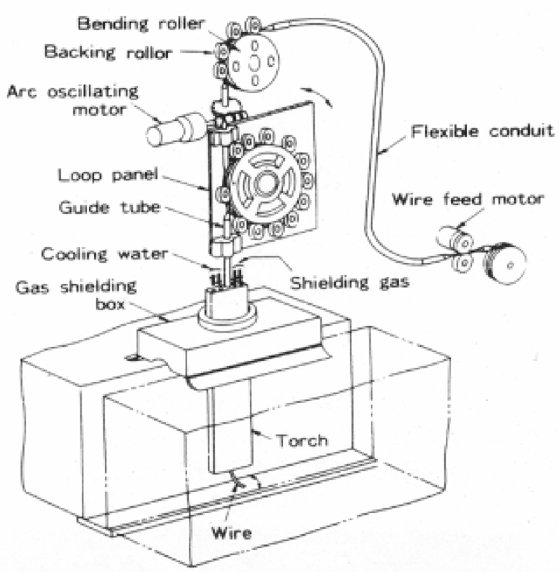


Figure 2.14 – Loop Nap method mechanism [96]

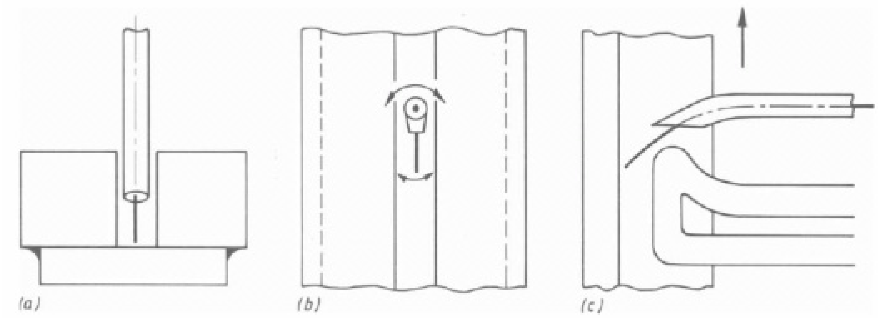


Figure 2.15 – Bent contact-tip: (a) end view, (b) top view, (c) side view including copper bend weld pool support [87]

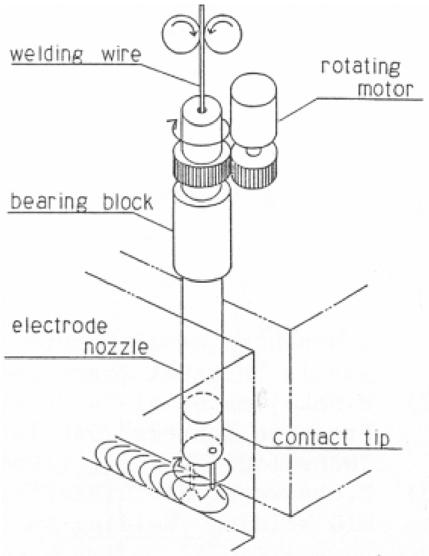


Figure 2.16 – High Speed Rotating Arc principle [50]

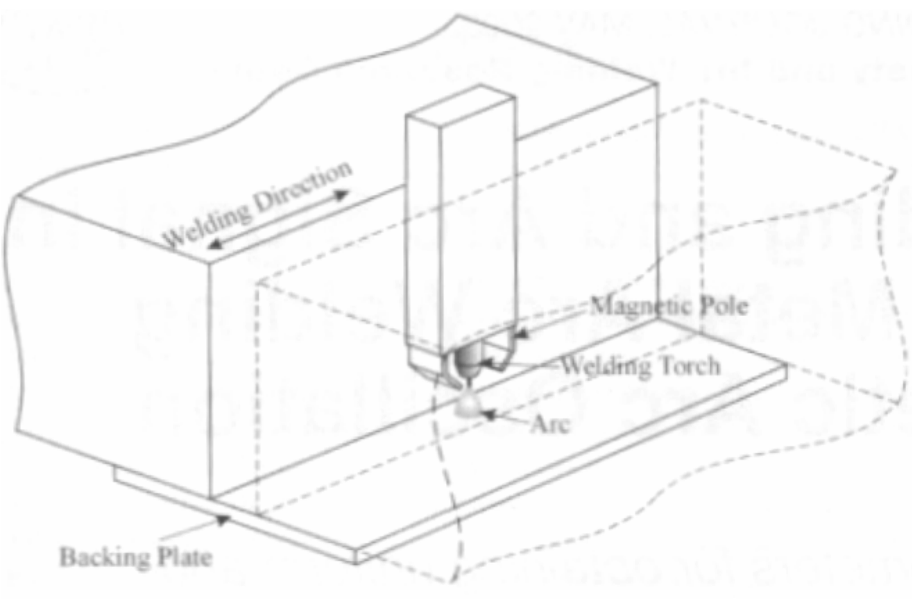


Figure 2.17 – Electromagnetic Arc Oscillation [88]

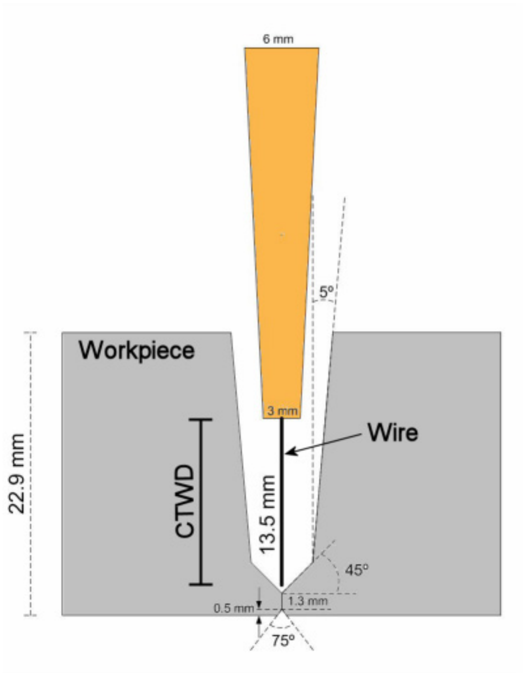


Figure 2.18 – Scaled figure of a narrow contact tip in position for the 1st pass in the 5° bevel narrow groove for mechanical torch oscillation

Cross-seam control is essential in most cases not only to achieve good welds but also because the welding tool is inside the groove with tight margins with the sidewalls and low tolerances for errors. The robustness and low cost implementation of through-the-arc sensing systems is capable of providing major improvements in control of narrow groove welding.

A cross-seam control system proposed for through-the-arc sensing for narrow groove welding is the high speed rotating arc [50] with GMAW. The working principle is the same as that explained previously for the CV GMAW open groove. Due to the torch concept, the wire tip and therefore the arc can be very close to the sidewall, enough to produce arc signal variations for control. On the open groove, the signal variation is smoother because it follows the groove inclination while in the narrow groove it is more abrupt.

A variation of the rotating torch is the rotating/oscillating torch with bent contact-tip proposed by Eichhorn et al [38]. The bent contact-tip performs semi-complete rotations around its main axis creating an oscillating effect across the seam. The welding process used was GMAW-P. The angle of rotation of the contact-tip is predefined and may vary accordingly with the groove width changes detected by the arc sensing system. The system self-adjusts torch travel speed in case of groove width changes to compensate for deposition rate. An electronic circuit finds peak currents from the pulsed arc current signal and feeds them into a computer for further processing. It is not clear from the published work how the signal is then processed to perform the cross-seam control. This system has good potential for improved penetration in a narrow groove due to the rotating bent contact-tip although the experimented values used were too slow for pipeline welding applications (contact-tip rotating speed below 1 Hz and travel speed was 0.11 m/min).

Baba et al [97] developed a through-the-arc sensing cross-seam control system for narrow groove pipe joining based on the Loop Nap technology. The welding process used (named Pilot Arc Welding) is a CV GMAW process with a switching circuit to produce a peak and a background current synchronised with the wire tip oscillation. When the bent wire reaches the sidewall, the current rises to its peak value and is lowered to the background value during the travel from one sidewall to the other. Oscillation frequency is about 0.66 Hz for flat positions and 0.55 Hz for vertical and overhead positions. The arc current was used for cross-seam control. The signal was acquired through a shunt resistor and filtered with a very low band pass filter (cut-off frequency at 1.5Hz). An ADC in a micro-computer was used to take 16 samples of arc current when the arc is at peak current (wire tip near the sidewall). This value is stored and compared to the previous acquired average when the wire tip was in the opposite sidewall. The difference is processed to adjust the torch path accordingly.

Since the system was developed to work in an orbital solution, the weld pool suffers influence from gravity forces when in vertical and overhead positions. This means that the torch height (CTWD) decreases when going from flat horizontal to vertical and overhead position. The authors prepared the algorithm with a compensation variable according to the torch inclination around the pipe.

Its coefficient value was determined by experimentation and regression analysis. Unfortunately this system produces low deposition rates for the actual demands of the pipeline industry and the Loop Nap technology is quite complex with very sensitive moveable parts.

2.2.4 CTWD control

The importance of a steady CTWD and therefore arc length has been known for some time [18, 98, 99]. Better bead shapes with improved quality are attained when this control is well performed.

In the development of cross-seam control, some researchers have also approached CTWD control, with their systems having both controls working together. In some other cases, this control was studied and developed as a stand-alone process using similar controllers found for cross-seam control such as ANN and FLC. Yamamoto et al [100] developed an ANN for estimation and real time measurement of arc length and wire extension, based on arc physics and ANN training. A change of wire type or shielding gas demands network retraining.

Using short-circuiting resistance measurements on dip transfer in GMAW pipeline welding, Di Pietro et al [101] proposed a system for stick-out control to compensate pipe ovality. When arc voltage drops below 5 V and arc current rises above 50 A, the short-circuit cycle condition is triggered and the resistance is then measured and averaged. If the calculated value exceeds a percentage deviation from the reference resistance value, the control corrects the torch position until the difference is inside the control limits. Also Di Pietro et al [59] proposed CTWD control for GMAW-P by analysing the pulse frequency variations during welding. A Fast Fourier Transform (FFT) analysis of the pulsed arc signal has shown that at different CTWD values, different pulse frequencies were attained. This relationship makes it possible to achieve CTWD control. It was found that this behaviour was produced by the power supply adaptive mode system to self-control the arc length. The successful implementation of a CTWD control based on this principle is dependent on the power supply manufacturer and in the way the self-control of arc length is implemented.

2.2.5 High Speed Oscillation

High speed oscillation is a technique that incorporates some of the technologies already reviewed like the rotating arc and pendular oscillation. Although there is no clear definition of what is considered high speed oscillation, some authors consider it to require frequencies above 5 Hz. This technique is claimed to produce steady and even sidewall melting [102] with improved characteristics of the weld bead [103, 104] and more sensitivity for seam tracking [53], using CC and CV GMAW processes. No published information was found in the application of high speed oscillation for GMAW-P. For CO_2 CV GMAW process, torch oscillation frequencies between 5 Hz and 15 Hz were found to produce better results in bead shape appearance and lower spatter generation [103].

Arc signal sensitivity in high speed oscillations is the major benefit pointed out by many researchers for cross-seam and CTWD control systems. It is based on the larger variations of arc signal found with higher torch oscillation frequencies caused by the dynamic-state of the arc process in GMAW. The arc does not have time to accomplish its self-regulation mechanism to achieve the equilibrium point with the stick-out, hence arc voltage and/or arc current variations are more accentuated. The High Speed Rotating Arc is the most researched method that takes advantage of this natural mechanism of GMAW. Mechanical limitations of weaving mechanisms on achieving high speeds was presented by Jeong et al [52] when comparing sensitivity between the rotating arc system with the conventional weaving system.

On the other hand, rotating arc mechanisms usually have fixed oscillation widths. To tackle this problem by taking advantage of the higher arc signal sensitivity of high speed oscillation, some authors explored different ways of performing arc oscillation. Kodama et al [105] developed a torch where the contact tube is fixed to a pivot performing a pendular weave by means of two solenoid coils and permanent magnets, with oscillation frequencies of up to 40 Hz. Another system using electromagnetic arc deflection oscillated the arc at a frequency of 30 Hz [106]. A patent from 1978 was found [107] with a device that converts through a cam a rotary movement from a motor into an oscillating motion connected to a GMAW torch. With the use of gears, very high oscillation frequencies can be achieved with conventional DC motors.

2.3 Summary

This literature review has attempted to show the state-of-the-art of actual through-the-arc sensing methods and techniques to perform welding automation. The application of arc sensors have been mostly developed for open groove and fillet welds using mainly CV and CC GMAW processes, although some reports were also found on the use of GMAW-P for the open groove.

The application of the narrow groove technique on thick-walled components has shown visible savings in time and costs of production, and it is being widely applied. Some methods were reviewed and most literature was found on the application of the high speed rotating arc using CV GMAW, with through-the-arc sensing seam tracking technology. The high speed rotating arc takes advantage of the higher arc signal sensitivity for control found in GMAW due to the torch high speed oscillation/rotation. No study was found on arc signal sensitivity for GMAW-P though.

High speed oscillation is a technique also reported as producing improved characteristics of the weld bead for GMAW. For narrow groove GMAW-P pipeline applications, the influence of this technique in fusion profile and bead shape is still unknown, as well as the optimum value of torch oscillation width for a particular groove width. Arc signal signatures in both cases are also important.

A clear gap was found in the literature on the application of arc sensing for narrow groove GMAW-P, for single and tandem welding procedures, in order to achieve torch position control for pipeline welding applications. The aim of this research was to develop a greater understanding of the factors influencing GMAW-P arc signals when welding inside a pipeline narrow groove, and to develop a set of algorithms that could guide a single wire or tandem welding oscillating torch along the groove path in order to achieve consistent and defect free welding.

3 Experimental methods

This project was aimed at evaluating the use of GMAW-P signals to achieve torch positioning control in narrow groove pipe welding. The welding trials were divided into three different sets, using different equipment for part or all of each set. This was mainly due to the specificity of each experiment that was performed. The first set of trials served to develop the basic concepts that were used subsequently for the rest of this project. The second set of trials was performed to methodically test the developed control software described in chapter 4. The third set of trials was performed to better evaluate arc signals for groove width detection.

These three experimentation sets will be referred to in this thesis as Experimentation phase 1, 2 and 3. They all use of a power supply, shielding gas, welding torch and a motion system. For signal acquisition a digital oscilloscope was used and in some trials the arc was filmed using a high speed video camera. For experimentation phase 3 the welds were also examined using standard metallographic procedures.

3.1 Experimentation phase 1 – Initial trials

The initial trials were aimed at analysing and evaluating through-the-arc sensing for torch position control in narrow groove pipe welding using GMAW-P. The results of this experimentation were used to achieve torch height and cross-seam position control based on the use of arc signals. A set of experiments was conducted where combinations of torch oscillation frequency and width were varied to produce different arc signal behaviours for subsequent analysis.

3.1.1 Equipment, materials and experimental procedure

The equipment used for the conducted trials was based on a commercial pipeline welding system, typical of the ones used in the field. It comprised a basic pipe welding setup made of a power supply, wire feeder, shielding gas supply, and a welding head with a single wire torch.

A Lincoln pulse power supply, Power Wave F355i, was used in constant current pulse non-adaptive mode. This power supply is designed for direct digital operation in robotic and automated systems, and has no external controls. Control signals are provided via Ethernet. Commands, status and pulse wave shapes are transmitted from a computer, using Lincoln “Wave Designer Pro” software [108]. This enables the user to graphically define all the parameters needed for the shape of the pulse wave in the synergic curve appropriate to the desired wire feed speed (Figure 3.1). The wire feeder, Lincoln model Power Feed 10 Robotic, is connected to and controlled by the power supply.

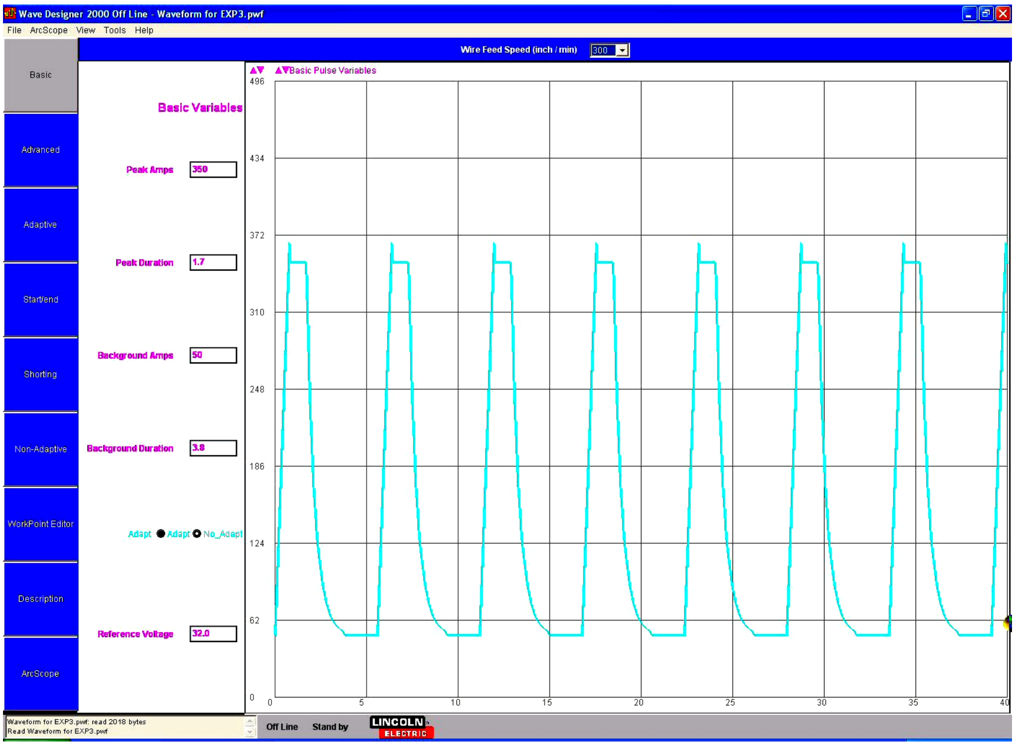


Figure 3.1 - Pulse wave shape generated by Lincoln Wave Designer Pro

The motion solution is from RMS Canada; model RMS MOW II (Figure 3.2). It comprises a welding head with an oscillator, a control device and a pendant, all connected via a CAN network at 512kbps baud rate. The oscillator creates a pendular movement on the torch, however since the pivot is displaced 150 mm from the end of the contact tip, and the weave width is less than 10 mm, the oscillation can be considered linear.

A series of initial trials were designed to evaluate the sensitivity of the system to variations in torch / wire position, and to provide the basis for algorithm development for control of cross seam position and CTWD.

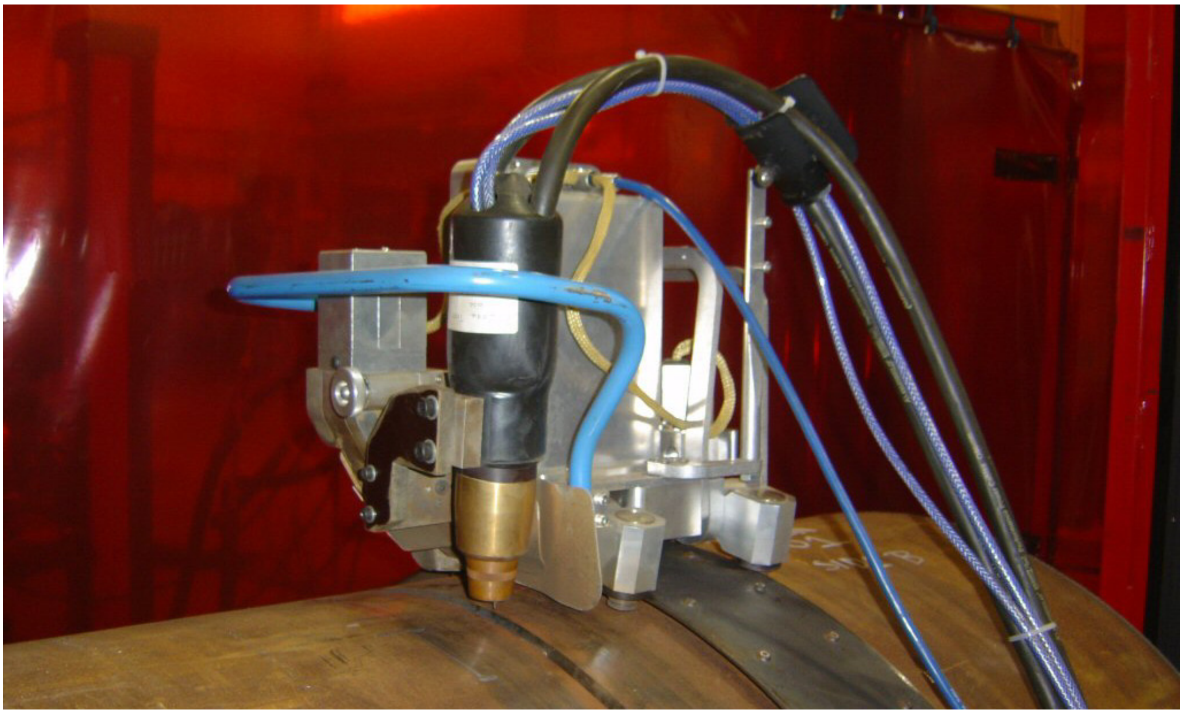


Figure 3.2 - RMS MOW II single torch pipeline welding head on pipe

Two lengths of 0.91 m diameter by 19 mm wall thickness API5L:X80 steel pipe joined by a root pass were used in the welding trials. The weld preparation used for the experimental trials (Figure 3.3) was typical of pipeline tie-in welds, with a 15° bevel, rather than 5° bevel often used on narrow groove mainline welds. All welds were performed to simulate the first fill pass in a pipe joint, and the pipe was rotated slightly between welds, so that all welds were in the flat position.

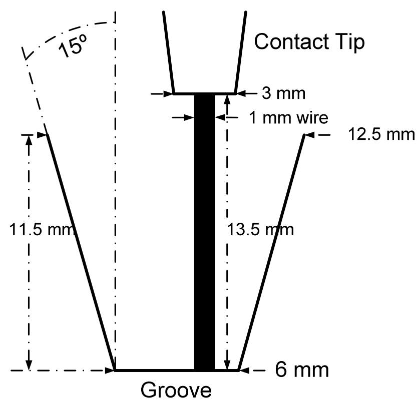


Figure 3.3 - Weld preparation profile for experimentation phase 1

The base welding conditions were as follows:

Welding process	**GMAW-P**
Peak current	**350 A**
Peak time	**1.7 ms (including ramp up time)**
Background Current	**50 A**
Background time	**1.8 ms**
Frequency	**181 Hz**
Wire feed speed	**7.6 m/min**
Wire	**Carbofil NiMo1, 1mm diameter**
CTWD	**13.5 mm**
Shielding gas	**BOC Tri-mix (5%He, 12.5%CO_2, 82.5% Ar)**
Gas flow rate	**20 l/min**
Travel speed	**0.38 m/min**

Signal acquisition was performed with a Yokogawa Oscilloscope ScopeCorder DL750 with 1:10 ratio voltage probes and LEM PR1030 Hall-effect current probes. Sample rate was 10,000 samples per second (10 KHz) with 400Hz digital filter to avoid signal aliasing.

Using the system described, the following series of trials were conducted to examine the effects of torch oscillation rate and oscillation width as described in Table 3.1.

Table 3.1 – Experimentation phase 1 trials definition

Trial	Oscillation Frequency (Hz)	Oscillation Width (mm)
A1	3.33	5
A2	3.33	6
A3	3.33	4
A4	5.0	5
A5	1.67	5
A6	0	0

3.2 Experimentation Phase 2 – Control algorithms test bed

These trials served as a test bed for the developed CTWD and cross seam position control software described in section 4.1. Known band misalignments with the groove centre were introduced to enforce torch path corrections by the welding head through the control algorithms. These positional corrections along with arc signals were recorded for subsequent analysis and are reported in section 5.2.

Experimentation phase 1 and control algorithms development trials used single wire GMAW-P Lincoln Power Wave F355i power supply. The data recorded from these trials was for development purposes only. For the purpose of testing control algorithms reliability in correction speed and accuracy, and also to test robustness and adaptability of the algorithms to a new situation, the welding equipment and procedure were changed. For phase 2, tandem welding trials were conducted with two GMAW-P synchronised Lincoln Power Wave F455M power supplies, using different pulse parameters from phase 1. The changing of the pulse parameters was made for two reasons:

a) to test the robustness of the algorithms on a new situation
b) to use a newly developed welding procedure for pipeline welding

Welding head, pipe configuration and material were maintained from previous trials.

3.2.1 Equipment, materials and experimental procedure

Two lengths of 0.91 m diameter by 19 mm wall thickness API5L:X80 steel pipe joined by a root pass were used in the welding trials. The weld preparation used for the experimental trials (Figure 3.4) was the 5° bevel often used on narrow groove mainline welds. Figure 3.5 shows the welding station setup for the experiment.

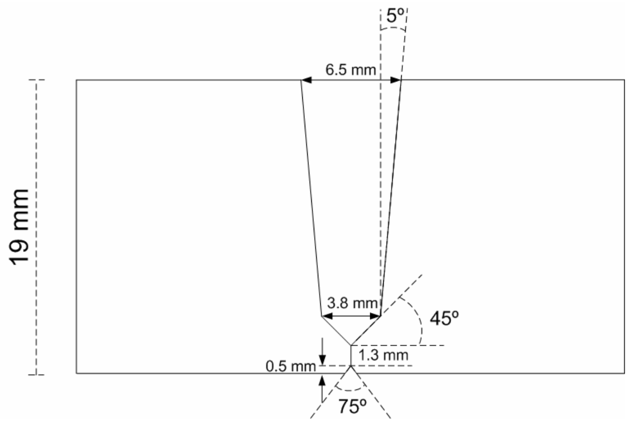


Figure 3.4 – Narrow groove used for trials B1 to B4

Figure 3.5 – Welding station for experiments B1 to B4 with two GMAW-P power supplies, pipe and RMS MOW II welding head on the pipe band

The power supplies used were the Lincoln Power Wave 455M. The tandem GMAW pulse shapes are shown in Figure 3.6. The oscillation width values presented were obtained and calibrated to a CTWD of 13.5 mm.

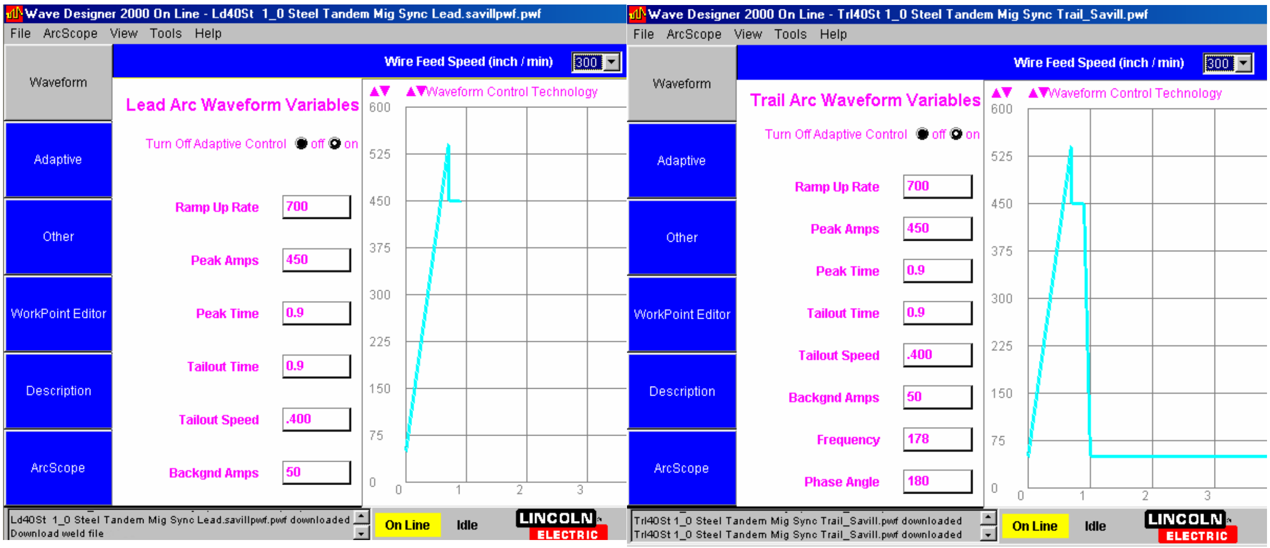


Figure 3.6 – Lead and trail pulses for single and tandem wire trials

The base welding conditions for both wires were as follows:

Welding process	**GMAW-P**
Peak current	**450 A**
Peak time	**0.9 ms (including ramp up time)**
Background Current	**50 A**
Background time	**4.7 ms**
Frequency	**178 Hz**
Wire feed speed	**7.6 m/min**
Wire	**Carbofil NiMo1, 1mm diameter**
Shielding gas	**BOC Tri-mix (5%He, 12.5%CO_2, 82.5% Ar)**
Gas flow rate	**20 l/min**
Workpiece base metal	**X100 HSLA Steel**
CTWD	**13.5 mm**
Torch oscillation frequency	**8.33 Hz**
Travel speed	**0.76 m/min**

Table 3.2 describes the setup of each trial. Horizontal and vertical band to groove centre misalignments are described in the table as horizontal and vertical deviations. The torch has to move to the right to follow the groove centre, following the torch direction. The “Set Voltage” parameter is needed for the control algorithms and it is explained in section 4.1. The welding length was 120 mm.

Table 3.2 – Experimentation phase 2 trials definition – Narrow groove welding

Trial	**Horizontal deviation (mm)**	**Vertical deviation (mm)**	**Oscillation Width (mm)**	**Set Voltage (V)**
B1	4.8	0.5	2.5	22
B2	4.8	0.5	2.5	20
B3	4.8	2.8	2.5	22
B4	4.8	2.8	6	22

For the analysis of trials B1 to B4, it was necessary to obtain the relationship between averaged arc voltage and CTWD. Table 3.3 shows the trials used to generate this data in bead on pipe trials.

Table 3.3 – Experimentation phase 2 - Voltage / CTWD – Bead on pipe trials

Trial	CTWD (mm)
B5	**11.5**
B6	**12.5**
B7	**13.5**
B8	**14.5**

Signal acquisition was performed with a Yokogawa Oscilloscope ScopeCorder DL750 with 1:10 ratio voltage probe and LEM PR1030 hall-effect current probe. Sample rate was 10 KHz with 4 KHz digital. Also, torch positional messages sent by the control algorithms were logged for subsequent analysis of torch position corrections. Section 4.1 describes the messaging system in detail.

3.3 Experimentation Phase 3 – Torch oscillation width and frequency

Torch positional control using through-the-arc sensing for the narrow groove with GMAW-P was at this point accomplished for torch height and cross-seam control. Two important factors were still to be analysed: the influence of torch oscillation width and torch oscillation frequency on arc signals sensitivity. This third experimentation phase was devised to study arc signal behaviour in order to understand the factors that contribute to signal variation in time with different torch oscillation widths and frequencies. This phase of experimentation was needed to generate required information for control of weave width.

3.3.1 Equipment and materials

This experimentation phase required the development of a new welding station with a fixed torch and a moving table (Figure 3.7). This experimentation required a different approach not possible with a moving welding head. A moving table enables more accurate control over the motion system and the use of a high speed camera positioned in front of the table. High speed images of the arc and the space around allow the evaluation of metal transfer and arc behaviour (size, shape and movement) in relation to the distances between the arc and sidewalls.

The welding equipment setup was the same as used in experiments A1 to A6 for gas, wire, power supply and pulse parameters. Signal acquisition was also performed by the same digital oscilloscope and voltage/current probes. The main difference was in the motion system with the introduction of the moving table and the newly developed high speed oscillation torch. In terms of data acquisition, the differences from previous experimentation included the use of a high speed camera and a Linear Variable Differential Transformer (LVDT) sensor attached to the contact tip extension tube. This sensor returns a voltage signal indicating the contact tip physical oscillation position. This signal was acquired by the digital oscilloscope synchronised with arc signals. In this way, arc signals can be referenced with the torch weaving position.

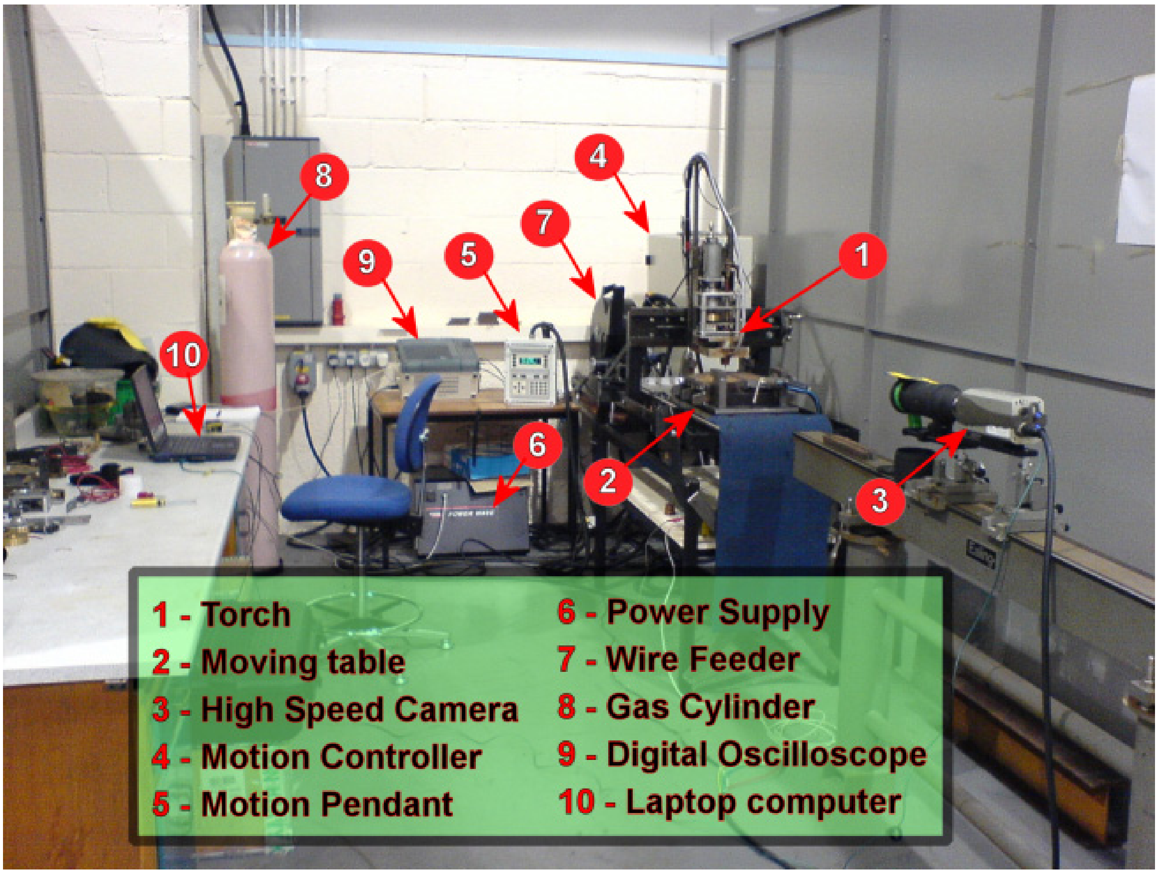


Figure 3.7 – Welding station for experimentation phase 3

3.3.1.1 Moving table and motion system

The moving table has a flat base of 530 mm x 300 mm x 20 mm made of steel that is mounted on two parallel stainless steel shafts of 1.2 m x Ø 27 mm, by means of roller bearings. A third parallel and central shaft (worm) with 800 mm x Ø 21 mm has a thread of 5 mm spacing and crosses the flat base underneath through a worm bearing hole (right view of Figure 3.8). The rotational movement of this shaft moves the flat base forwards and backwards driven by a brushed DC motor. The rotation speed of the motor is monitored by an encoder. The motor power and encoder signals are controlled by a Trio ® Motion Controller device (MC216 with P300 axis expander). This device also controls the torch motors as described in the section 3.2.3. The table speed was calibrated and can achieve speeds from 0 to 1.5 m/min for a maximum length of 345 mm.

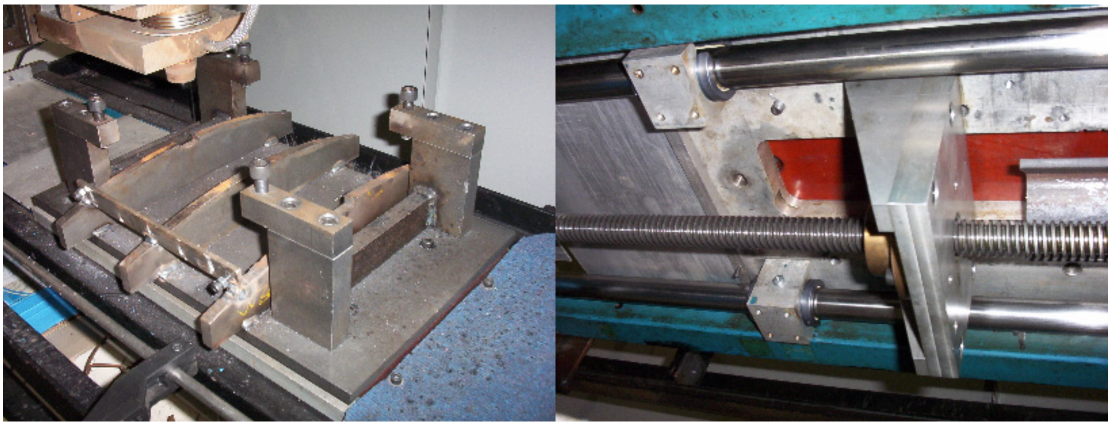


Figure 3.8 – Moving table top view (left) and bottom view (right)

An insulated jig attached to the table holds the plates to be welded (left view of Figure 3.8). This jig has three curved support beams made of steel with a central slot. The curved support beams have the same radius of curvature as a 1.32 m (52") diameter line pipe. The plates used for the experiments were extracted from 1.32 m diameter line pipe with 23 mm wall thickness. Four Ø 12 mm clamping bolts positioned in each corner hold the plates in position. Two bevelled plates form a groove. The plates are bevelled similarly to the line pipe bevels (5° bevel angle) to perform the same type of groove when fixed to the rig. Maximum dimensions for a single plate are 270 mm length, 290 mm width and 50 mm thickness.

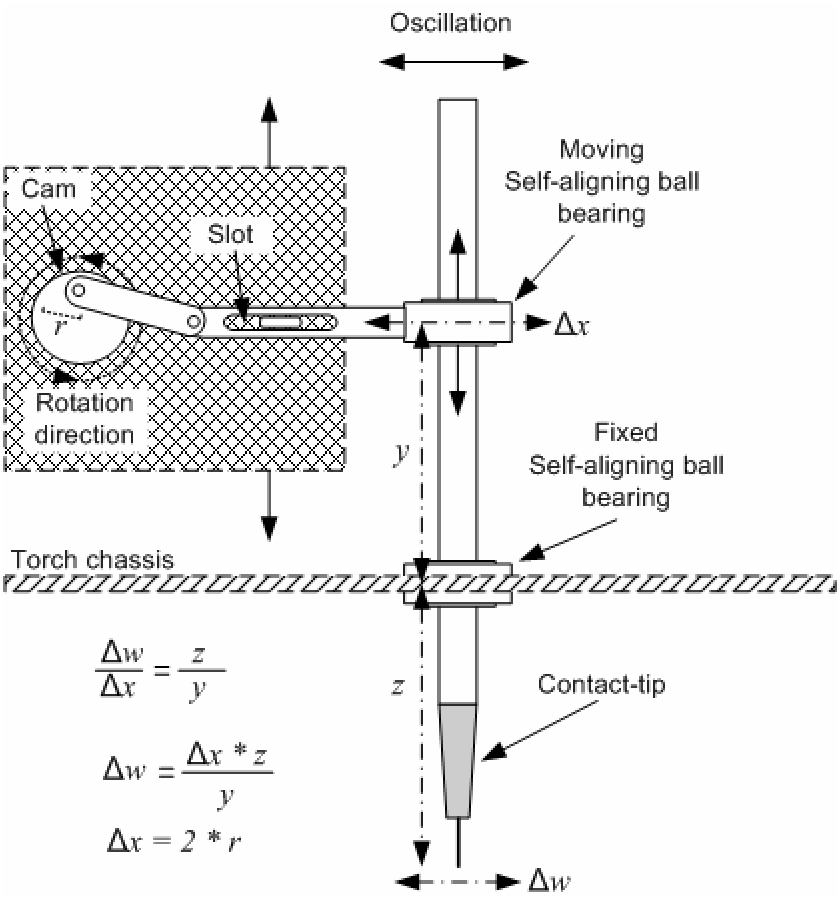
Attached to the motion controller is a Trio membrane keypad (P503) pendant with 37 buttons and 4 x 20 character display screen. The software developed to control the motion controller is in BASIC language specific for Trio devices and it is listed in Appendix F. The software allows plate travel speed and the number of runs per length to be adjusted. This means that one plate length can provide several short test welds. The majority of the experiments used 5 runs of 50 mm on a 250 mm length plate.

3.3.1.2 High speed oscillation torch

The idea to develop a new torch arose from the limitations on actual torch maximum oscillation speed of pipe welding head oscillators. No commercial system was found that could achieve the intended frequencies. Electric motors are typically limited in constantly reversing their rotational direction to frequencies usually lower than 10 Hz. Oscillation widths may also change with the increase of oscillation frequency due to the motor inertia on reverting direction.

To counteract this problem, engineers often use the cam principle to convert a rotational movement into a linear movement. Therefore, any oscillation frequency can be achieved just depending on the motor rotation speed limit (revolutions per minute) in one direction only. The disadvantage of this principle is the fixed oscillation width. The principle of a moving pivot on a lever oscillation was used to overcome this limitation. This principle is based on having the contact tip attached to a tube pivoted by a self-aligning ball bearing. In one side of the bearing resides the contact-tip and in the other, another bearing moves up and down the tube. This second bearing is then attached to a slider that is attached to the cam. The cam is rotated only in one direction by means of a motor. The cam diameter creates a fixed oscillation width and the up/down moving bearing changes that width proportionally. Figure 3.9 presents a diagram explaining the principle.

A 3,500 rpm motor produces a 58.3 Hz oscillation frequency. Depending on the motor torque and the force exerted by the cam system, gearing up the motor can produce higher oscillation frequencies. In this work, the motor used produced a maximum stable frequency of 25Hz and this defined the maximum working oscillation frequency for this torch.



Where

Δw	oscillation width of the wire tip
Δx	oscillation at the moving bearing
z	length from the wire tip to the centre of the fixed bearing
y	length from the centre of both bearings

Figure 3.9 - High speed torch principle

An identical principle applied to a high-speed weaving welding torch was found in a patent from S. Oshima in 1978 [107]. A picture of the developed torch for this project is shown in Figure 3.10, based on the principles shown in Figure 3.9. The second motor that moves the upper part of the torch up and down to change the oscillation width is not visible in the image. Both motors and encoders are controlled by the Trio Motion Controller along with the moving table.

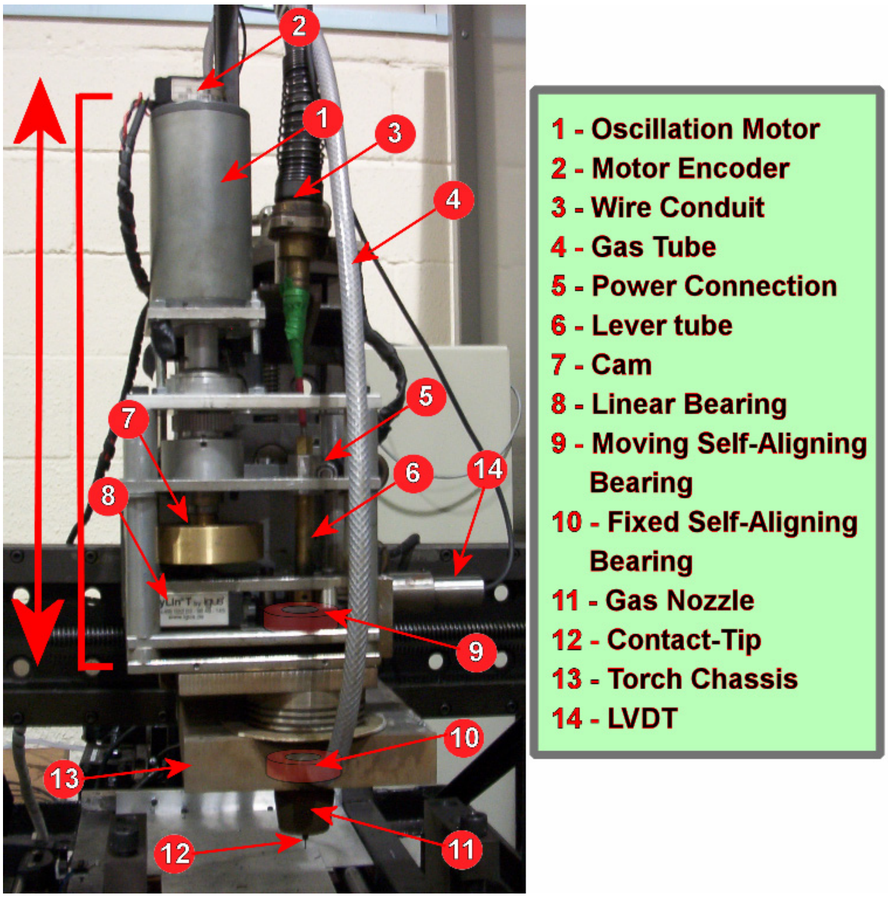


Figure 3.10 – Developed high speed torch

Another aspect of this torch when compared with conventional GMAW torches is how the oscillation movement is produced. For instance, the Cranfield tandem GMAW torch weights more than 2Kgs excluding cables. To accurately produce an oscillatory movement with a torch this size and weight, the oscillator has to exert a massive force making the whole welding head shake when high speed oscillations are demanded. In the case of the developed torch, the gas nozzle is steady and only the contact tip moves across the seam. This can also produce better gas flow if the nozzle is adapted to the circumstance.

3.3.1.3 High-speed video camera

The major advantage of the moving table setup is the ability to use a high-speed camera for arc image filming. Due to the fixed torch and moving workpiece, the camera can always have the same distance to the arc and is always in focus. The camera used for this experimentation was a Vision

Research Phantom ® V4.1 capable of acquiring 32,000 frames per second (fps). Although for image acquisition the unit is usually expressed in “frames per second”, it can also be expressed in frequency units - Hertz (1 fps = 1 Hz). To simplify the reading, frequency terminology will be used.

The camera sensor has a maximum resolution of 512 x 512 pixels achieved with 1 KHz capture. Frame rate and image size are inversely related due to the memory limit of the camera. For a frame rate of 5 KHz, the maximum image size is 256 x 128 pixels and for 10 KHz is 128 x 128 pixels. The high speed camera and digital oscilloscope were synchronised and used the same acquisition rate of 5 KHz. Some trials were made at 10 KHz but due to the camera maximum image size, it did not cover all of the intended image width. Camera and oscilloscope were triggered by the same switch connected to the trigger pins of both devices and no lag was found on the data captured.

A 400 mm focal length lens was used together with a 2 X teleconverter lens. The whole equipment was mounted on a rail with micro-adjustable stands, facilitating the focus and zooming adjustments (Figure 3.11). High magnification and a long focusing distance enable the camera to be used away from the welding environment while capturing the whole arc image. A neutral density filter was positioned in front of the lens to reduce the amount of arc light. This worked in conjunction with the lens diaphragm adjustment.

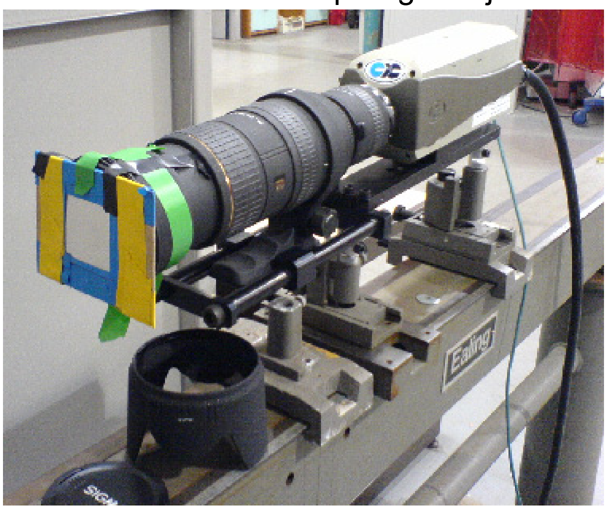


Figure 3.11 – High speed video camera, lenses and filter on the rail

3.3.2 Experimental procedure

The experiments were divided into three sets. The first set evaluated arc signal sensitivity with oscillation frequency on a 45° inclined plate, for CV GMAW and GMAW-P comparison. The second set evaluated arc signals and weld metal penetration on a single sidewall with a 5° preparation angle to simulate only one bevelled side of a pipe. Torch sidewall proximity, torch oscillation frequency and CTWD were the varying variables. The third set was identical to the previous one but with the inclusion of a second 5° opposite sidewall to form a groove. There was no CTWD variation in this trial set. Detailed descriptions of these experiments can be found in Appendix H.

3.3.2.1 Arc signal sensitivity comparison between GMAW and GMAW-P

For this set of experiments, a similar approach to Pan's [19] experimentation for GMAW with the high speed rotating torch was taken. The workpiece was tilted at 45º thus creating a ramp where the arc length is forced to vary between short to long to short in each oscillation excursion (Figure 3.12). By using different oscillation frequencies in the range found in previous studies, the same type of dynamic arc behaviour study can be made. This test was performed with CV GMAW and GMAW-P in order to compare arc signals behaviour in both cases.

Where

Δw oscillation width

L_{a1} long arc length

L_{a2} short arc length

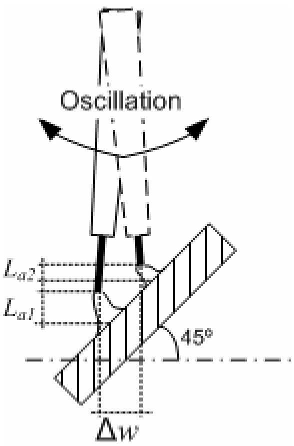


Figure 3.12 – Experiment setup with a 45º ramp for CV GMAW and GMAW-P

The base welding conditions for GMAW-P process were as follows:

Peak current	**350 A**
Peak time	**1.7 ms (including ramp up time)**
Background Current	**50 A**
Background time	**1.8 ms**
Frequency	**181 Hz**
Wire feed speed	**7.6 m/min**
Wire	**Carbofil NiMo1, 1mm diameter**
Shielding gas	**BOC Tri-mix (5%He, 12.5%CO_2, 82.5% Ar)**
Gas flow rate	**20 l/min**
Travel speed	**0.5 m/min**
Oscillation width	**2.5 mm @ 13 mm CTWD**
Workpiece base metal	**Carbon steel**
Workpiece thickness	**6 mm**

For CV GMAW, the welding conditions were:

Set voltage	**26.5 V**
Wire feed speed	**7.5 m/min**
Wire	**Carbofil NiMo1, 1mm diameter**
Shielding gas	**BOC Tri-mix (5%He, 12.5%CO_2, 82.5% Ar)**
Gas flow rate	**20 l/min**
Travel speed	**0.5 m/min**
Oscillation width	**3.7 mm @ 13 mm CTWD**
Workpiece base metal	**Carbon steel**
Workpiece thickness	**6 mm**

Table 3.4 describes the values of the different trials in terms of oscillation frequency as a function of short CTWD (short arc) and long CTWD (long arc). Torch weaving on a 45º slope creates a short arc when the maximum excursion of the torch oscillation has a smaller CTWD and a long arc in the opposite maximum excursion.

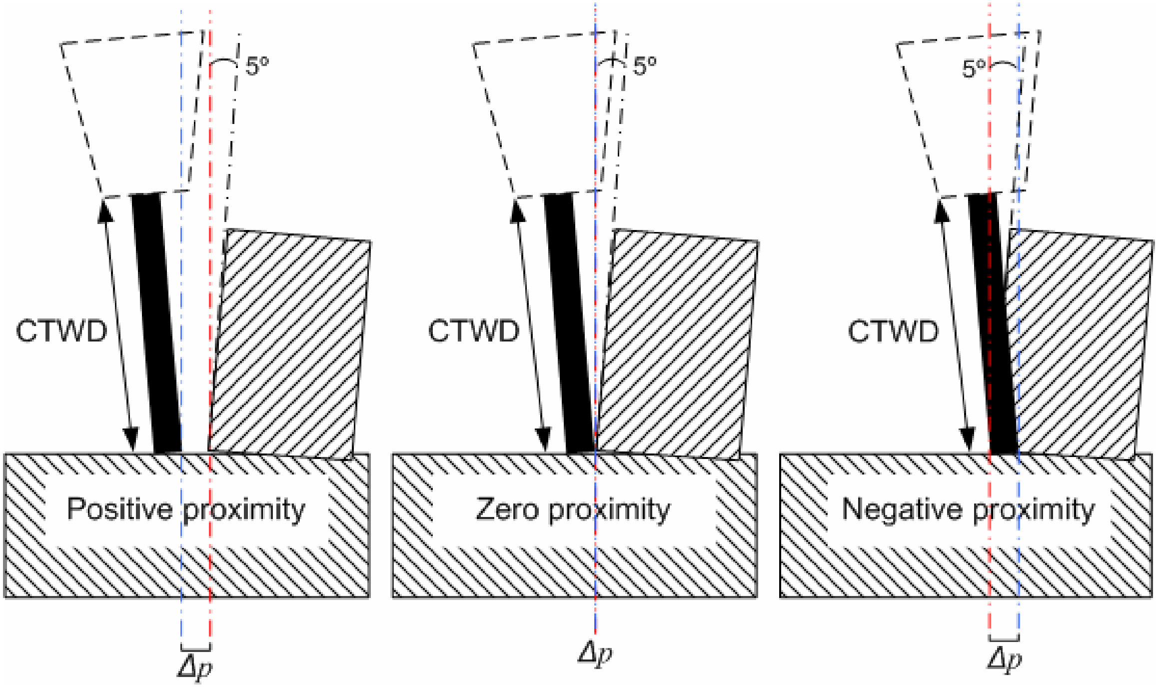
Table 3.4 – Arc sensitivity trials definition

Trial	Welding process	Oscillation Frequency (Hz)	short CTWD (mm)	Long CTWD (mm)
C1.1	GMAW-P	5	13.2	15.7
C1.2	GMAW-P	10	13.2	15.7
C1.3	GMAW-P	15	13.2	15.7
C1.4	GMAW-P	20	13.2	15.7
C1.5	GMAW-P	25	13.2	15.7
C2.1	GMAW-P	5	15.4	17.6
C2.2	GMAW-P	10	15.4	17.6
C2.3	GMAW-P	15	15.4	17.6
C2.4	GMAW-P	20	15.4	17.6
C2.5	GMAW-P	25	15.4	17.6
C3.1	GMAW-P	1	13.2	15.7
C3.2	GMAW-P	3	13.2	15.7
C3.3	GMAW-P	5	13.2	15.7
C3.4	GMAW-P	7	13.2	15.7
C3.5	GMAW-P	9	13.2	15.7
C4.1	CV GMAW	5	12.4	14.7
C4.2	CV GMAW	15	12.4	14.7
C4.3	CV GMAW	25	12.4	14.7
C5.1	CV GMAW	3	12.9	15.3
C5.2	CV GMAW	6	12.9	15.3
C5.3	CV GMAW	9	12.9	15.3
C5.4	CV GMAW	12	12.9	15.3
C5.5	CV GMAW	15	12.9	15.3

3.3.2.2 Single sidewall trials with 5° preparation angle

This set of experiments evaluated arc behaviour for arc signals sensitivity analysis and weld metal penetration on a single sidewall with 5° preparation angle, using GMAW-P (Figure 3.13). Torch proximity to the sidewall, torch oscillation frequency and CTWD were varied to study their influence on arc signals variation and on both bottom and lateral sidewall weld metal penetration. Potential welding defects were also analysed by metallographic analysis of weld profile. This test establishes the relationship between arc signals and torch sidewall proximity, important in achieving torch oscillation width control.

To complement this study, trials were also made to determine the influence of CTWD with a constant torch oscillation frequency and sidewall proximity.



Where

Δp wire edge distance to the sidewall corner

Figure 3.13 – Single sidewall proximity experimentation setup and wire proximity definition

The blue lines of Figure 3.13 represent the wire edge and the red lines the corner formed by the sidewall and the bottom workpiece meaning the zero position. The red lines are perpendicular with the bottom workpiece and form 5º with the sidewall inclination. Wire proximity from the sidewall (*Δp*) is the distance between the red line and the blue line as shown in Figure 3.13.

The base welding conditions were as follows:

Welding process	**GMAW-P**
Peak current	**350 A**
Peak time	**1.7 ms (including ramp up time)**
Background Current	**50 A**
Background time	**1.8 ms**
Frequency	**181 Hz**
Wire feed speed	**7.6 m/min**
Wire	**Carbofil NiMo1, 1mm diameter**
Shielding gas	**BOC Tri-mix (5%He, 12.5%CO_2, 82.5% Ar)**
Gas flow rate	**20 l/min**
Travel speed	**0.5 m/min**
Oscillation width	**2.5 mm @ 13 mm CTWD**
Bottom base metal	**Carbon steel**
Bottom thickness	**12 mm**
Sidewall base metal	**Carbon steel**
Sidewall thickness	**9 mm**
Sidewall height	**23 mm**

Table 3.5 - Experimentation phase 3 – single sidewall proximity trials definition

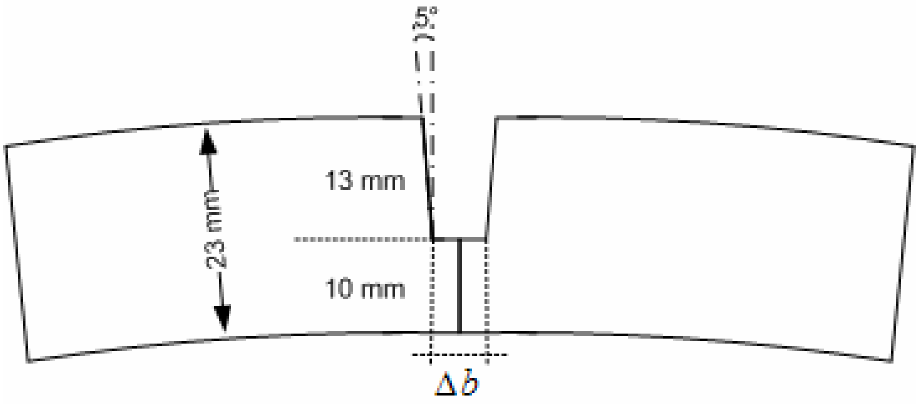
Trial	Oscillation Frequency (Hz)	Proximity (mm)	CTWD (mm)
D1.1	5	0	13.5
D1.2	10	0	13.5
D1.3	15	0	13.5
D1.4	20	0	13.5
D1.5	25	0	13.5
D2.1	5	1	13.5
D2.2	10	1	13.5
D2.3	15	1	13.5
D2.4	20	1	13.5
D2.5	25	1	13.5

D3.1	5	-1	13.5
D3.2	10	-1	13.5
D3.3	15	-1	13.5
D3.4	20	-1	13.5
D3.5	25	-1	13.5
D4.1	5	0	13
D4.2	5	0	14
D4.3	5	0	15
D4.4	5	0	16
D4.5	5	0	17

3.3.2.3 Double sidewall (groove) trials with 5° preparation angle

This set of experiments evaluated arc behaviour for arc signals sensitivity analysis and weld metal penetration on a double sidewall or groove with 5° preparation angle, using GMAW-P. Proximity was measured by the same process as shown on Figure 3.13 but for both sidewalls. Torch proximity to the sidewall and torch oscillation frequency were varied to study their influence on arc signal variation and on both bottom and lateral sidewall weld metal penetration. Potential welding defects were also analysed by metallographic analysis of weld profile. This test establishes the relationship between arc signals and torch sidewall proximity inside the groove for torch oscillation width control.

In single sidewall trials, the use of two parts (bottom workpiece and sidewall) as displayed in Figure 3.13 did not allow the detection of potential lack-of-sidewall fusion defects in the corner formed by the parts. Because of that, the groove for this set of trials was formed as shown in Figure 3.14. Similar measurements were made in this set of trials as for the single sidewall trials. The effect of groove width changes on arc signals was then assessed.



Where

Δb groove bottom width

Figure 3.14 – Double sidewall proximity experimentation

The base welding conditions were as follows:

Welding process	**GMAW-P**
Peak current	**350 A**
Peak time	**1.7 ms (including ramp up time)**
Background Current	**50 A**
Background time	**1.8 ms**
Frequency	**181 Hz**
Wire feed speed	**7.6 m/min**
Wire	**Carbofil NiMo1, 1mm diameter**
Shielding gas	**BOC Tri-mix (5%He, 12.5%CO_2, 82.5% Ar)**
Gas flow rate	**20 l/min**
Travel speed	**0.5 m/min**
Oscillation width	**2.5 mm @ 13 mm CTWD**
Workpiece base metal	**X100 HSLA Steel**

Table 3.6 - Experimentation phase 3 –groove proximity trials definition with an oscillation width of 2.5 mm @ 13.5 CTWD

Trial	Oscillation Frequency (Hz)	Proximity (mm)	Groove bottom width (mm)	Comments
E1.1	5	-0.5	2.5	Off-centre 0.3 mm
E1.2	10	-0.5	2.5	Off-centre 0.3 mm
E1.3	15	-0.5	2.5	Off-centre 0.3 mm
E1.4	20	-0.5	2.5	Off-centre 0.3 mm
E1.5	25	-0.5	2.5	Off-centre 0.3 mm
E2.1	5	0	3.5	Off-centre 0.6 mm
E2.2	10	0	3.5	Off-centre 0.6 mm
E2.3	15	0	3.5	Off-centre 0.6 mm
E2.4	20	0	3.5	Off-centre 0.6 mm
E2.5	25	0	3.5	Off-centre 0.6 mm
E3.1	5	+0.5	4.5	
E3.2	10	+0.5	4.5	
E3.3	15	+0.5	4.5	
E3.4	20	+0.5	4.5	
E3.5	25	+0.5	4.5	
E4.1	5	-0.5	2.5	
E4.2	10	-0.5	2.5	
E4.3	15	-0.5	2.5	
E4.4	20	-0.5	2.5	
E4.5	25	-0.5	2.5	
E5.1	5	0	3.5	
E5.2	10	0	3.5	
E5.3	15	0	3.5	
E5.4	20	0	3.5	
E5.5	25	0	3.5	

Trials E6 to E8 were made with an oscillation width of 3.7 mm for a CTWD of 13.5 mm. The other parameters were kept the same.

Table 3.7 - Experimentation phase 3 –groove proximity trials definition with an oscillation width of 3.7 mm @ 13.5 CTWD

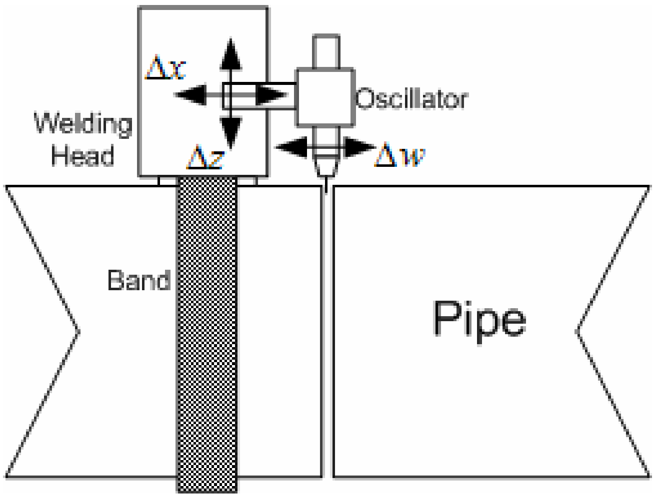
Trial	Oscillation Frequency (Hz)	Proximity (mm)	Groove bottom width (mm)	Comments
E6.1	5	+0.15	5	
E6.2	10	+0.15	5	
E6.3	15	+0.15	5	
E6.4	20	+0.15	5	
E6.5	25	+0.15	5	
E7.1	5	+0.65	6	
E7.2	10	+0.65	6	
E7.3	15	+0.65	6	
E7.4	20	+0.65	6	
E7.5	25	+0.65	6	
E8.1	5	+0.55	5.8	Off-centre 0.5 mm
E8.2	10	+0.55	5.8	Off-centre 0.5 mm
E8.3	15	+0.55	5.8	Off-centre 0.5 mm
E8.4	20	+0.55	5.8	Off-centre 0.5 mm
E8.5	25	+0.55	5.8	Off-centre 0.5 mm

4 Software development for analysis and control

Throughout this project, computer programs had to be developed that could fulfil the different requirements of the project. Some of these programs were developed to help with the analysis of the data generated by experimentation and some to achieve the proposed control. This chapter is divided in two parts: the first part is dedicated to the monitoring and control software and respective algorithms and the second part to the analysis software.

4.1 Monitor and control software algorithms

The development of algorithms for control described here was based on the analysis of the first phase of experimentation. The aim was to achieve control of two axes (X and Y) of the welding head as shown in Figure 4.1. The axis W control (oscillation width) is discussed on chapter 6 and was not implemented in this phase.



where

Δx	torch cross seam position
Δz	torch height position
Δw	oscillation width

Figure 4.1 – Schematic representation of axes on a conventional pipe welding head

The welding head used for the control algorithms development was the same used for experiments A1 to A6 – RMS MOW II. This welding head is not autonomous and needs the control of an external device, using a pendant. The communication between these two devices is by Controller Area Network (CAN) from Bosch ®. CAN was created by Bosch ® for the automotive industry. This network is known for its simplicity of implementation and reliability being tested and used in the automotive industry around the world.

An implementation of CAN was used by the commercial company (RMS) that supplied the pipe welding head and controller used in this work. A CAN hub was created where all devices communicate via CAN, including the welding head, the pendant operated by the welder, the power supply controller and the arc signal mean values measurement device. The latter was named the VISENSE device by RMS (V for voltage, I for current, SENSE for through-the-arc sensing) and this name is used throughout this work. A picture of the VISENSE device is shown in Figure 4.2. It is based on a Digital Signal Processor (DSP) from Motorola ® (DSP56F803) with the complete circuitry for arc signal acquisition, processing and CAN messaging, ideal for control algorithms implementation.

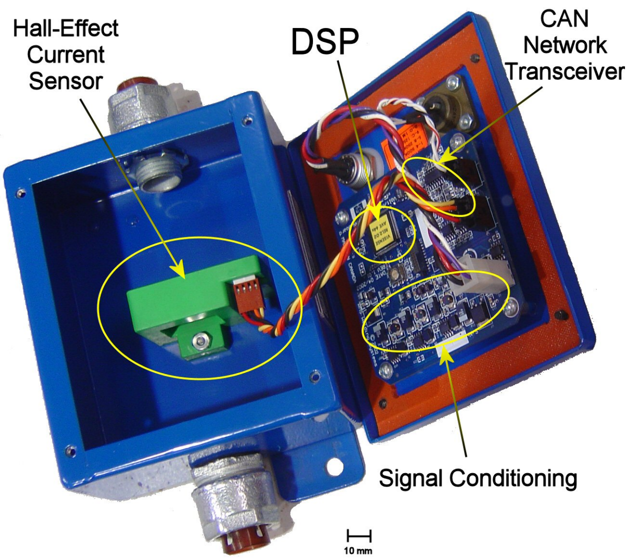


Figure 4.2 - RMS ® VISENSE device for arc signals acquisition, processing and CAN messaging

All the CAN enabled devices in the RMS network implementation receive and send CAN messages to the network. The majority of these messages start from the pendant as commands and the network devices reply with feedback messages of their specific tasks. A standard self broadcast message is each device's ID. Any message in this type of network is listened to by all and multiple devices can utilise it. Pendant messages to the welding head are moving commands for the head to perform in any of its axes. The same type of message can be sent by any device in the network. This feature was exploited in order to send torch position corrective messages simulating pendant messages, using the VISENSE device. For this, the VISENSE device was completely reprogrammed to accomplish the new tasks of torch guidance and control by through-the-arc sensing, as detailed in section 4.1.1. RMS supplied all the information necessary about their CAN messaging implementation and the VISENSE device programmable details.

To understand and control the whole process, two software programs for two platforms were developed. The main algorithms were integrated in the software developed for the VISENSE device and a second Microsoft Windows ® for PC software runs on a laptop and monitors the message flow on the CAN for debugging purposes. The VISENSE software was developed with Metrowerks CodeWarrior Integrated Development Environment (IDE) version 4.2.6.922 from Freescale Semiconductor Company ® [109] for Motorola ® DSP568xx family and the PC software was developed with Microsoft ® Visual Basic 6.0 IDE [110] for Microsoft Windows PC platforms.

The program for the VISENSE device was uploaded through a special developed onboard JTAG connection to the parallel port of the PC where debugging was also possible. This program was developed mainly in assembly language to increase processing speed with some parts of the code in C language [111, 112]. The DSP56803 has a 16 bit engine with dual Harvard architecture and processes 40 million instructions per second (MIPS) at 80 MHz core frequency. On the PC side a laptop computer (Sony Vaio) was used with an Intel Pentium III at 1GHz with 384MB of RAM and 30GB of hard disk storage space. To communicate with the CAN, a Vector CANCardX PCMCIA board from Vector Informatik GmbH ® was used [113] and a special connection to the RMS CAN had to be developed. The RMS system operates the CAN at 512 kbps speed [114].

4.1.1 DSP program functional description

The software developed for the DSP was started from scratch. An event driven processing environment was created by the use of different hardware generated interrupts like the CAN messaging, the Analog to Digital Converter (ADC) and from different Timers. Since no operating system is present, everything had to be developed to be controlled, including hardware drivers, events scheduling, messaging system, etc. The use of interrupts in this case is very important to establish priorities and to manage the events in an orderly fashion. All system errors, stalls and crashes result only from the developed software. A very powerful and controllable framework essential for automation was created in this way.

In order to achieve a consistent weld quality, it is important to sustain a good arc length control [115-117]. In GMAW-P, the control of arc length (arc voltage) can be achieved by changing arc current pulse parameters or by varying CTWD accordingly. Pulse parameter change may create procedure problems and inconsistent results. Varying or adjusting CTWD values until the arc voltage is restored is the right approach then, due to the steady current characteristic provided by the experimented power supply. In most cases, it was a CTWD variation that created initially the arc length variation. This is the type of correction usually made by the welder when guiding the torch. The strategy followed in this work was to replicate the welder’s behaviour by moving the torch up and down to correct CTWD, in order to attain the correct voltage value.

From the results and discussion of experimentation phase 1, it was concluded that peak voltages are more consistent than background or average voltages and therefore more appropriate for control. In the case of CTWD control, peak voltages in the groove centre should be measured and their value must be maintained as constant as possible in order to provide a good weld quality. This implies previous knowledge of an optimum peak voltage to be used and that is only possible by methodical experimentation. In other words, new welding procedures have to be created to include the new "*optimum peak voltage*" value. Since it is not available in the present and the costs of redoing the existing welding procedures are very high, it was decided that CTWD control should follow the average voltage instead. This value is already defined in actual welding procedures. This decision may compromise algorithm control efficiency and reduced control accuracy but it does not invalidate the results as it is shown in results (section 5.2.1).

To use the welding procedure average voltage, some user input must exist to communicate with the VISENSE device. One existing RMS pendant feature is the average voltage value definition for CV GMAW power supplies. When this value is set, a CAN message is sent to the power supply controller to adjust the power supply voltage level. This CAN message is also captured by the VISENSE device. For a GMAW-P welding procedure, this message is not used by the power supply controller and can be used by the control algorithms inside the VISENSE device as the welding procedure average voltage reference.

The welding process starts when the welder presses the Arc Start button on the pendant sending a CAN message of ARC START. The VISENSE device captures the message and starts the internal process of activating the timers that will pace the whole process. The arc voltage is acquired at 10 KHz and averaged as a moving average algorithm. Every 0.1 s the averaged voltage is compared to the referenced voltage. If the averaged voltage is higher than the reference voltage, the CTWD value has to decrease and vice-versa. A corrective CAN message is sent for UP/DOWN torch correction with a proportional value resultant from the calculated difference divided by a tuning coefficient (PID controller).

As an example of this control system, the welder defines 20 V as the intended arc voltage to be sustained and adjusts a certain CTWD. When the arc goes on, the system calculates the average arc voltage and if that value is only 17 V, the device starts sending the torch UP commands every 0.1 ms of a calculated amount from the proportional coefficient value. This coefficient should be tuned according to the motion system in use. The torch goes up until the arc average voltage and arc reference voltage are even. The same occurs in the opposite direction. Figure 4.3 shows the program block diagram.

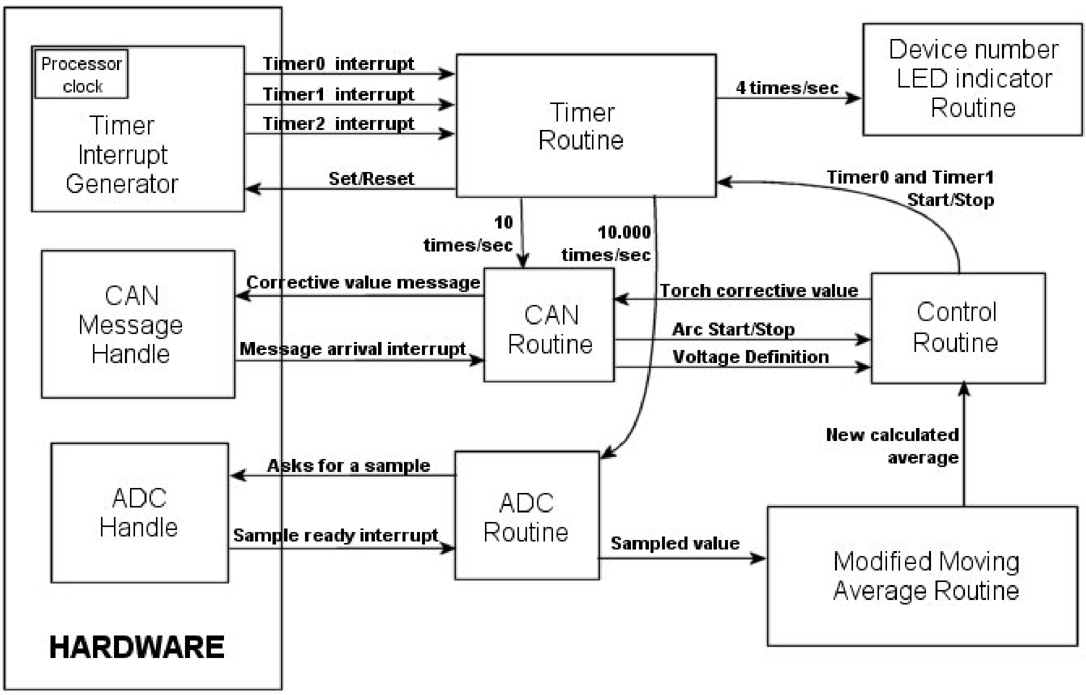


Figure 4.3 - Block diagram of the initial VISENSE CTWD program version.

The initial software package was sent to RMS Canada for field tests and used successfully by them to control torch height. Seam tracking was then the second phase to be developed. For this next step, changes in the RMS welding head had to be implemented. It was important to know the torch position while oscillating. These changes were made by RMS and new firmware was uploaded into the welding head. The new welding head firmware sends a CAN message every time the torch reaches one of the three positions:

- Maximum Left Excursion
- Centre of Excursion
- Maximum Right Excursion

This three positional messages technique is independent of oscillation width and can be used in any welding process and procedure. The message is intercepted by the VISENSE device and is used to track the torch position enabling the second phase of the development. This phase created some structural changes to the initial system and the final result can be seen in Figure 4.4. In summary, the control routine that calculates the correction to be made in the CTWD now also calculates the tracking error.

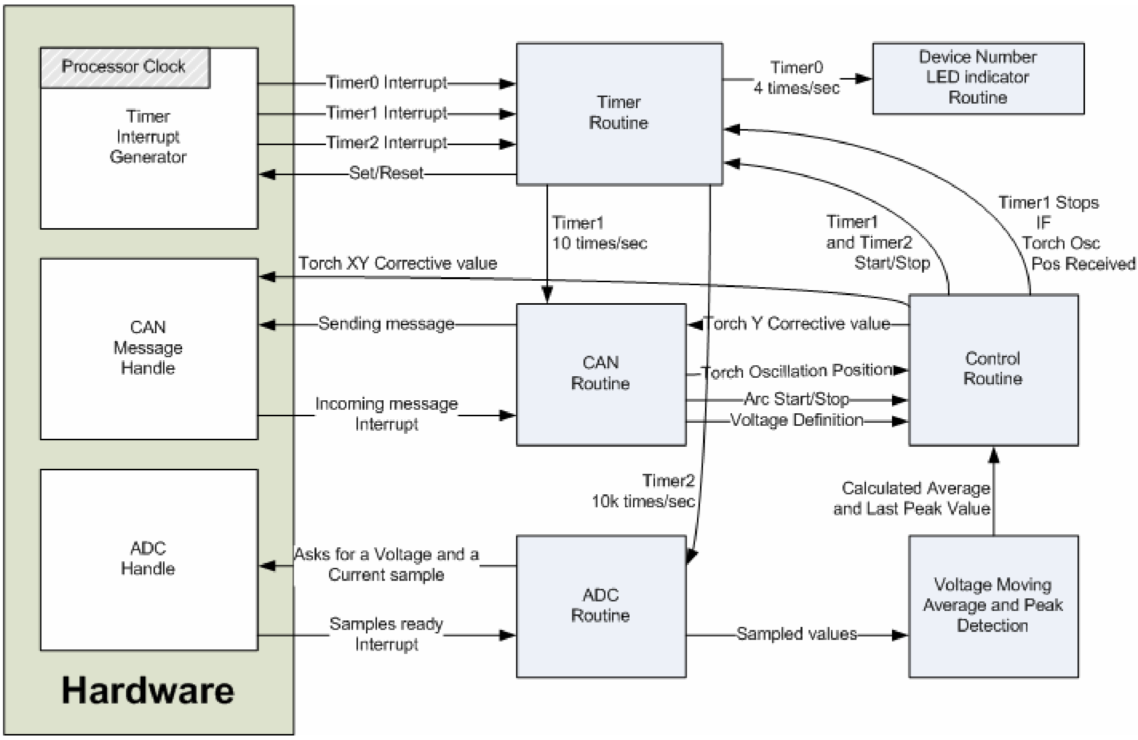


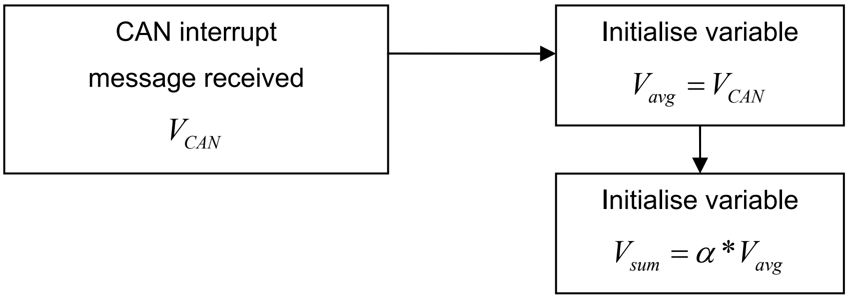
Figure 4.4 – Block diagram of the final VISENSE seam-tracking program version

When the process starts with the ARC START CAN message, the system behaves as a CTWD adjustment only. To activate the cross seam position control algorithm, at least one positional CAN message should be received. Then the program changes internal routines. It stops the timer that sends CTWD corrective CAN messages every 0.1 s and follows the new pace defined by the torch positional messages. This new virtual timer establishes CTWD corrections when the VISENSE device receives “Torch Centre” messages, and seam-tracking corrections when it receives “Torch at Left“ and “Torch at Right” messages.

When the VISENSE device is powered up, the program starts with the ‘Main’ routine that only contains a call to the ‘Init’ routine and a main loop to keep the program running indefinitely. The ‘Init’ routine initialises all the global variables and hardware, namely:

- reading and setting the device number by reading the onboard rotary switch
- setting up the timers
- setting up the CAN subsystem and message filters
- setting up the ADC controller
- enabling the Timer0 for the device number LED indicator

The process then enters into an idle stage waiting for CAN messages. When a CAN message filter is triggered for a valid incoming message, this one is processed and the respective routine is called. The first message waiting to be processed is the arc voltage reference message (Figure 4.5). If this message is not sent, the guidance processes will never start. Having received this message, the system waits then for the ARC START message to start processing. On arrival, this message enables Timer1 and Timer2 that defines the pace of the whole process until an ARC STOP message is received. Although there was no positional message information so far, the system calculates the voltage average (Figure 4.6) and detects the voltage peaks (Figure 4.9).



where

V_{avg} arc average voltage

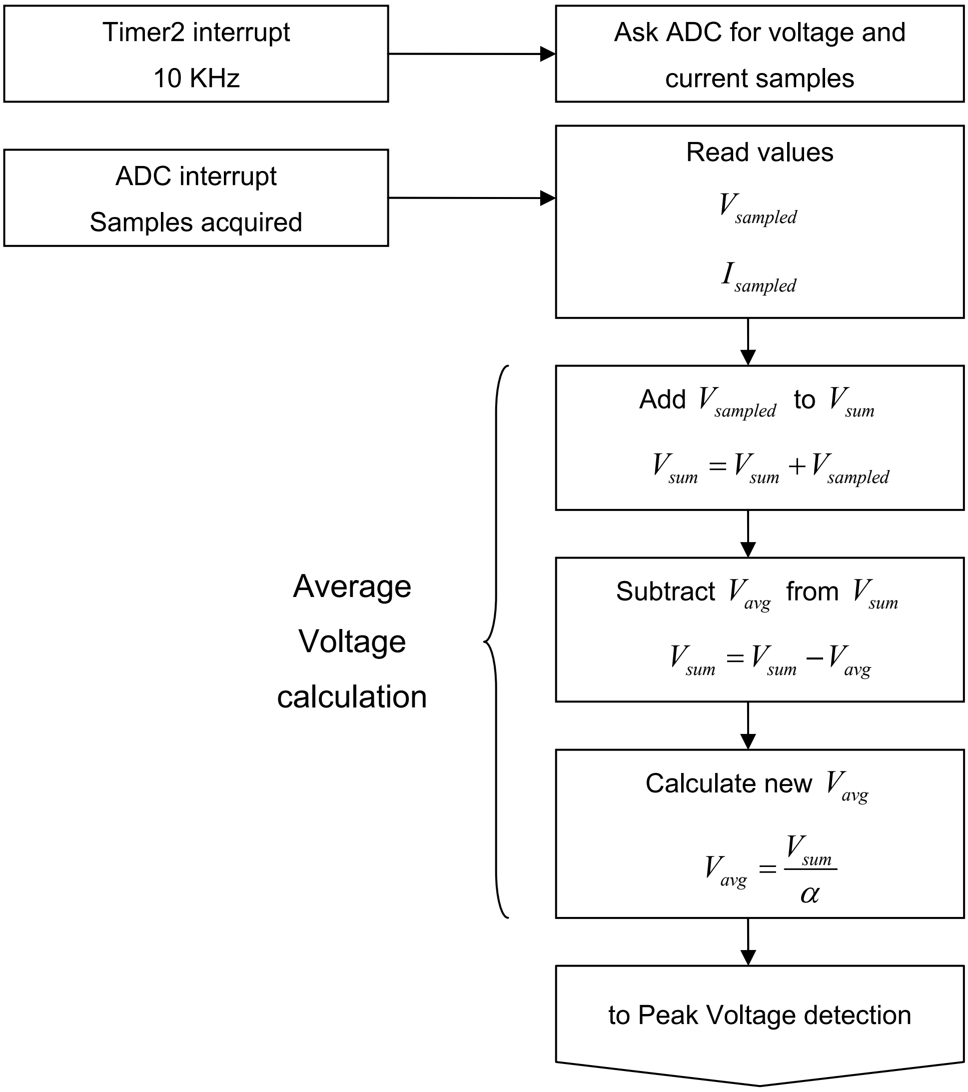
V_{CAN} arc voltage reference

V_{sum} arc voltage accumulation variable

α averaging window or smoothing factor - the number of samples to be used on average calculation

Figure 4.5 – Flow chart of variable initialisation on arc reference voltage CAN message arrival

The averaging window is adjusted by testing different system reactions to different values. A higher value reduces the sensitivity to short term variations slowing the corrections. A lower value makes the system more sensitive but also less stable. A value of 2^{10} samples (1024) was used with good results. For a 10 KHz sampling rate, this averaging window is around 0.1 s. To increase the speed of calculations, normal arithmetic operations should be avoided because it takes too many processor cycles to execute. Instead, simple binary shifting operations can easily implement a division by 2 if it is a binary right shift or a multiplication by 2 is it is a binary left shift. In this case, the averaging window is a base 2 value to simplify the calculations.



where

V_{avg}	arc average voltage
V_{sum}	arc voltage accumulation variable
$V_{sampled}$	sampled arc voltage
α	averaging window or the number of samples to be used on average calculation

Figure 4.6 – Flow chart of arc average voltage calculation when the arc is on and at a rate of 10 KHz

From the results and discussion of the first phase of experimentation, peak voltages (voltage at pulse peak current) were found to be more consistent and reliable than background or averaged voltages, in GMAW-P. This was the fundamental principle which the control algorithm development was based upon. Peak voltages evolution follows a trend related to the torch position across the seam. Two voltage levels can be asserted in a complete oscillation cycle: a minimum peak voltage when the torch is in its maximum excursion and a maximum peak voltage when the torch is in the centre of oscillation. This fact is discussed later in chapter 6.

For cross-seam control development, this fact can be used to detect off-centre torch position. By comparing both peak voltages from the maximum torch excursions in opposite directions, it is possible to ascertain if the torch is not in the groove centre. This voltage difference also shows the direction and the respective amount of torch misalignment.

To process this control, two algorithms were developed. A first algorithm detects peak voltages from the acquired arc voltage signal and the second algorithm processes torch extreme positions CAN messages by comparing the measured peak voltages in both positions. The latter also processes the result of the comparison and sends a positional correction CAN message to the welding head.

The first algorithm processes the continuous stream of data from the ADC by detecting the highest value of the stream only for values over a predefined threshold value. Being a pulsed signal, the threshold should be defined with a value around the average of the signal being acquired. Figure 4.7 shows an example of an arc current pulse with different thresholds. From the figure, and considering the threshold equal to the pulse average, when the stream of data values is higher the threshold (between t_1 and t_2), a valid pulse was found and the maximum value detected above the threshold is the pulse peak (i_{max}). This is valid for both arc current and voltage pulses. When the stream of data values goes lower the threshold (after t_2), a new pulse is in progress and the process is reset to a new peak finding.

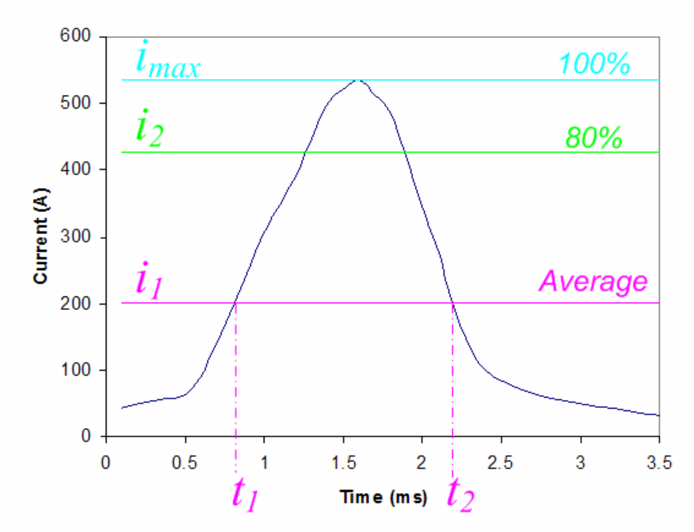


Figure 4.7 – Example of different threshold values from a real arc current pulse

The aim of the peak finding algorithm is to detect peak voltages, not necessarily through the voltage data stream. Some spurious peak voltages above threshold values may occur, as shown in Figure 4.8.

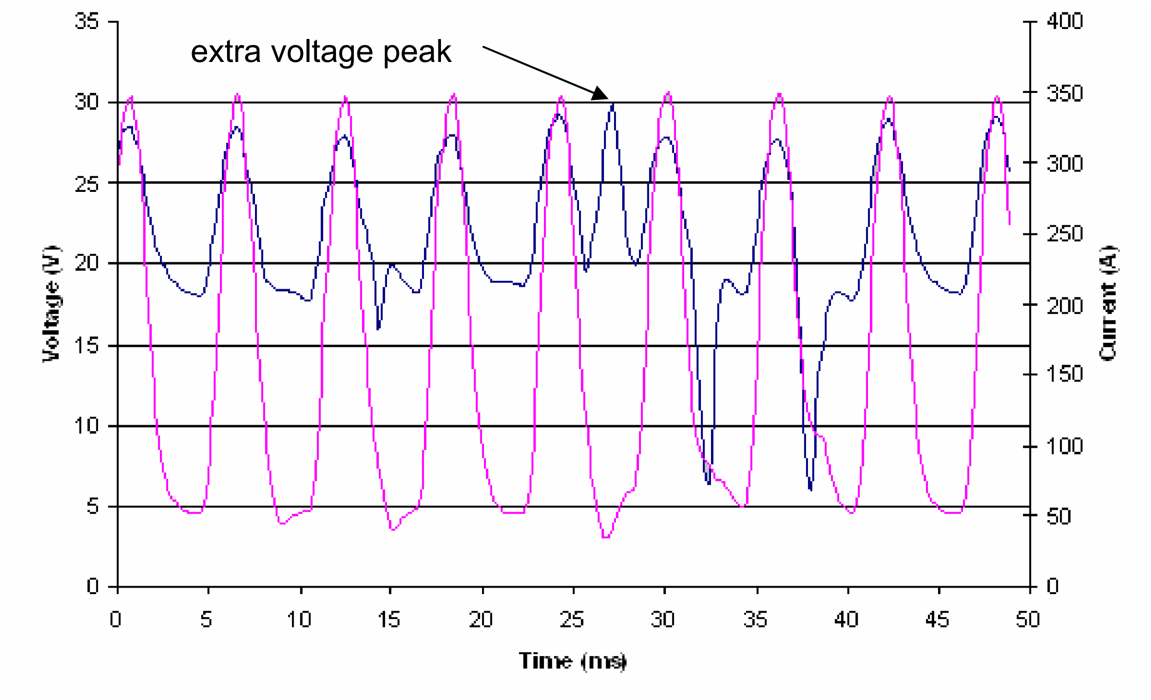
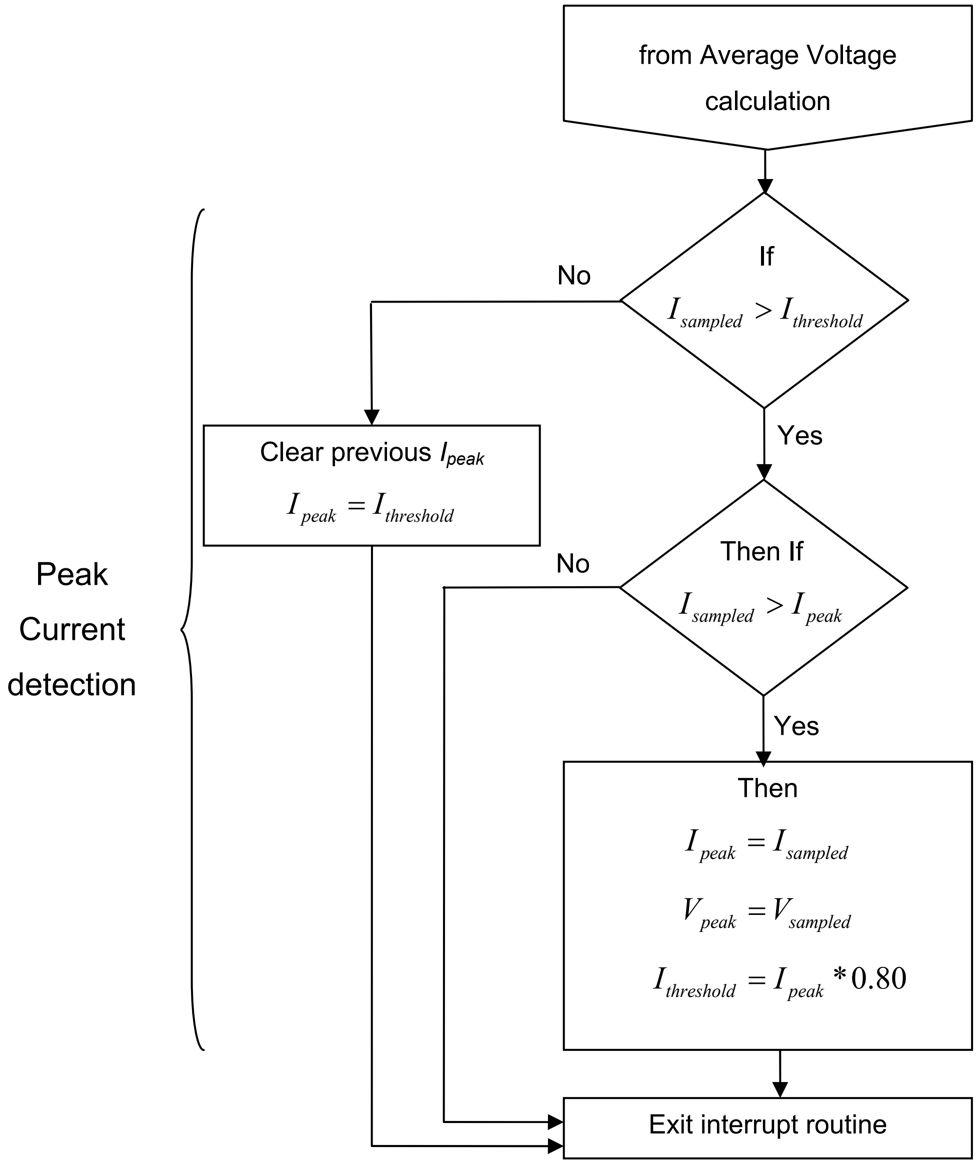


Figure 4.8 – Initial 50 ms of trial A1 – voltage in blue and current in red

To avoid this occurrence, peak currents were used instead to trigger the detection of peak voltages showing more consistence than peak voltages. When a peak current is detected, the voltage value in the same position of the data stream indicates the correspondent peak voltage. A threshold value of 80% peak current was used in this work with good results. Figure 4.9 shows the peak finding algorithm flowchart.



where

$I_{sampled}$	sampled arc current
I_{peak}	detected arc peak current
$I_{threshold}$	calculated arc current threshold
$V_{sampled}$	sampled arc voltage
V_{peak}	detected arc peak voltage

Figure 4.9 – Peak finding algorithm flowchart

4.1.2 PC program functional description

The VISENSE device is not provided with a graphical display and debugging is only possible through the code development interface. No process monitoring tool was available and so a Microsoft Visual Basic PC program was developed (Figure 4.10). This software uses the CAN subsystem to send and receive CAN messages, emulates the operator pendant and monitors corrective messages sent by the VISENSE device. The source code can be found in Appendix III.

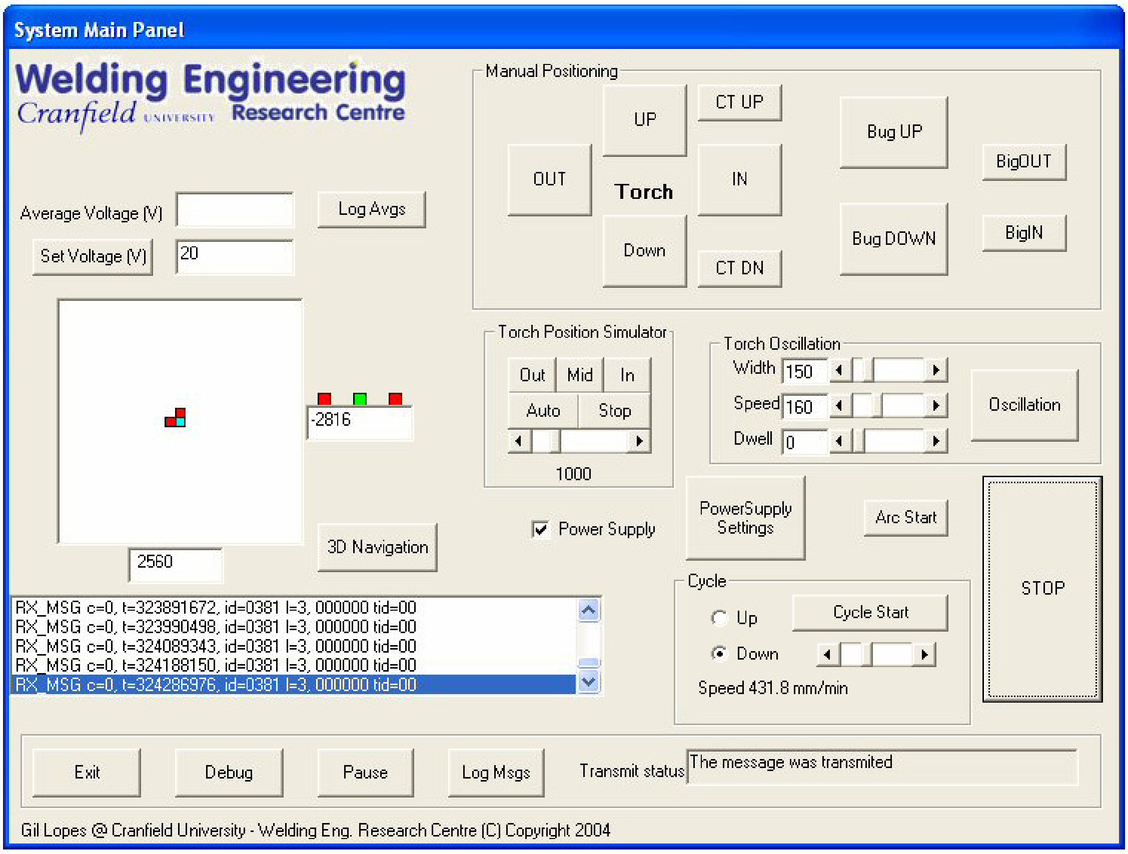


Figure 4.10 – Microsoft Visual Basic PC program main screen

To best describe program functionality, the main screen was divided in blocks and described separately. The first block is the welding head operability called “Manual Positioning” in the main screen. When the program is used to simulate the operator’s pendant, all welding head motion functions are available (Figure 4.11 - a).

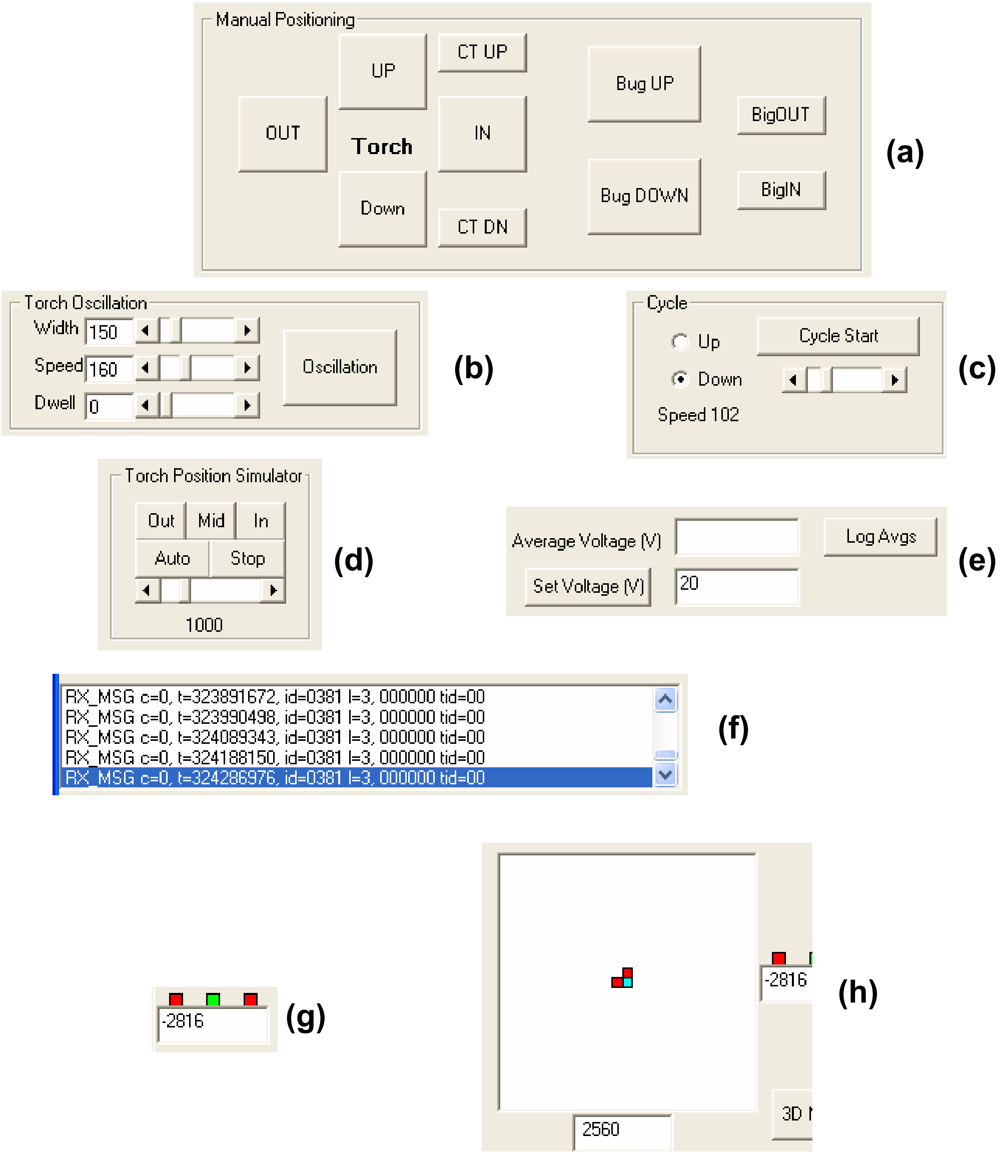


Figure 4.11 – Monitoring software main screen blocks

The user controls the entire torch and welding head positioning (Figure 4.11 - a), as well as welding procedure parameter definitions including oscillation speed and width (Figure 4.11 - b), welding head travel speed and direction (Figure 4.11 - c). No difference in performance was found between the real pendant and the developed one. The welding head responded promptly with both pendants and both can operate simultaneously. The software was also enabled to send sequentially torch positional messages, simulating the welding head (Figure 4.11 - d). The three positional messages can be sent isolated by pressing the correspondent screen button or automatically. In this case the user can change the speed between messages to simulate different torch oscillation frequencies. This feature is very useful to test and debug the VISENSE device without the welding head.

Also visible on the main screen and working as an output:

- "Set Voltage (V)" button to define the reference voltage
- "Cycle Start" button that starts the motion system (welding head travel and torch weaving)
- "Arc Start" button initiates the arc process through a sub-program that communicates with the power supply via Ethernet using the Internet Protocol.
- "Stop" button stops all processes and sub-process started by the user.

As an input, the main screen shows a list of incoming CAN messages (Figure 4.11 - f) and translates some of them to a logical visualisation such as:

- the average voltage computed by the VISENSE and sent via CAN (Figure 4.11 - e)
- the three positional messages sent by the welding head represented by three red squares (Figure 4.11 - g) that blink to green when the position has been reached
- the CTWD correction messages sent by the VISENSE device and displayed as a red vertical bar which increases in length in the direction of the torch movement by the amount of correction required (Figure 4.11 - h)
- the seam-tracking correction messages sent by the VISENSE device and displayed as a red horizontal bar that increases in length in the direction of the path correction required (Figure 4.11 - h)

Power supply settings can be adjusted by pressing the appropriate button to open the "Power Supply Settings" screen. The sub-program that controls the power supplies uses a Dynamic Link Library (DLL) developed by Tom Doyle (J.

Ray McDermott – USA) in agreement with Lincoln Electric® to operate with this type of networked power supply. The DLL simplifies the communication enabling the sending of basic commands to operate the power supplies.

To better visualise the motion correction during the welding process, 2D and 3D graphs are updated during welding (Figure 4.12). The 2D graph shows the deviations of path and CTWD by red and green lines, respectively. The 3D graph simulates the pipes being welded. The green fixed line represents the virtual path of the torch if no corrections were applied. The red line is drawn during welding with the vertical and horizontal torch path corrections. The graphs are National Instruments [118] objects and in the case of the 3D graph, the user is able to perform online changes on the perspective and zooming for better visualisation of the corrections being made or already made.

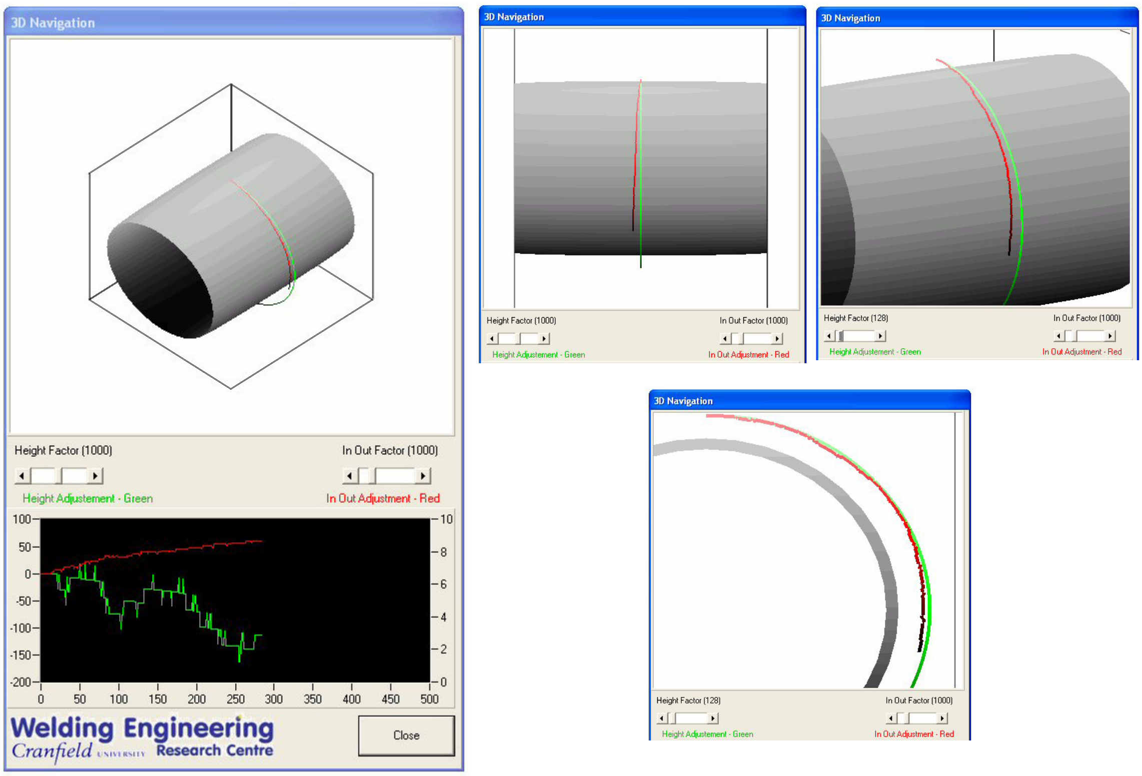


Figure 4.12 – 2D and 3D graphs of the torch corrections during the welding process

As a message logging solution, two types of message files can be created. One type records every single incoming and outgoing message and the other type records just the average voltage received messages. An external program was developed to help in analysing these files and to identify possible errors or anomalies.

4.2 Software for analysis

Arc signals are complex due to their frequent variation in time. The use of digitised information enables rapid processing of this complex information to extract significant features by use of modern signal processing techniques.

The most basic approach for signal analysis is the use of spreadsheets. Macro enabled spreadsheets such as Microsoft Excel are able to produce rapid results with low computational effort. Experiments A1-A6 were first analysed using this process and the code is shown in Appendix A. The algorithms used are explained together with the results in chapter 5.

Unfortunately, this analysis is of limited capacity and for instance is not able to cope with simultaneous visualisation of high speed images synchronised with arc signals. Correlations between image frames and sampled signals can only be achieved by software enabled to work with both types of data. Some researchers have developed their own software to perform this type of analysis [119] but they are mostly proprietary systems, not easily adaptable to changing requirements. To overcome this situation, a computer program was developed for the specific requirements of this analysis with capabilities for future expansion.

4.2.1 WeldData – analysis software main screen

A key part of this project was the detailed analysis of complex voltage and current waveforms, torch position data and arc images from the high speed camera. The objective of this software was to help in the extraction of key features of voltage and current waveforms in specific torch oscillation positions. It was also to provide synchronised visualisation of arc images from high speed video, at specific points in the waveform traces. Figure 4.13 shows the main screen of WeldData program.

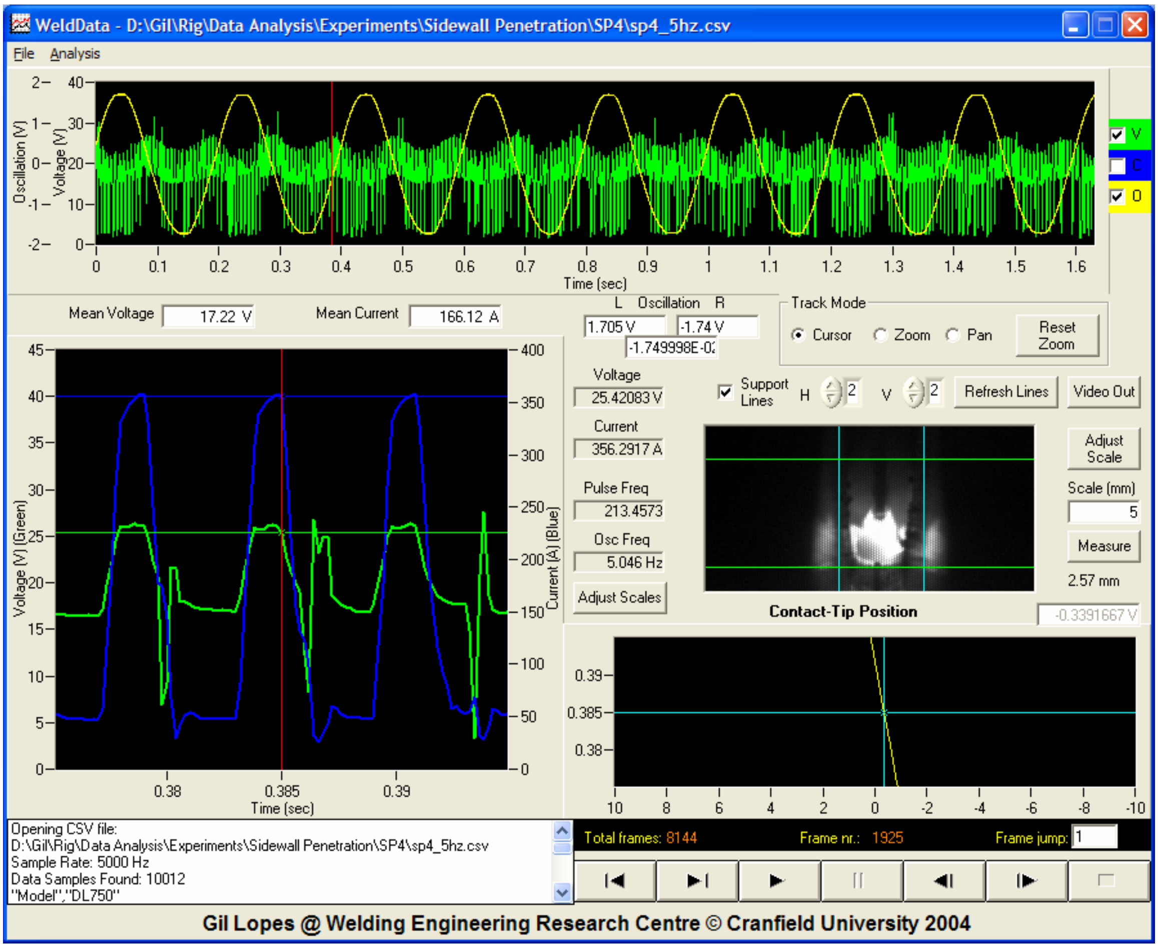
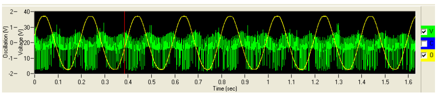


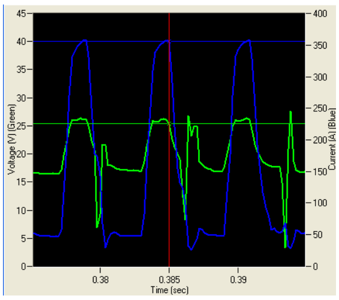
Figure 4.13 – WeldData main screen

The software code is detailed in Appendix G. The main screen gathers all the important aspects and features of interest to be shown like signal waveform viewing (the whole and the detail), basic file information and basic signal analysis (mean values).

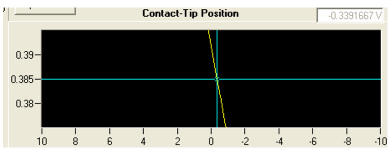
This software was developed following the Common User Access (CUA) guidelines [120]. There is a top menu bar with a file menu where the user selects the file to be opened and an analysis menu where the user can select different analyses to be performed. The main screen is arranged with:



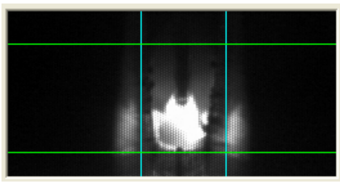
a) Top graph – Shows the complete sampled waveforms. The user selects which waveforms are to be seen from the select boxes on the right of the graph. Zooming and pan features are also available. The cursor (red vertical line) defines the position to be measured and can be dragged to a different position.



b) Bottom left graph – Shows in detail a small window of the voltage and current values extracted from the main signal. The time size (X-axis) of this graph can be adjusted to increase or decrease signals detail. The centre of this graph in the X-axis is the cursor position of the top graph.

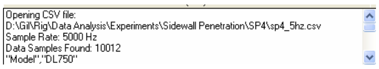


c) Bottom right graph – Shows the torch oscillation signal with time in the Y-axis. In this way the waveform follows the natural visible movement of the contact tip as seen from the high-speed camera point of view.



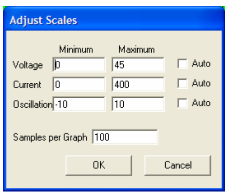
d) Video box (middle right side of the screen) – Shows the loaded video correspondent to the loaded arc signals. When a signal file is open, the program searches in the same folder for a video file with the same name as the signals file. The time of video defines the time limit of arc signals to be shown in the top graph.

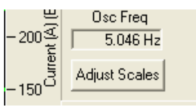
e) Control buttons – “Plays” the video synchronised with the signals. It also enables the play of frames one-by-one forwards and backwards. Random frame positioning and/or beginning/end positioning is also available.



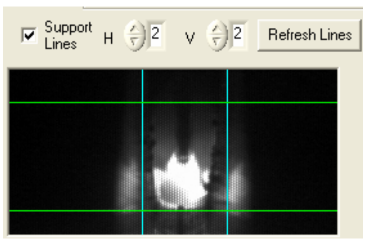
f) File information (bottom left) – It shows the header information from the exported oscilloscope arc signals data file.

g) Calculated values – Beneath the top graph are the mean voltage and mean current values of the whole loaded signal. Left, right and left-right difference of torch oscillation values are next. In the vertical centre of the main screen are located the measured values of voltage and current at the cursor position position. Beneath is the calculated arc pulse frequency (valid for GMAW-P) and torch oscillation frequency.



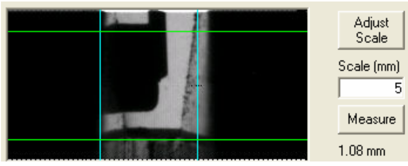


h) Bottom left and right graph scales – The scales of these two graphs can be changed to fixed values or auto values.

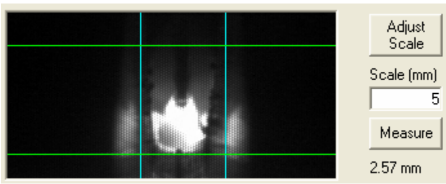


i) Vertical and horizontal support lines – The user can add up to 9 vertical (blue in the left image) and 9 horizontal (green in the left image) lines. These lines are support lines to help on image feature detection and size measurement. They can be dragged to the desired position.

j) Video out – Plays the video with the same abilities of step-by-step, reverse, etc. but in an external resizable window.



k) Adjust scale button – Serves to calibrate the image measurement system. When this button is pressed, in the existence of a jpeg file with the same name as the signals data file and in the same folder, the video file box shows the jpeg image. The user then has to pick two points from the image where a previous length is known. This calibration length should be written in the box beneath the button.



l) Measure button – After defining previously the calibration scale, the user can select from the video two points and the length is returned visually beneath the button.

Mean values were calculated by averaging the whole sampled signal. Pulse frequency was calculated by detecting and averaging the time difference between detected pulse peaks in the current. For torch oscillation frequency, a similar algorithm was used. The peak detection algorithm was based on the peak detection algorithm developed for the seam-tracking system presented in the section 4.1.1.

4.2.2 WeldData – average versus peak voltages analysis screen

For the purposes of this project, this analysis demonstrates the comparison between voltage average and voltage peaks extracted from voltage pulses.

Figure 4.14 shows the analysis screen. On the top, the graph has the extracted voltage peaks from each pulse in blue and the averaged voltage in green. The average was calculated with a conventional moving average algorithm [121-123]. The smoothing factor (average window) is initially calculated based on the number of samples for a complete arc current pulse. The smoothness of the averaged voltage signal quality tends to degenerate if the number of samples chosen is not the same length as the pulse period or its multiples. The higher the number, the smoother the averaged signal will be. The value used in the program is the inverse of the pulse frequency calculated initially when the data file is loaded divided by the sample rate of the sampled signal.

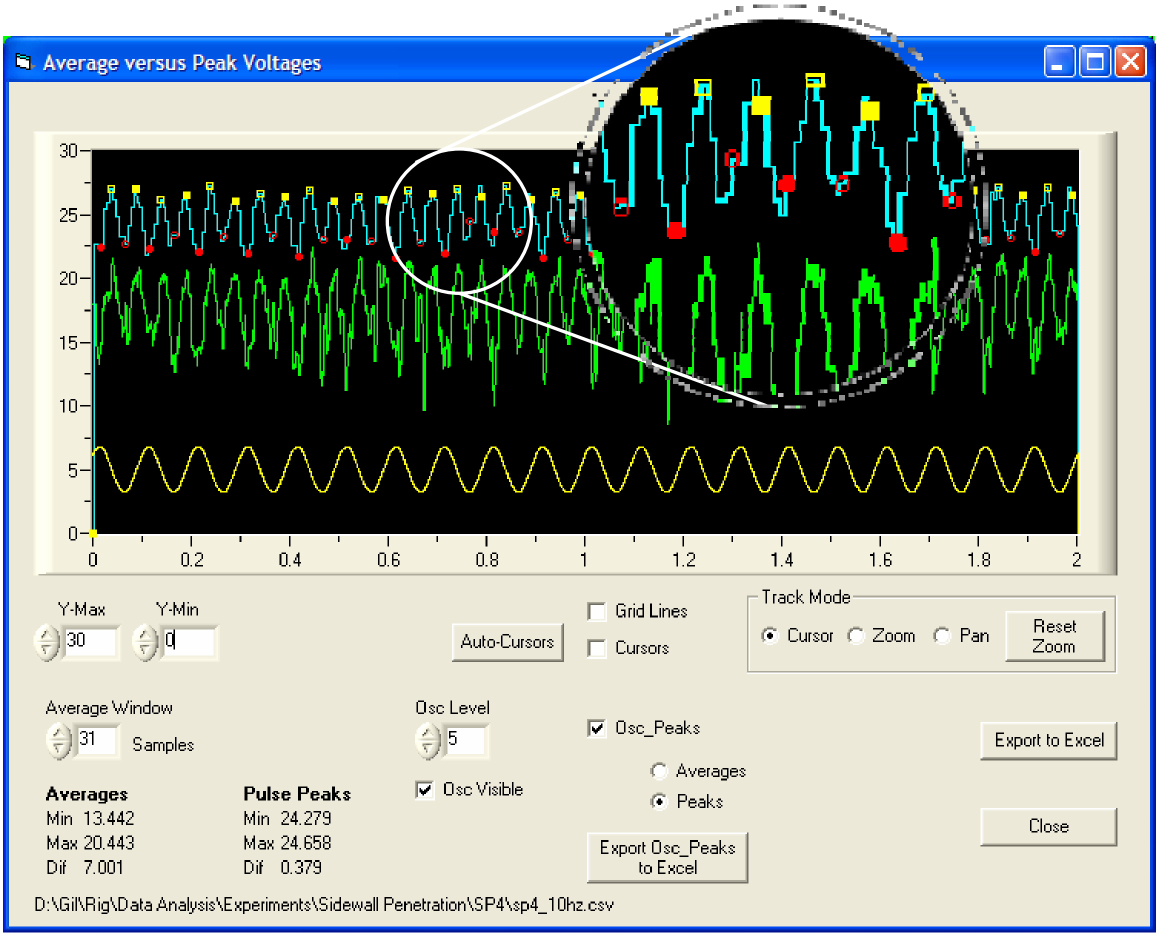


Figure 4.14 – Average versus peak voltages analysis screen

The graph from Figure 4.14 also shows the torch oscillation signal in yellow. This plot is visible by default but can be turned off. It helps in the analysis of arc voltage behaviour in relation with the torch oscillation position. The oscillation vertical position can be adjusted in a way that the user can overlap the signals to facilitate the comparison. The Y-axis scale of the graph can be adjusted. Zooming and pan functions are also available. Four cursors are available for amplitude comparisons. The cursors are of the same colour code of the signals. The difference values are located on the bottom left of this screen. Graph major gridlines can be used as well.

An important output of this analysis is shown in the graph by small symbols represented as yellow squares and red circles overlapped with the waveforms. These symbols represent the torch in one of its four detected positions from the LVDT signal:

a) extreme left with a solid red circle
b) extreme right with an empty red circle
c) centre of oscillation coming from the left with an empty yellow square
d) centre of oscillation coming from the right with a solid yellow square

The symbols can be overlapped on voltage peaks or voltage average waveforms by user selection. They can also be deactivated. The graph data can be automatically exported to Microsoft Excel by pressing the buttons visible on the screen. The "Export to Excel" button exports the entire waveforms shown in the graph whereas the "Export Osc_Peaks to Excel" only exports the data (time and voltage) relative to the four coloured symbols described earlier.

4.2.3 WeldData – peak voltages versus background voltages analysis screen

For the purposes of this project, this analysis demonstrates the comparison between voltage peaks extracted from voltage peak pulses and background voltages extracted in the pulse background period. Figure 4.15 shows the analysis screen. The base and features of this screen are similar to the previous screen.

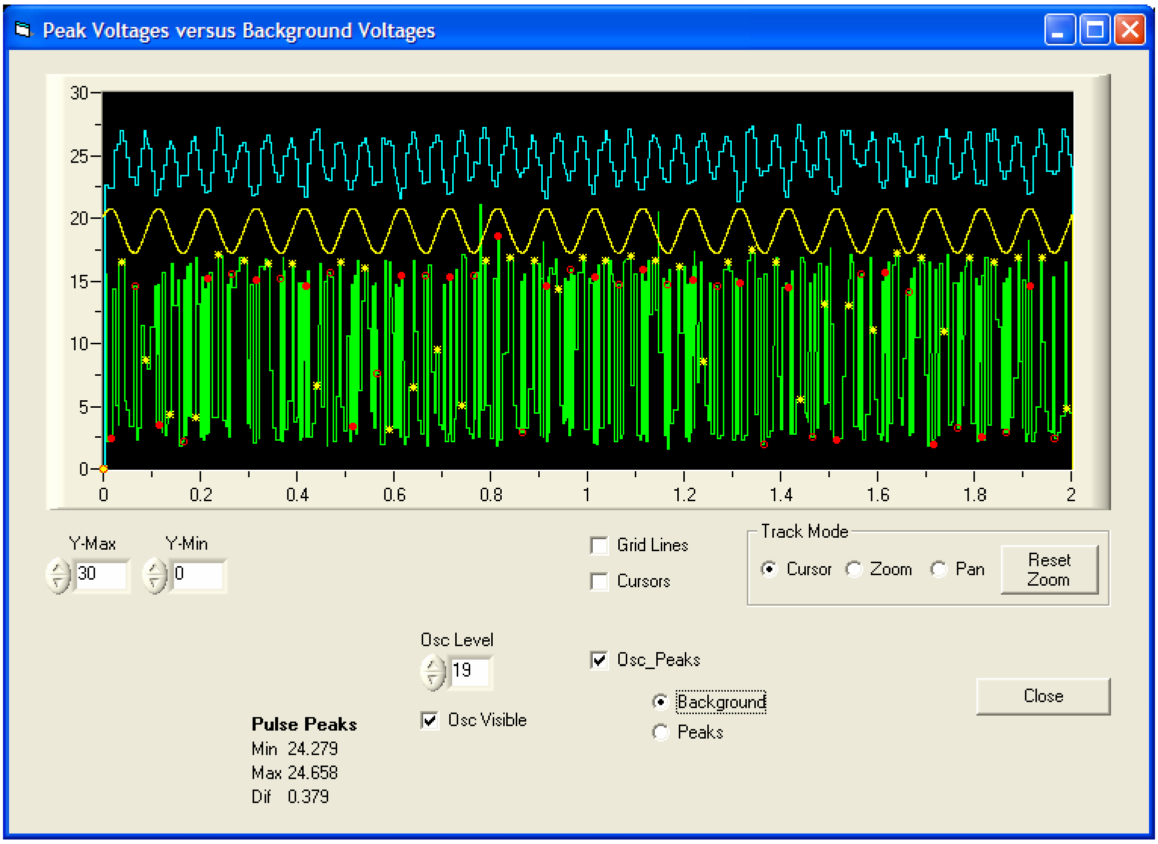


Figure 4.15 - Peak voltages versus background voltages analysis screen

Background voltages where detected in a similar way to the peak voltages except that instead finding a maximum voltage in each pulse cycle, the minimum voltage was found. In case of regular dip transfer as shown in Figure 4.15, the minimum value found for the background voltage will be the short-circuit voltage value.

For this analysis, torch oscillation position is represented by three coloured symbols indicating respectively:

e) extreme left with a solid red circle
f) extreme right with an empty red circle
g) centre of oscillation with a yellow asterisk

4.2.4 WeldData – average current

This analysis was developed to show the evolution of the pulsed current waveform average. The moving average algorithms used are similar to previous ones already explained.

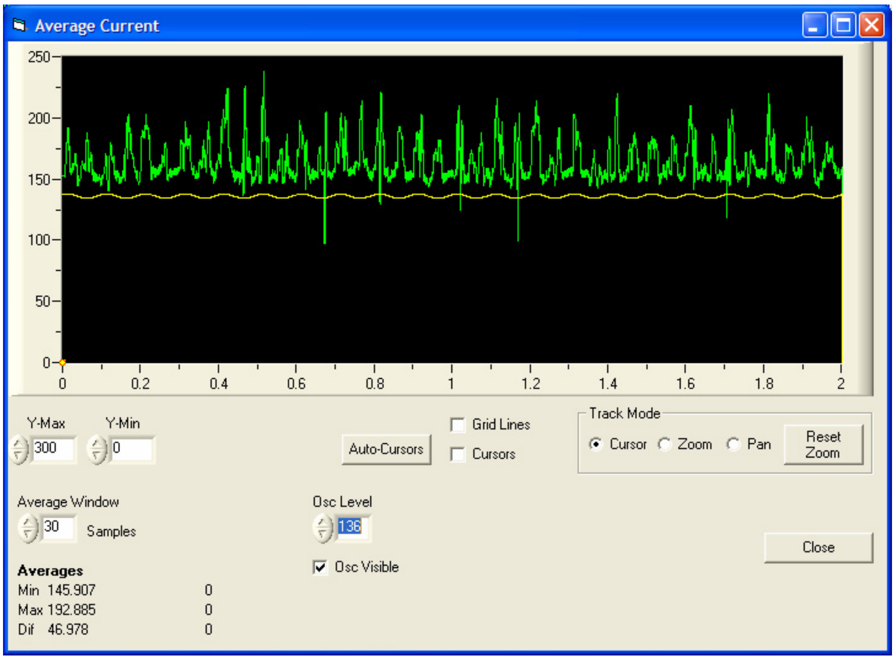


Figure 4.16 – WeldData – Average current

Figure 4.16 shows the results of a moving average with an average window of one pulse cycle. Relationships between averaged current and torch oscillation can be drawn with this analysis.

4.2.5 WeldData – Current versus Voltage Cross-plot

Cross-plots are a type of analysis frequently used by researchers to identify patterns and trends in the current/voltage evolution. Figure 4.17 shows a cross-plot created by plotting simultaneous values of voltage and current. This type of plot provides additional information on the behaviour of the welding process.

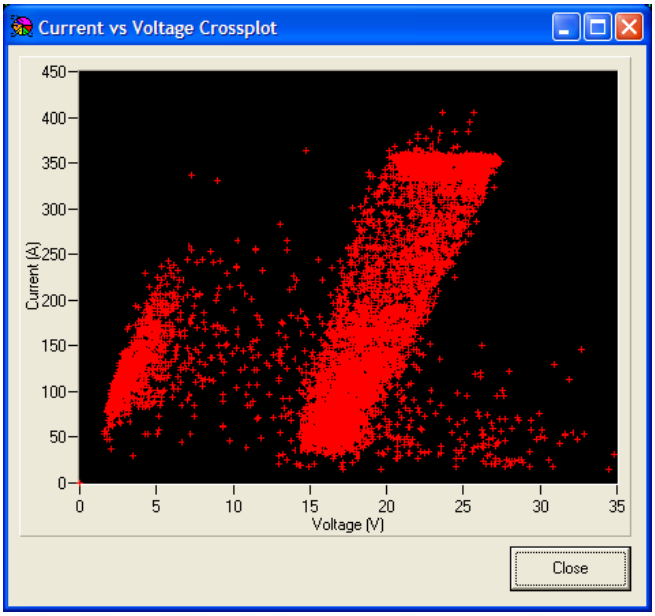


Figure 4.17 – WeldData – Current versus voltage cross-plot

4.3 Other software algorithms developed for data analysis

Experimentation phase 3 was mainly analysed by the use of WeldData software described in section 4.2, but phase 1 and 2 were based only on spreadsheet analyses. Data handling and automated calculation algorithms were programmed in Microsoft Visual Basic ® for applications included in Microsoft Excel 2003 ®. It was through this process that software algorithms were developed to deal with the raw data files from the digital oscilloscope (phase 1) and also from CAN messages (phase 2). The source code of the different algorithms can be found in Appendix A for phase 1 and Appendix C and Appendix D for phase 2.

4.3.1 Experimentation phase 1 - analysis algorithms

Figure 4.18 shows the main screen of the developed program containing the analysis algorithms. 'Sheet Name' and 'Sheet File Name' are input fields automatically filled in by the program when it is executed.

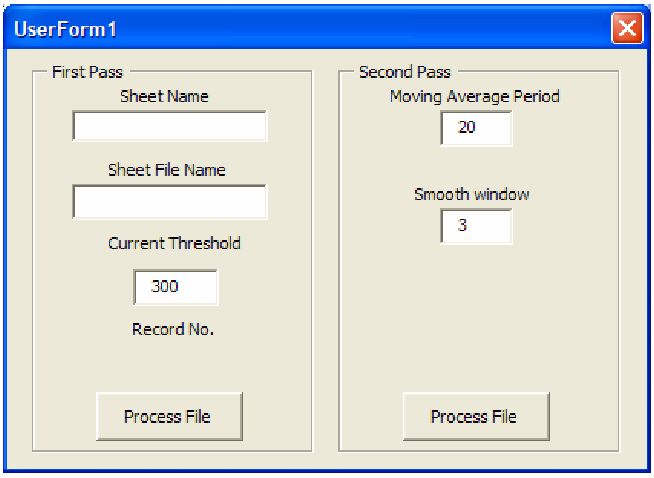


Figure 4.18 – Experimentation phase 1 – analysis algorithms main screen

The software is processed in two passes. The first pass opens the raw data file and extracts the peak voltages based on the peak current. The 'Current Threshold' value (300 as in the main screen of Figure 4.18) establishes the threshold line beyond which a current pulse is considered valid. This algorithm works as described in the flowchart of Figure 4.9. The end result is a column in the spreadsheet with all the peak voltages found in the raw data file. The second pass uses this filtered data and creates a smoothing average using the input parameters found in the main screen. This moving average algorithm is similar to the flowchart of Figure 4.6. At the same time, it performs maxima and minima findings using the moving average as threshold level. It implements a similar peak finding algorithm in two ways; it detects maximum values when the data is over the moving average and minimum values when it is below. The resultant data is divided in three columns separated by centre, left and right side torch positions of oscillation. This is done by rotation, i.e., the new value found will belong to one of the three torch positions simulating the torch behaviour: left-centre-right-centre. A sample output of the generated graph can be seen in the results chapter, Figure 5.3 (p. 98). The blue line in this graph represents the output data of first pass and the others from second pass.

4.3.2 Experimentation phase 2 - analysis algorithms

Due to the complexity of having two separate data sources (oscilloscope and CAN messages) and for better results demonstration, the analysis is presented in two parts. The first part deals with CTWD control, involving arc signals and CAN messages and the second part with cross seam position only, with CAN messages.

4.3.2.1 CTWD control analysis algorithms

For the analysis of CTWD control, the algorithms are presented in two parts: signal analysis and CAN message analysis. The final result was the combination of both results as shown by the graphs in the results chapter (section 5.2.1).

Signal analysis of the CTWD control was performed by developing computer algorithms replicating the VISENSE control algorithms. Hence, arc signals acquired during experimentation were used as the input of the computer replicated algorithms and the resultant output plotted on graphs. Thus, it is possible to analyse the behaviour of the VISENSE control algorithms during experimentation.

Figure 4.19 shows the main screen of the program developed containing the algorithms. ‘Worksheet Name’ is an input field to define the spreadsheet name where the processed data should be put in. The browse button opens the Open File Dialog to help on the selection of the raw data file to be used. ‘Smoothing factor’ and ‘Initial Voltage’ are two parameters for the moving average calculation. Their values should be the same used by the VISENSE device for the control.

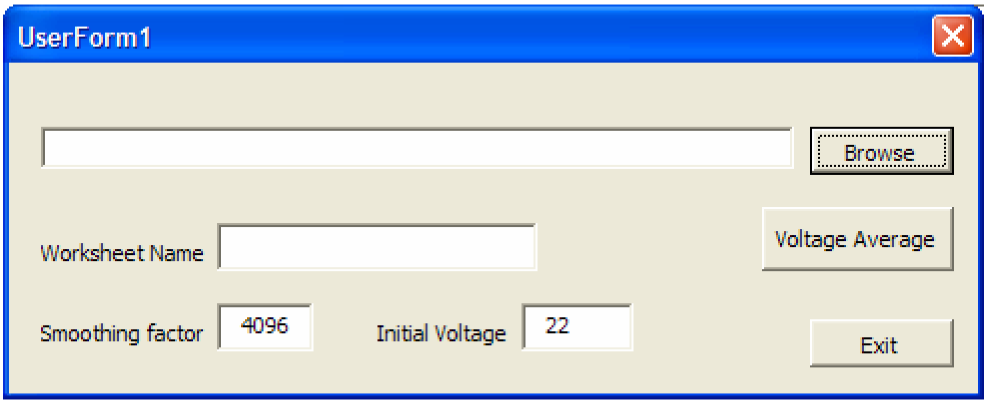


Figure 4.19 – CTWD control analysis main screen using arc signals and voltage moving average

VISENSE control algorithm for CTWD control is the moving average algorithm, as described in the flowchart of Figure 4.6. The algorithm developed for this analysis was a replica of it. The results are two spreadsheet columns the first one being the sample time and the second the averaged voltage at that time.

For the CAN messages, the objective was to extract the relevant data from torch up and down motion commands. The logged CAN messages have the following structure:
Date<tab>Time<tab>Send (TX) or Received (RX) Message; Timestamp (t) with 10 µs resolution; Message identification (id); Message length in hexadecimal characters(l); Data in hexadecimal format

The following is an example of logged CAN messages:

10-18-2004 14:32:18 RX_MSG c=0, t=1960966252, id=0251 l=2, 0005 tid=00
10-18-2004 14:32:18 RX_MSG c=0, t=1960974051, id=03A1 l=2, 0000 tid=00
10-18-2004 14:32:18 RX_MSG c=0, t=1960974064, id=0261 l=2, FFFD tid=00

As shown in the above example, the Time field is in the same second for the three messages. There is not enough resolution in this field to be used for correct message timing. On the other hand, the timestamp field *t* is built from a 10 µs internal timer in the CAN board. This was essential to understand and position each arriving message in each time slot and to synchronise it with arc signal filtered data.

Figure 4.20 shows the program main screen.

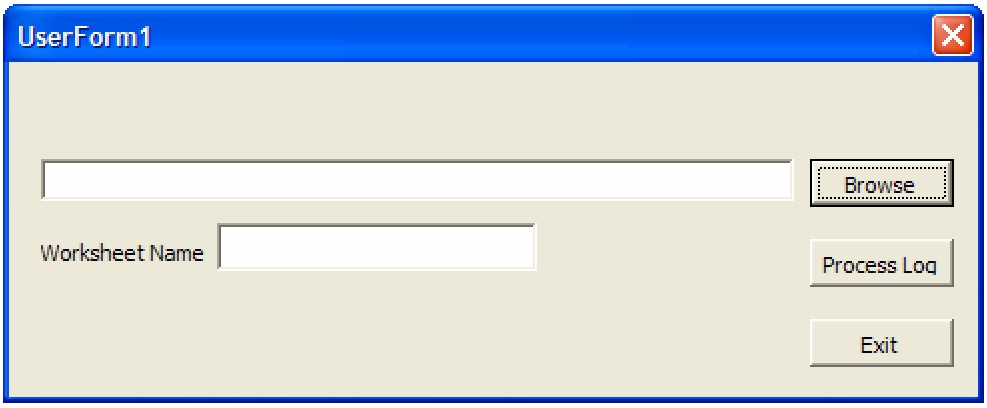


Figure 4.20 – CTWD control analysis main screen using CAN messages

'Worksheet Name' input field is the spreadsheet name where the filtered data should be put in and browse button is to open the File dialog window to choose the raw data file of logged CAN messages. The algorithm starts processing when 'Process Log' button is pressed by opening the raw file and seeking for torch up and down messages (id=0261). When a message is found, it converts the timestamp to a relative time inside the process and processes the message value (Data field) to determine the amount of correction sent in the message. This value is added to or subtracted from a summing variable and the record added to the spreadsheet. The final result is two columns in the spreadsheet containing the relative time of the message and the summing value of the correction. Therefore it is possible to observe in the resultant graph the torch up and down evolution along the welding run (Figure 5.7 – p. 101).

Both moving average and CAN messages filtered data is organised into a single spreadsheet and the graphs plotted. Standard deviation of the voltage averaged signal was also calculated.

4.3.2.2 Cross seam position control analysis algorithms

This analysis was performed similarly to the CTWD control CAN messages analysis. The main difference is the message identification (id=0251). The modus operandi of the algorithm is the same where time and amount of correction is extracted from the logged CAN messages file. The resultant two columns in the spreadsheet are the time and the summing result of the corrections at each time as shown in the graphs of Figure 5.10, Figure 5.11 and Figure 5.12 (p. 103 and p. 104).

For standard deviation calculation, only Microsoft Excel data manipulation was used. First a linear regression was calculated and the resultant equation used to calculate the absolute difference between the real value and the predicted value of each point. This data was then organised into a third spreadsheet column and the standard deviation calculated with it.

5 Results

The results demonstrated in this chapter come from the three experimentation phases described in chapter 3. Through-the-arc sensing for GMAW-P was initially assessed in phase 1. CTWD and cross seam position control were evaluated in phase 2. The effect of torch oscillation frequency and sidewall proximity on arc signal sensitivity and weld metal penetration was assessed and evaluated in phase 3, to optimise control of groove width in the 5º narrow groove with GMAW-P. The analysis was mainly based on detailed processing of the arc voltage and current data using the techniques described in section 4.2. For phase 3, weld metal macrosections were also taken for assessment of weld quality.

5.1 Experimentation phase 1 – Initial trials

Figure 5.1 shows a 300 ms period of raw voltage and current data from trial A1. It is apparent from Figure 5.1 that current waveform pulses do not change with time whereas voltage waveform pulses are clearly modulated.

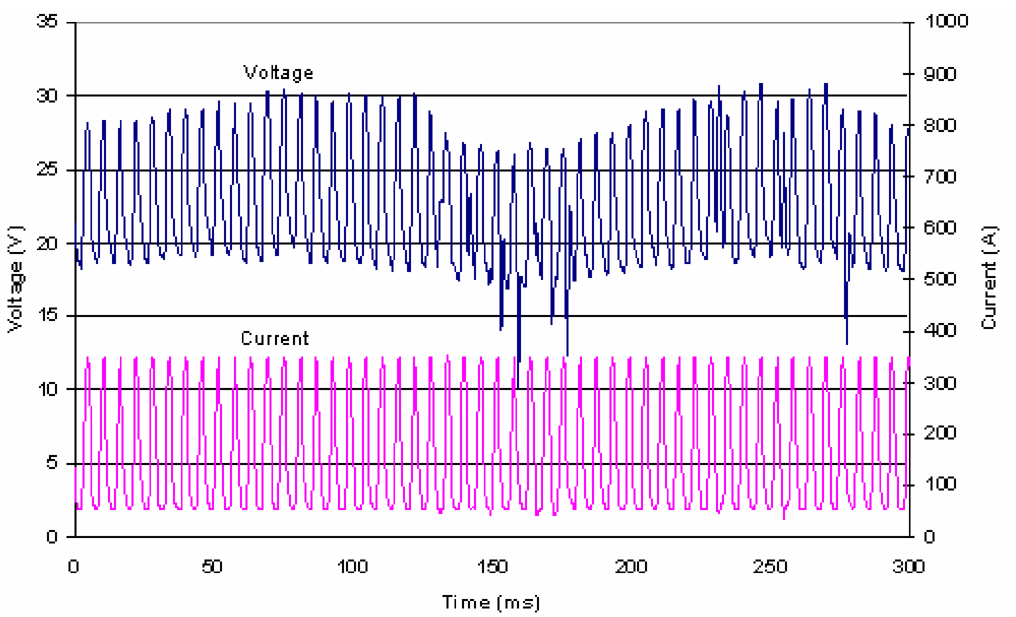
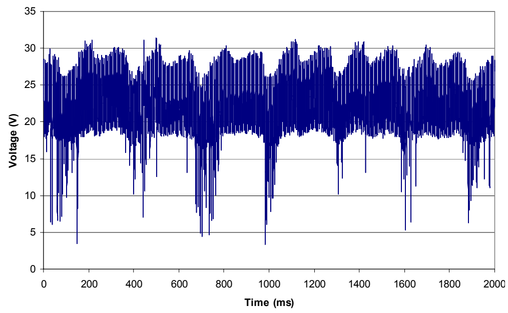
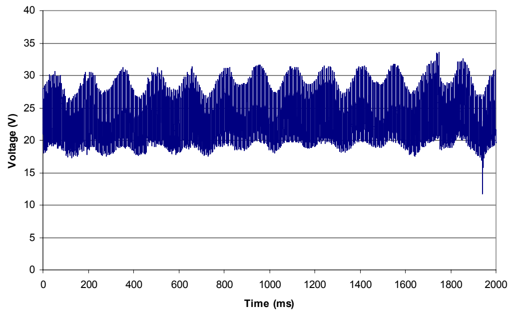


Figure 5.1 – Current and voltage data from trial A1.

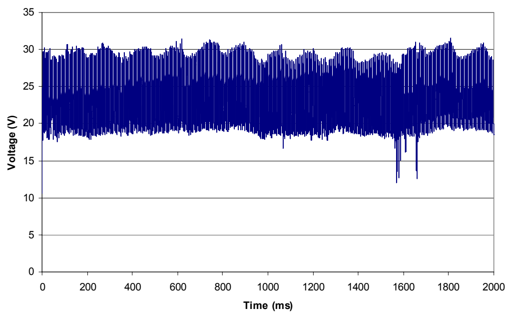
Figure 5.2 shows 2s period of raw pulsed voltage data from trials A1 to A6. Although not visible individually, voltage pulses vary in time creating the clear modulation in the waveforms of Figure 5.2. The modulation frequency in each case is equal to the torch oscillation frequency.



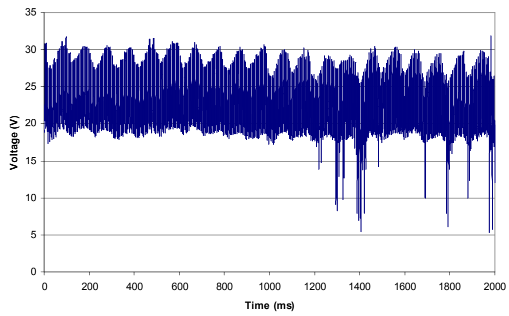
Trial A1 - **OF** 3.33 Hz - **OW** 5 mm



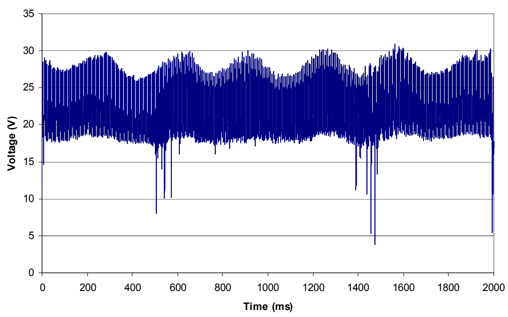
Trial A2 - **OF** 3.33 Hz - **OW** 6 mm



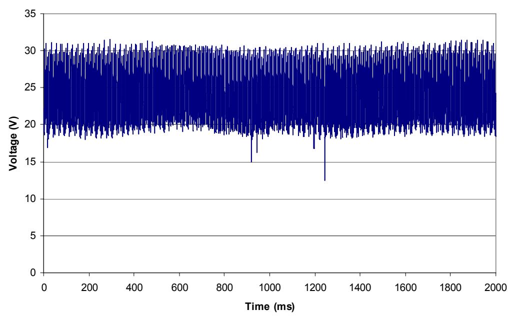
Trial A3 – **OF** 3.33 Hz – **OW** 4 mm



Trial A4 – **OF** 5 Hz - **OW**5 mm



Trial A5 – **OF** 1.67 Hz - **OW** 5 mm



Trial A6 - **OF** 0 Hz - **OW** 0 mm

OF – ***Torch Oscillation Frequency***

OW – ***Torch Oscillation Width***

Figure 5.2- Raw voltage data from trials A1 – A6

Detailed results from each trial can be found in Appendix I. The original data was processed in two passes as follows: first a maximum finding algorithm was performed to extract the peak voltage values of each pulse. This algorithm is similar to the one described in Figure 4.9 (p. 76). Arc current was used to detect valid pulses and ignore spurious pulses as indicated in Figure 4.8 (p. 75). A threshold value of 300 A for the current was used to verify when a valid pulse occurred. The initial signal data was extracted directly from the raw data files and the resultant peak voltages were exported to a spreadsheet. By this means, the lower frequency modulations apparent in the voltage signal that represent the effects of weaving from one side to the other of the weld preparation could be isolated. The results of this operation can be seen in Figure 5.3 as the oscillating trace.

The second pass uses maximum and minimum finding algorithms on the data generated by the first pass. The results of this pass are represented in the graphs of Figure 5.3 and Figure 5.4 as the blue, red and green lines. In Figure 5.3 the first minimum corresponds to the approach of the arc to the left hand (LH) wall of the weld preparation, and the second minimum corresponds to the approach of the arc to the right hand (RH) wall. Since the LH voltage is significantly lower than the RH voltage, it is apparent that in this trial the torch is off centre, and closer to the LH than the RH wall (Blue and Red lines in Figure 5.3 and Figure 5.4). This pattern is repeated throughout the trace. The first voltage maximum (Green line in Figure 5.3 and Figure 5.4) corresponds to the centre position of the torch oscillation, and this value, and subsequent maxima can be related to arc length and CTWD at the centre position.

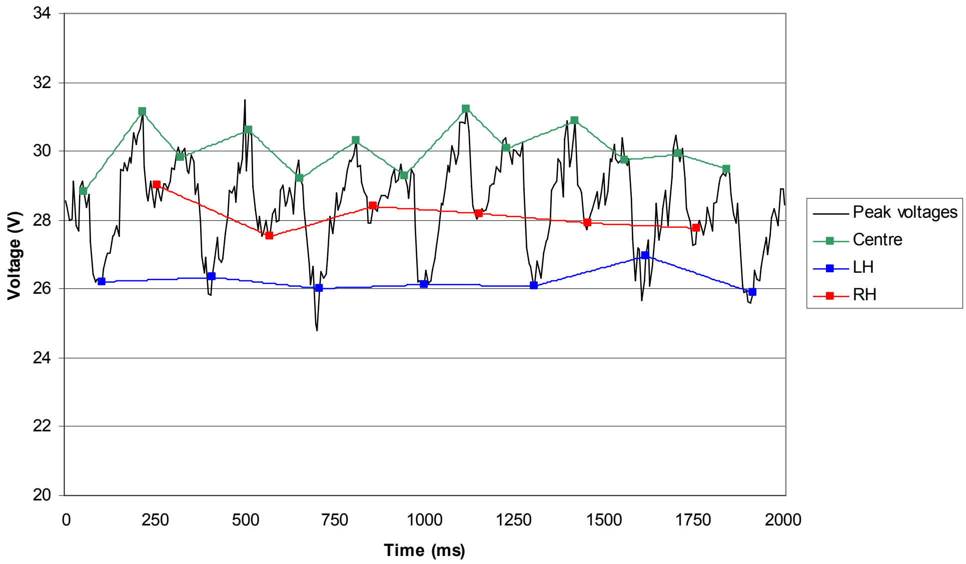


Figure 5.3 - Processed data from applying the first and second pass algorithms to trial A1 voltage data

The effects of the second pass algorithm can be seen more clearly in Figure 5.4 where the first pass data has been removed.

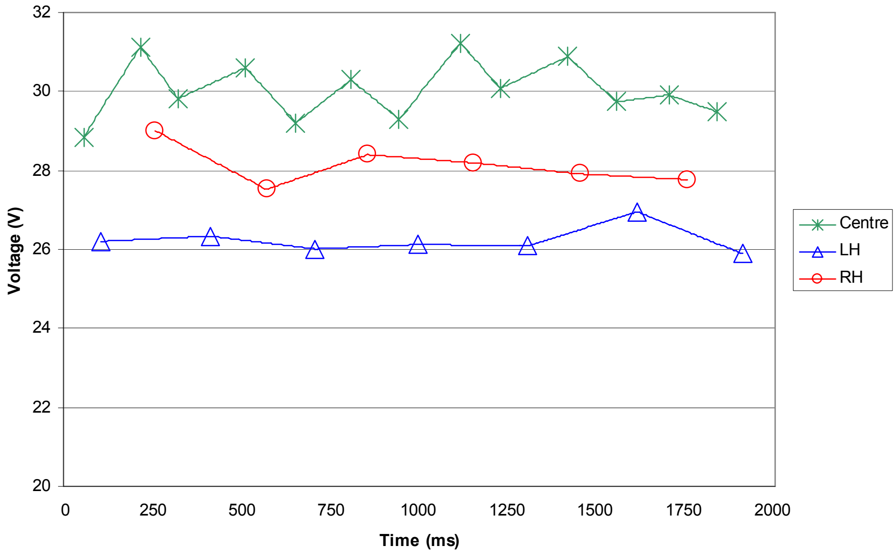


Figure 5.4 - Output data at the conclusion of the second pass algorithm from trial A1

The raw data from trials A1 to A6 was processed by the algorithms. From each trial, LH and RH voltage data were averaged and subtracted from the maximum voltage data, to generate the graph shown in Figure 5.5, plotted against oscillation width. The relationship between torch oscillation width and peak voltage difference between the values at the centre of oscillation and the values at maximum torch excursion is clear from the graph.

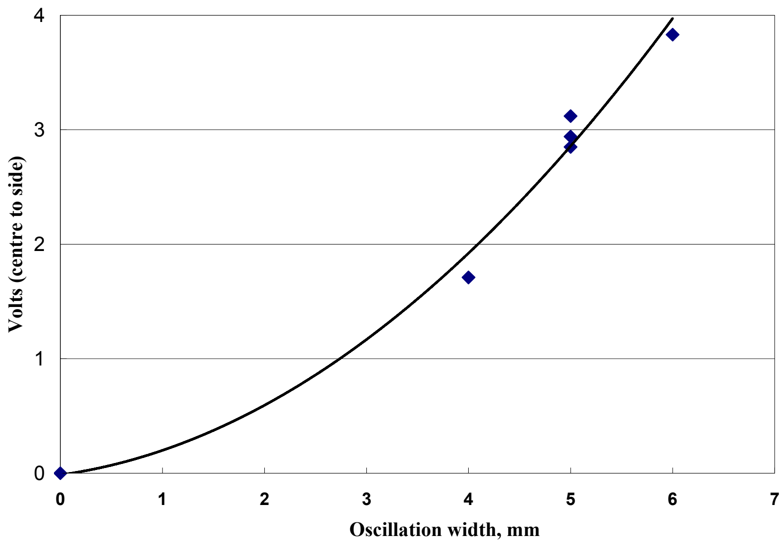


Figure 5.5 - Oscillation width influence on peak voltages (centre – sidewall)

5.2 *Experimentation phase 2 – Control algorithms test bed*

The results presented in this section were obtained from experimentation of the CTWD and cross-seam control algorithms implemented in the VISENSE device, and described in section 4.1. This experimentation was a test bed for the developed algorithms and was detailed in section 3.2. For the purpose of testing the control algorithm's reliability in correction speed and accuracy, and also to test robustness and adaptability of the algorithms to a new situation, the welding equipment and procedure were replaced for phase 2 work. The data shown in this section can be seen detailed in Appendix E. The software algorithms developed for this analysis are described in section 4.3.2.

An initial relationship was extracted from trials B5 to B8 and is shown in Figure 5.6. The results of voltage average against CTWD are plotted in this graph. This test was performed with the same welding setup of trials B1-B4 but outside the groove (bead on pipe), on runs of 6 s. These results are essential for further calculations because they relate voltage average variation with CTWD length. The resultant slope value (0.5045) of the calculated regression will be used as a multiplication factor in section 5.2.1 to determine the standard deviation of voltage average against torch height corrections.

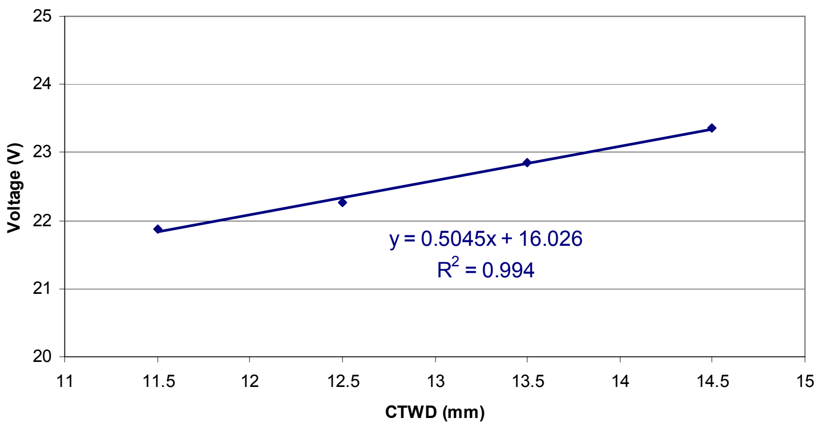


Figure 5.6 – Voltage versus CTWD relationship

To better demonstrate the following results, they were divided in two sections: CTWD control and cross seam control. It should be noted that no valid data was obtained from trial B2 due to setup problems. This trial only performed 50% of the total welding length (120 mm) creating a step of 2.8 mm height between pass 1 and pass 3 (trials B1 and B3 respectively).

5.2.1 CTWD control

CTWD control results are shown in the graphs of Figure 5.7 to Figure 5.9 in the form of voltage moving average (blue line) and torch height variation or vertical corrections (red line), extracted from trials B1, B3 and B4 respectively. Trials B3 and B4 were made on previous beads that varied in height, in order to evaluate the performance of the system in making corrections. Torch height variation was obtained by the control messages sent by the VISENSE device via CAN. Torch height standard deviation is shown in millimetres in each graph. It was calculated by multiplying the standard deviation of voltage data by the slope factor calculated previously and shown in Figure 5.6.

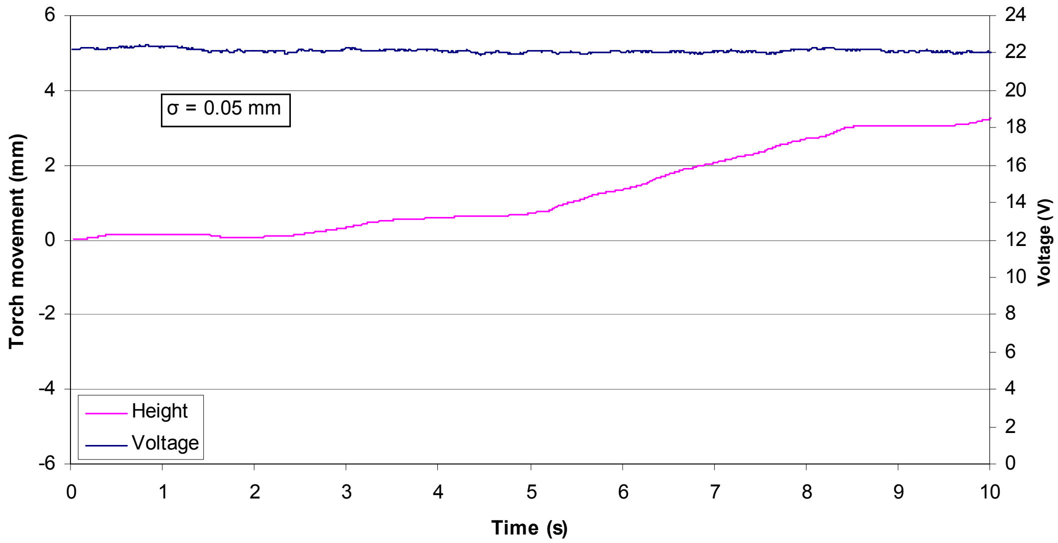


Figure 5.7 – Trial B1 – Torch height variation and voltage moving average

The graph of Figure 5.7 shows a small fluctuation of voltage average with a standard deviation of 0.05 mm, with the torch moving upwards 3 mm, during the 10 s of the experiment.

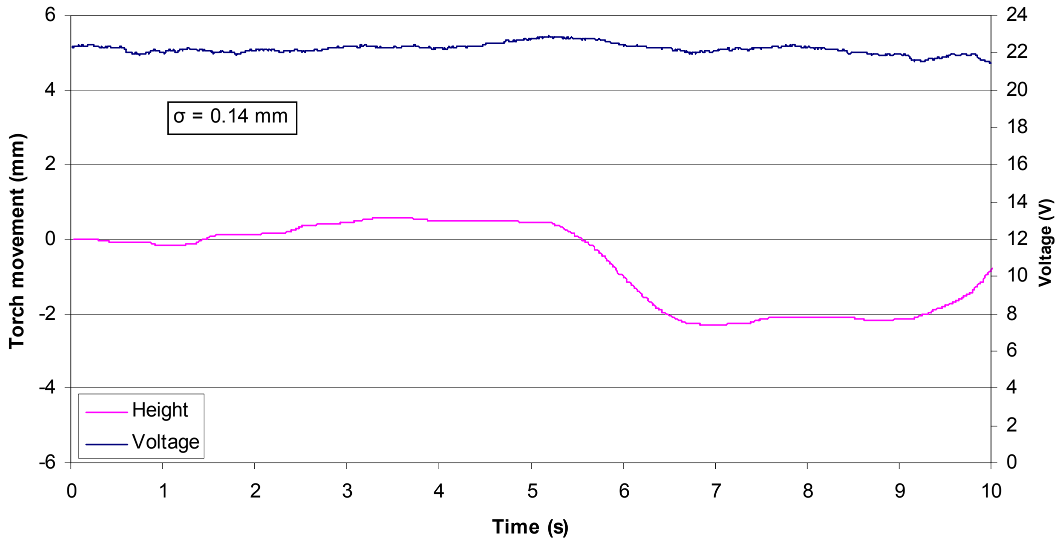


Figure 5.8 - Trial B3 – Torch height variation and voltage moving average

The graph of Figure 5.8 also shows a small fluctuation of voltage average reflected in a standard deviation of 0.14 mm. The torch moved slightly upwards 0.4 mm until the experiment completed 5.5 s. Then it dropped 2.8 mm in 1 s, staying constant for 3 s, rising 1 mm in the last second of the experiment.

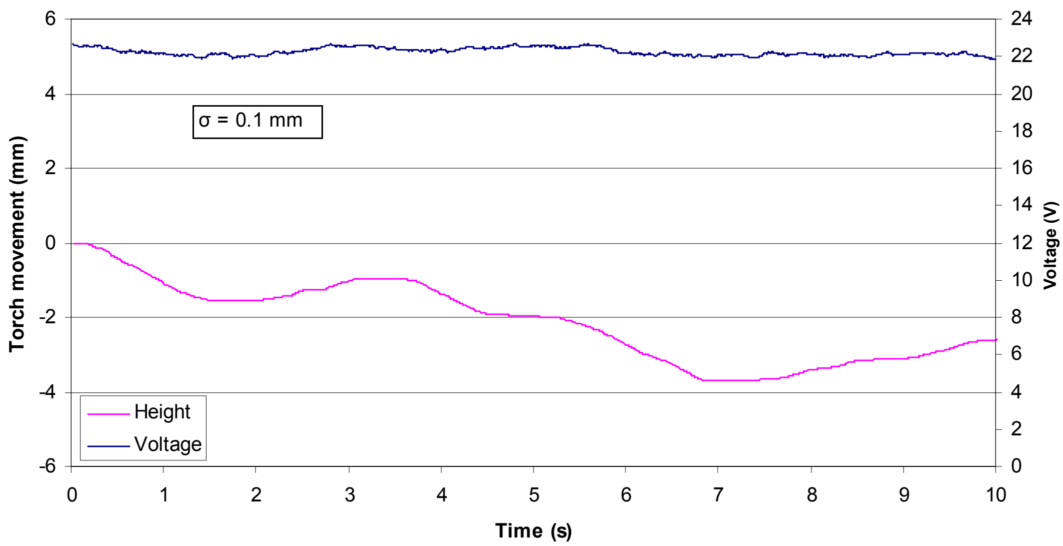


Figure 5.9 - Trial B4 – Torch height variation and voltage moving average

The graph of Figure 5.9 shows a small fluctuation of voltage average creating a standard deviation of 0.1 mm. The torch height had a downward trend with strong fluctuations in the initial 7 s of the experiment reaching a minimum of -3.8 mm, finishing the experiment with -2.3 mm.

5.2.2 Cross seam control

Cross seam control is shown in the graphs of Figure 5.10 to Figure 5.12 for trials B1, B3 and B4 respectively. The blue line of the graphs illustrates torch path horizontal corrections in millimetres, obtained from the CAN messages. A positive value means a torch path correction to the right and a negative value a torch correction to the left, following the torch direction. This correction value is fixed (0.1 mm) allowing the magnitude of the corrections to be determined. The red line is a linear regression of the blue line with the expression presented in each graph, along with the standard deviation and R^2 (correlation coefficient).

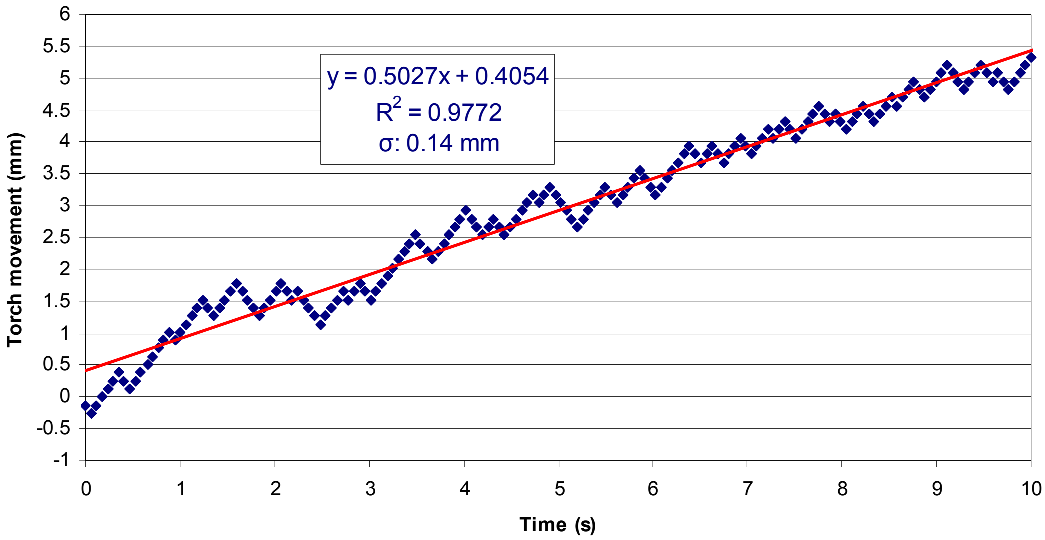


Figure 5.10 - Trial B1 – Cross seam control position

Figure 5.10 shows a linear trend, with some variations around the mean position, finishing at maximum of 4.8 mm. The trend slope is 0.5027 with a correlation coefficient of 0.9772 and a standard deviation of 0.14 mm. Maximum deviation from the trend line at any point is 0.68 mm.

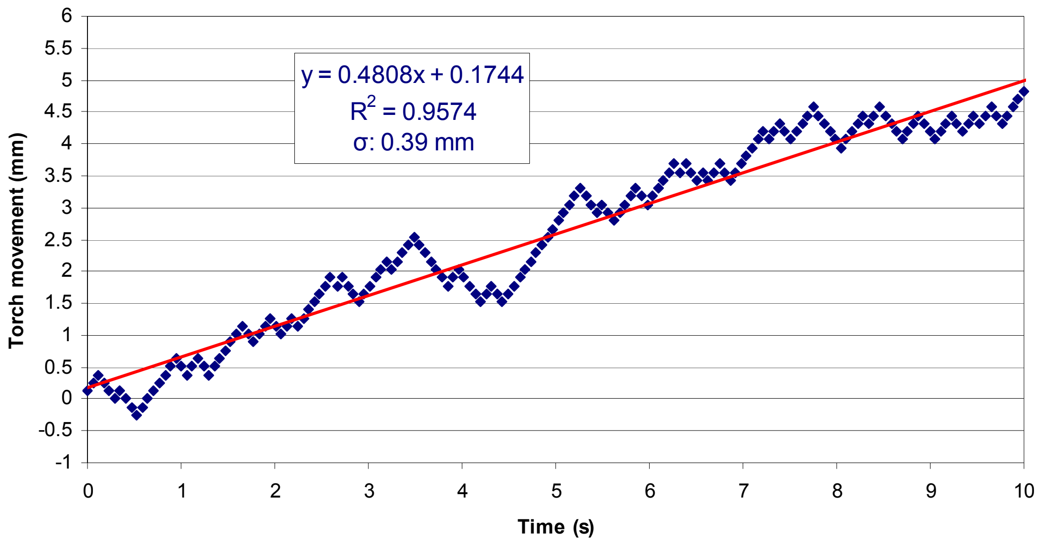


Figure 5.11 - Trial B3 – Cross seam control position

Figure 5.11 also shows an overall linear trend, finishing at maximum of 4.8 mm. Some tracking instabilities seem to have occurred between 3.5 and 4.5 s. The trend slope is 0.4808 with a correlation coefficient of 0.9574 and a standard deviation of 0.39 mm. Maximum deviation from the trend line is 0.78 mm.

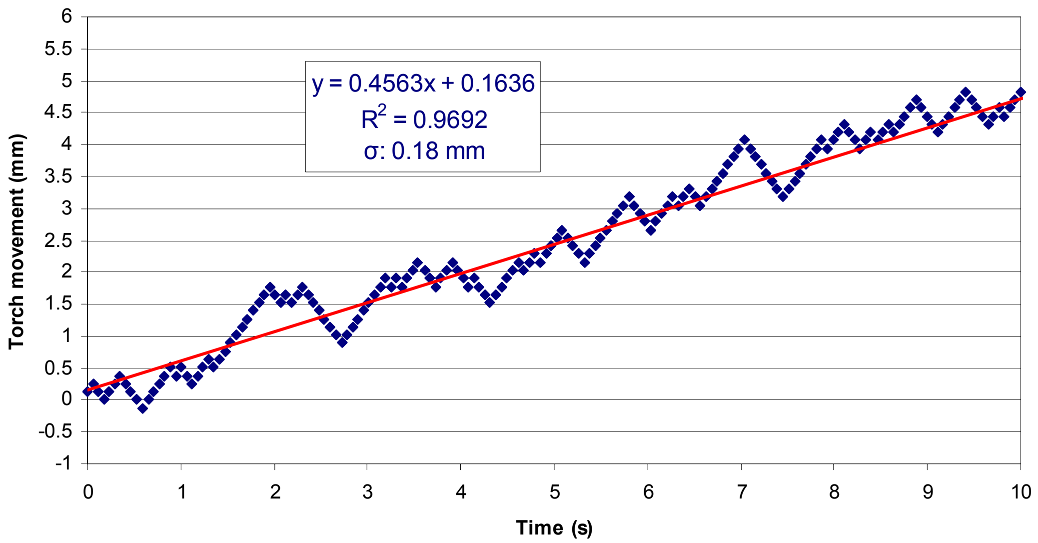


Figure 5.12 – Trial B4 – Cross seam control position

Figure 5.12 again shows a linear trend, finishing at maximum 4.8 mm. The trend slope is 0.4563 with a correlation coefficient of 0.9692 and a standard deviation of 0.18 mm. Maximum deviation from the trend line is 0.72 mm.

Figure 5.13 to Figure 5.15 are top view pictures of the weld bead after each trial. Figure 5.16 is a side picture showing the down step after trial B4.

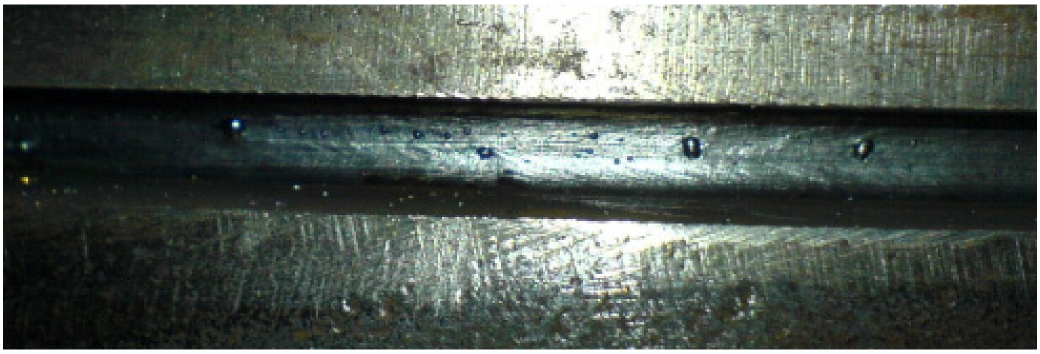


Figure 5.13 – Bead profile after trial B1

Figure 5.14 – Bead profile after trial B3

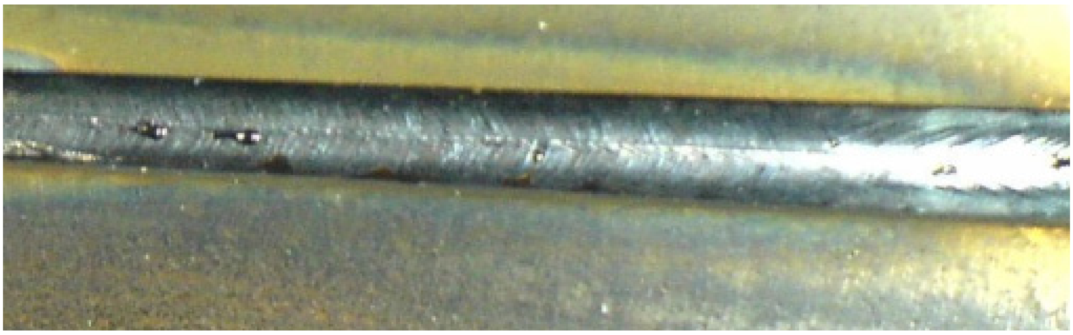


Figure 5.15 – Bead profile after trial B4

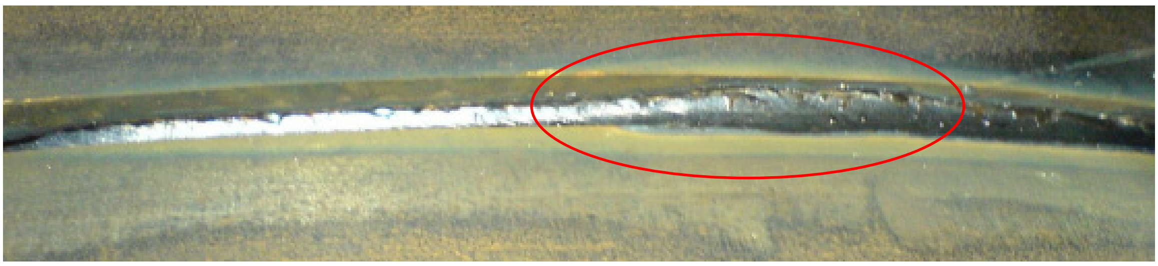


Figure 5.16 – Side bead shape after trial B4 with visible down step of 2.8 mm

5.3 Experimentation phase 3 – Torch oscillation width and frequency

Experimentation phase 3 was devised to study the influence of wire proximity to sidewall, at different torch oscillation frequencies and CTWD, on arc peak voltages at specific torch positions and on weld metal penetration. A rig with a high speed oscillation torch was specially developed to perform the experiments. Voltage sensitivity with oscillation frequency was also analysed by comparing experiments conducted in both GMAW-P and CV GMAW processes.

To achieve groove width control using through-the-arc sensing in the 5º narrow groove with GMAW-P, it is important to know what values of wire proximity to the sidewall, torch oscillation frequency and CTWD must be used to obtain the optimum weld metal penetrations and minimise welding defects. This in turn will reveal what arc peak voltages should be followed at different torch positions, to achieve optimum control.

The results for this phase are divided in three sections:

a) Arc signal sensitivity comparison between CV GMAW and GMAW-P with a 45° plate angle
b) Single sidewall trials with 5° preparation angle
c) Double sidewall (groove) trials with 5° preparation angle

5.3.1 Arc signal sensitivity comparison between CV GMAW and GMAW-P with a 45° plate angle

Trials C1 to C5 were performed on a plate at an angle of 45° to the vertical as shown in Figure 3.12 (p. 55), to ensure that in each torch oscillation excursion, CTWD is changed. CTWD is made of two lengths (arc length and wire extension) that are constantly balancing themselves to keep the melting rate equal to the wire feed rate. When they are balanced, this system is known to have reached an equilibrium state. The variation of CTWD creates a rebalancing of the system equilibrium by instantaneous changes in arc length and a gradual adaptation of wire extension, until the new balance is found. Wire extension variation to the new balance point takes a certain amount of time (typically tenths of millisecond).

Complete system equilibrium recovery may not be successfully achieved if CTWD variations are sufficiently fast. In other words, by rapidly oscillating the torch up and down, arc length changes are reproduced instantaneously but wire extension may not achieve the full balance point. Arc length is directly related to arc voltages and so, voltage variation (sensitivity) to new CTWD values is affected by torch oscillation frequency. Estimating recovering times and its influence in arc voltage sensitivity is important for arc sensing based control algorithms development.

In this set of experiments, arc and torch position signals acquired during the trials were post-analysed by the WeldData software described in section 4.2. Each analysed file corresponds to each experiment made at a different oscillation frequency and CTWD as shown in Table 3.4.

The software extracted peak voltage of each pulse at four positions of the torch: extreme left, extreme right, centre coming from left and centre coming from right positions. In other words, when the torch crosses one of these positions, the last peak voltage at that position is used for GMAW-P. For CV GMAW trials, voltage was measured in each of the four torch positions. Each trial generates a number of voltage values equal to the total number of torch oscillations acquired multiplied by the four torch positions. These voltage values were exported to a spreadsheet and averaged to obtain a representative voltage value for each torch position in each trial. This technique has revealed a different consistent voltage value characteristic of the arc length recovery process. After processing all files from the trials, the spreadsheet contains four columns with an average value of the voltages found on the left, on the right, centre from left and centre from right, torch positions.

The final step was the calculation of voltage difference between the torch extreme and centre positions resulting in Left-Right Voltage for extreme values and CRight-CLeft Voltage for centre values. The order of the subtractions was the order of the higher value minus the lower value in the majority of the cases. Figure 5.17 to Figure 5.19 shows the graphs with the results of each voltage difference plotted against the respective oscillation frequency. Voltage difference at torch extremes (Left-Right) is represented by the pink dots and at torch centre (CRight-CLeft) is represented by the blue dots. The respective coloured lines are polynomial regressions of the plotted values.

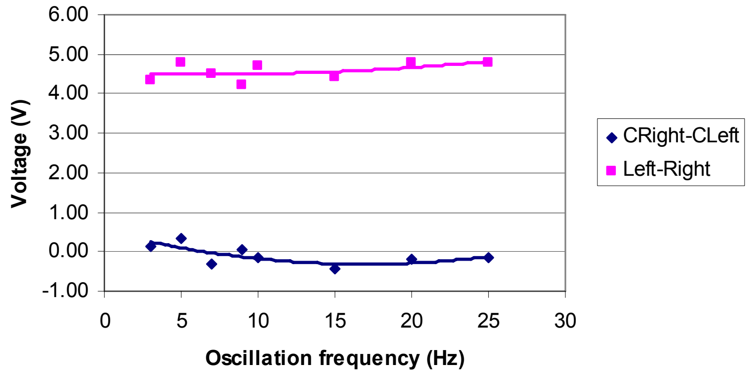


Figure 5.17 - Pulse peak voltage variation versus torch oscillation frequency for a CTWD between 13.2 mm and 15.7 mm with GMAW-P, trials C1 and C3

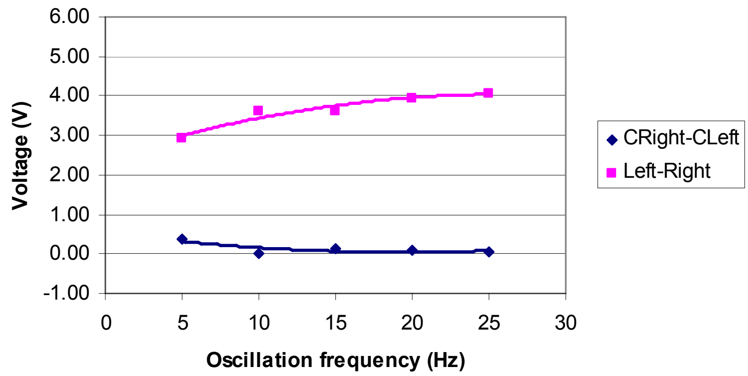


Figure 5.18 - Pulse peak voltage variation versus torch oscillation frequency for a CTWD between 15.4 mm and 17.6 mm with GMAW-P, trials C2

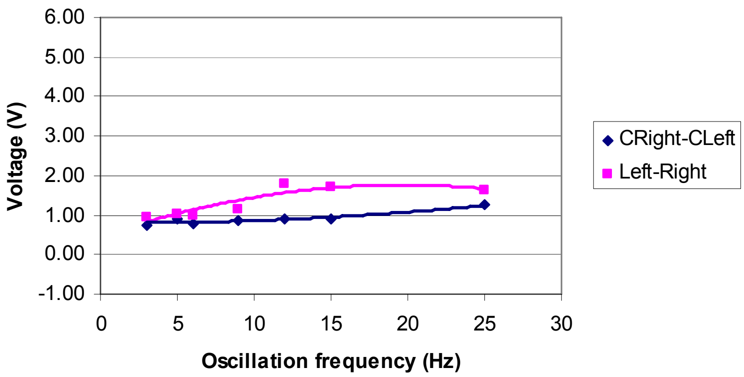


Figure 5.19 - Voltage variation versus torch oscillation frequency for a CTWD between 13.2 mm and 15.7 mm with CV GMAW, trials C4 and C5

From the graphs of Figure 5.17 and Figure 5.18 (GMAW-P) it is apparent that Left-Right voltages have different behaviours. With a CTWD varying between 13.2 mm and 15.7 mm, the graph of Figure 5.17 shows a small variation of Left-Right peak voltages found between 4 V and 5 V. On the other hand, with a CTWD varying between 15.4 mm and 17.6 mm, the graph of Figure 5.18 shows a voltage increase with oscillation frequency and voltage values between 3 V and 4 V. CRight-CLeft voltages were around 0 V.

For CV GMAW, the graph of Figure 5.19 shows a 80% increase of Left-Right voltages with oscillation frequency, mainly over 10 Hz. CRight-CLeft voltages also show a slight increase with oscillation frequency with its value always around 1 V.

It is clear from the graphs that there are significant differences in Left-Right voltage difference between GMAW-P and CV GMAW (between 3 V and 5 V for GMAW-P and lower than 2 V for CV GMAW). CRight-CLeft voltage also shows an apparent difference for both welding processes (GMAW-P always below 0.38 V with some negative values and CV GMAW always above 0.75 V). An important remark from the graphs of GMAW-P is the oscillation frequency influence in Left-Right voltage. It is more apparent for higher CTWD values, although with lower Left-Right voltages.

5.3.2 Single sidewall trials with 5° preparation angle

Trials D1 to D4 were performed with a single 5° sidewall on the right side (Figure 3.13 – p. 58) to evaluate arc behaviour for arc signals sensitivity analysis and weld metal penetration, using GMAW-P. The trials were conducted with variations in torch proximity to the sidewall, torch oscillation frequency and CTWD. The objective was to understand the variation in arc signals sensitivity and in both bottom and lateral weld metal penetrations. Arc signal analysis, high speed image visualisation and metallographic images from the welded specimens were the methods used to extract the information demonstrated in these results.

This analysis was performed in a similar way of the previous study in section 5.3.1. Peak voltages from the three torch positions were extracted by WeldData and averaged per torch position from each trial file. The result was exported to a spreadsheet forming four columns of averaged peak voltages for each trial from each torch position: torch at extreme left, torch at extreme right, torch at centre coming from the left, torch at centre coming from the right. A secondary spreadsheet was created with the weld metal penetrations of each trial. Three different analyses were then performed with the same data, to study the influence of the following factors on peak voltage sensitivity and weld metal penetration: Oscillation frequency, sidewall proximity and CTWD.

A different type of analysis was also performed with trial D4 to extract relevant information on arc behaviour with different values of CTWD. This study generated key information for GMAW-P dynamic state analysis for arc signal sensitivity.

5.3.2.1 Oscillation frequency influence for single sidewall trials

The data from both spreadsheets was grouped by trials with the same torch oscillation frequency and torch position. Both torch centre positions (CRight and CLeft) were merged by averaging and voltage from torch at left (opposite to the sidewall) was discarded. The data was then combined and the graphs of Figure 5.20 were plotted at different torch proximities. The graph shows the final values represented by coloured symbols and lines. The dark blue colour represents the resultant voltage with the torch at the sidewall or extreme right (Side) and the pink colour represents the resultant voltage with the torch at centre of oscillation (Centre). The green colour represents the resultant bottom weld metal penetration (Bottom P) and the light blue colour represents the sidewall weld metal penetration (Side P).

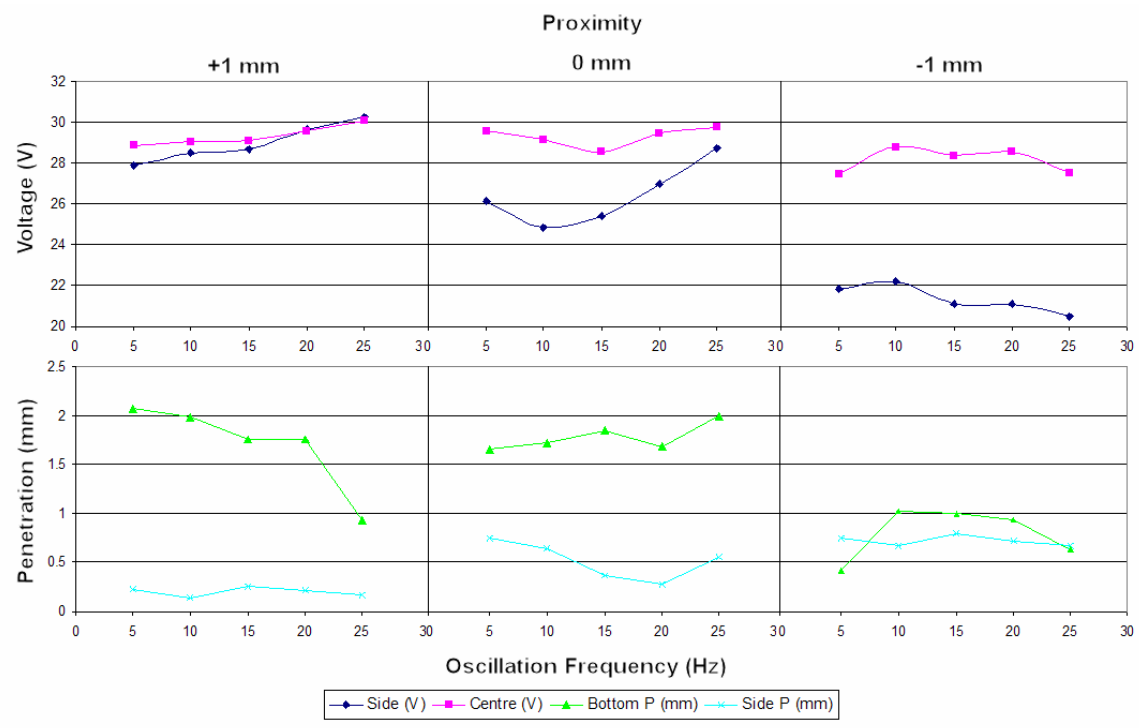


Figure 5.20 – Effect of oscillation frequency on peak voltage signals and weld metal penetrations in single sidewall trials, at different torch to sidewall proximities

Figure 5.20 show the graphs of torch oscillation frequency influence on peak voltages and weld metal penetration for different proximities to the sidewall. It is apparent from the graph that oscillation frequency increase produces a peak voltage increase in both Side and Centre positions at +1 mm proximity (28 V to 30 V), whereas a slight peak voltage decrease was found on Centre position at -1 mm proximity. At 0 mm proximity Side peak voltage increases by 3 V overall, although not a linear trend, with Centre voltages staying around 29 V. It is also apparent from the graph that proximity has a strong influence on peak voltage difference between Side and Centre torch positions, with a very small difference at +1 mm, and 6 V at -1 mm.

The graphs also show that weld metal penetration is influenced by oscillation frequency, depending on torch proximity to the sidewall. At +1 mm, it is apparent that oscillation frequency produces a reduction in the bottom penetration with no visible influence in sidewall penetration. At 0 mm, both bottom and sidewall fusions revealed opposite trends with oscillation frequency increase, by slightly increasing bottom penetration and slightly decreasing sidewall penetration. At -1 mm proximity, there is no clear trend in sidewall and bottom penetration. However, proximity to the sidewall shows a stronger influence in weld metal penetration than oscillation frequency, with an increase in sidewall penetration with closer proximity, and a strong decrease in bottom penetration at -1 mm proximity.

5.3.2.2 Sidewall proximity influence for single sidewall trials

From both spreadsheets described earlier, the data was grouped by trials with the same torch to sidewall proximity and torch position. It is important to note that it is the wire proximity to sidewall that was measured in the experiment and not the torch. The proximity was considered to be the distance formed by the corner of the workpiece bottom with the sidewall and by the outer edge of the wire (sidewall side) (Figure 3.13 – p. 58). Both torch centre positions (CRight and CLeft) were merged by averaging and voltage from torch at left (opposite to the sidewall) was discarded. The data was then combined and the graph of Figure 5.21 was plotted. The colour scheme and representation of this graph is similar to the previous graph.

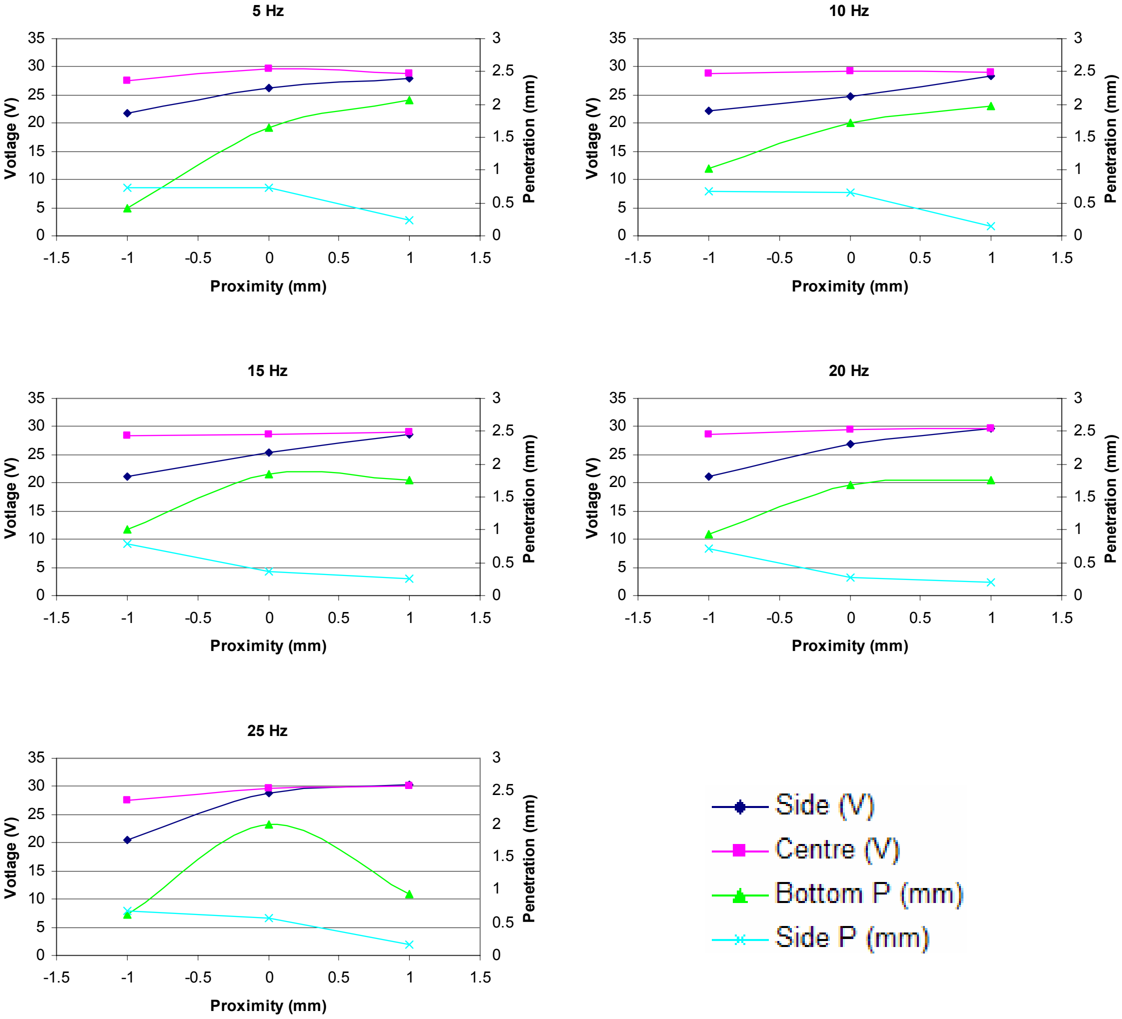


Figure 5.21 - Proximity influence in peak voltages and weld metal penetrations in single sidewall trials, at different torch oscillation frequencies

Figure 5.21 show the graphs of torch to sidewall proximity influence on peak voltages and weld metal penetrations, at different torch oscillation frequencies. It is apparent from the graphs that all show similar behaviour, revealing a clear relationship between peak voltages and fusion values depending on the sidewall proximity. Higher peak voltage difference is obtained between Centre and Side voltages with negative proximities almost no difference 1 mm away from the sidewall. However, at 0 mm proximity this difference is also influenced by oscillation frequency with a slight increase from 5 Hz to 10 Hz followed by a decrease after 15 Hz.

Bottom weld metal penetration visibly decreases with sidewall proximity and the opposite occurs with sidewall weld metal penetration.

5.3.2.3 CTWD influence for single sidewall trials

This analysis used only trial D4 data from both spreadsheets and the result was plotted on the graph shown in Figure 5.22. The colour scheme and representation for this graph is similar to the previous graph. Linear regressions were used for voltages and polynomial regressions for weld metal penetrations.

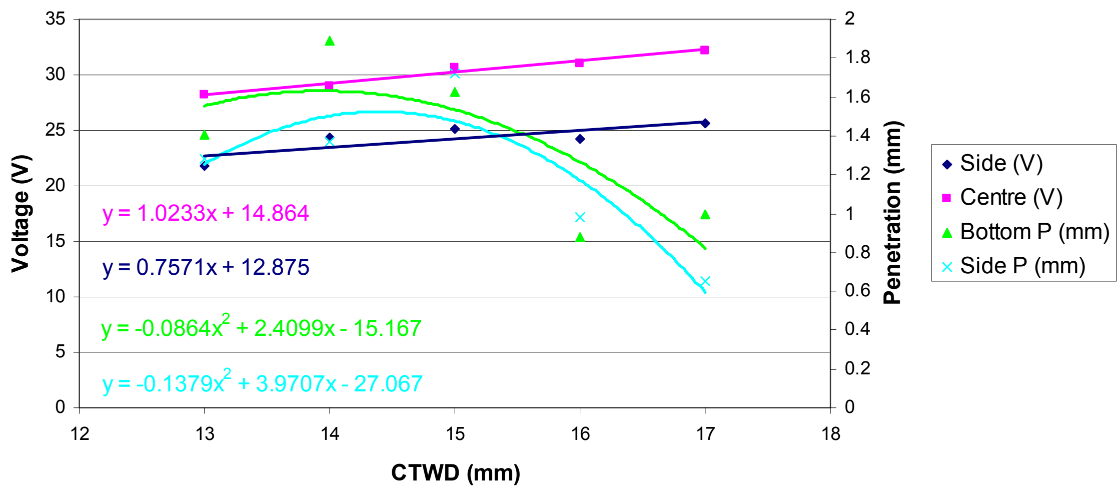


Figure 5.22 - CTWD influence in the arc peak voltage signals and in the bottom and sidewall weld metal penetrations in single sidewall trials (5 Hz oscillation frequency, 0 mm proximity)

The graph of Figure 5.22 shows a similar increasing trend of Centre and Side voltages. Voltage sensitivity in the difference between Centre and Side voltages is not visibly influenced by CTWD changes. Weld metal penetrations on the other hand are visibly affected by CTWD values showing optimum values for sidewall weld metal penetration at 15 mm CTWD and bottom weld metal penetration at 14 mm CTWD.

5.3.2.4 Arc behaviour analysis for CTWD variations, single sidewall trials

It was apparent from the trials conducted that there is a difference in the Centre voltage depending on whether the torch approaches the centre position from the left or from the right. An analysis was performed to study this voltage difference when varying CTWD. The data was extracted using WeldData and compiled in a spreadsheet. For each different CTWD trial, six measurements were taken from the arc high speed images, in positions close to centre point. Each measurement consists of extracting the difference in arc lengths and voltages from two points of the half torch oscillation cycle. Figure 5.23 shows the points in one torch weaving cycle where the measurements were taken, between Point A and Point B.

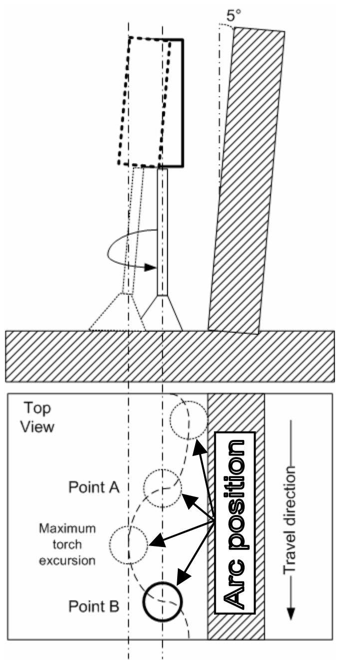


Figure 5.23 – Torch oscillation positions for measuring wire tip relative position

Points A and B have no specific distance from the sidewall and the criteria was to have the arc completely established vertically to the bottom workpiece. Time between the two sampled points was also measured and arc length difference was obtained by from WeldData using horizontal support lines distance difference as shown in Figure 5.24.

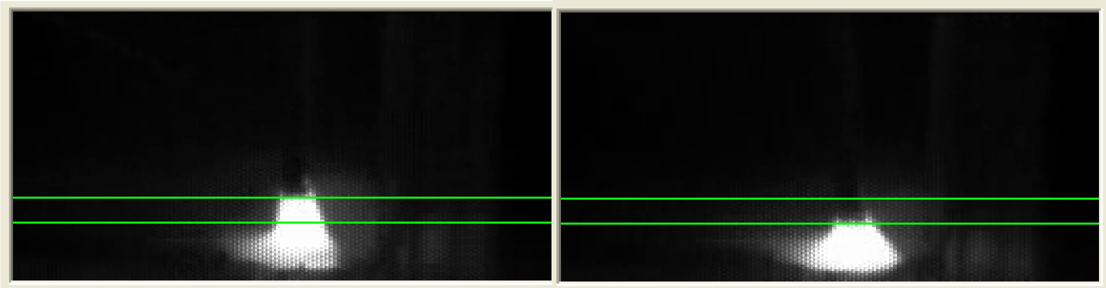


Figure 5.24 – WeldData horizontal support lines positioned at the wire tip - Point A (left) and Point B (right)

The extracted information was used to perform the following calculations: the measured time and extracted voltage of each measurement was divided by the respective arc length difference to obtain the time and voltage values per millimetre. The resultant values were then averaged for each CTWD and shown in the graph of Figure 5.25. This graph shows the average time the arc takes to change its length and respective voltage, for different CTWD values and per unit of length.

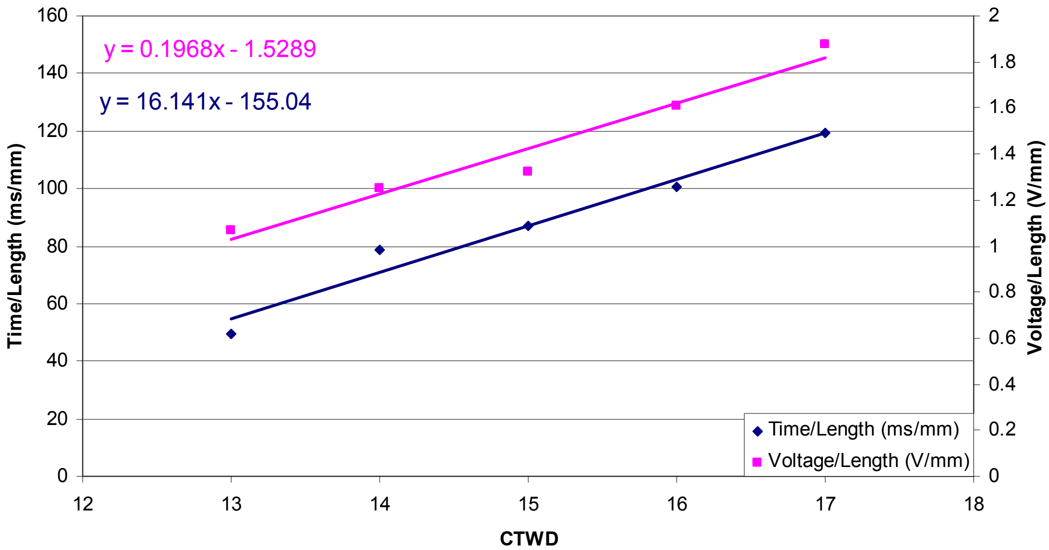


Figure 5.25 – Arc recovery time per millimetre and voltage per millimetre for CTWD values of trial D4 (5 Hz torch oscillation frequency)

The graph of Figure 5.25 shows a clear trend relationship between time and voltage changes with CTWD. Slope values from the linear regressions will be used in the discussion chapter, section 6.4.2.6.

5.3.2.5 Arc signal sensitivity analysis for single sidewall trials

From the conducted trials, peak voltages at four torch positions were extracted. The torch positions were extreme left, extreme right, centre coming from left and centre coming from right positions. In other words, when the torch position signal from the LVDT crosses one of the four positions, the last peak voltage at that position is measured. The measured voltages were exported to a spreadsheet and averaged to obtain a representative voltage value for each torch position in each trial. The graphs of Figure 5.26 were then plotted. This analysis is similar to the performed in section 5.3.1.

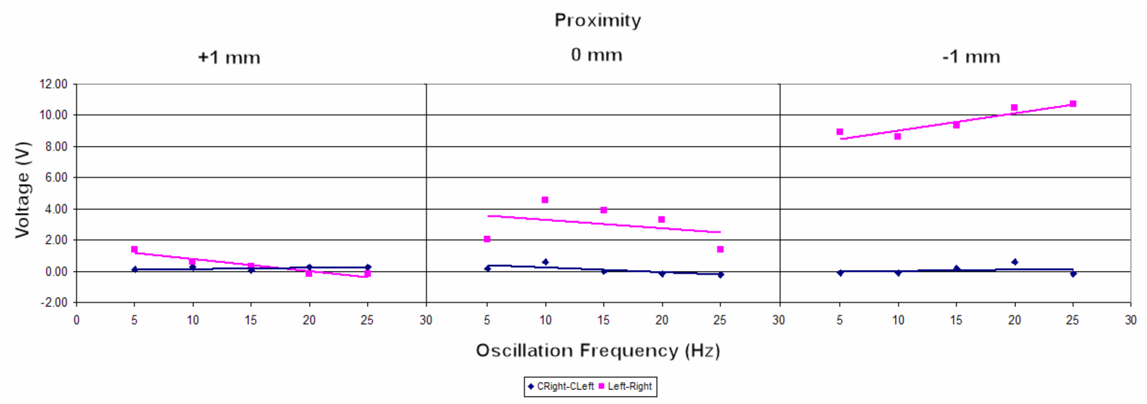


Figure 5.26 – Peak voltage difference (Left-Right and CRight-CLeft) as a function of oscillation frequency and for different sidewall proximities

Clearly from the graphs of Figure 5.26, Left-Right peak voltage difference is more visible with changing sidewall proximity than with oscillation frequency changes. It is apparent though that for +1 mm and 0 mm this voltage difference shows a small decreasing trend whereas at -1 mm it shows a visible increase with oscillation frequency increase. There is no apparent effect of either oscillation frequency or sidewall proximity on CRight-CLeft voltage.

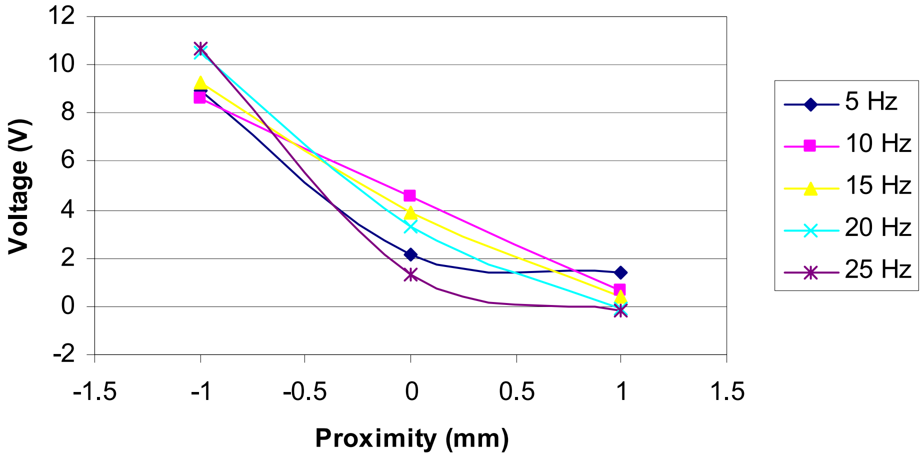


Figure 5.27 – Left-Right voltages versus sidewall proximity for different oscillation frequencies

The graph of Figure 5.27 was obtained with the values used in the previous graph of Figure 5.26 but plotted against sidewall proximity. It shows the clear descending trend of Left-Right voltages as the torch moves away from the sidewall. At 0 mm proximity, Left-Right voltage values show more influence of oscillation frequency than for -1 mm and +1 mm proximities.

For CTWD variation trials, the same analysis for the four torch positions was performed. Figure 5.28 shows the results. It is apparent from the graph an increasing trend of Left-Right voltages with an increase of CTWD. On the other hand CRight-CLeft shows little change with CTWD variations.

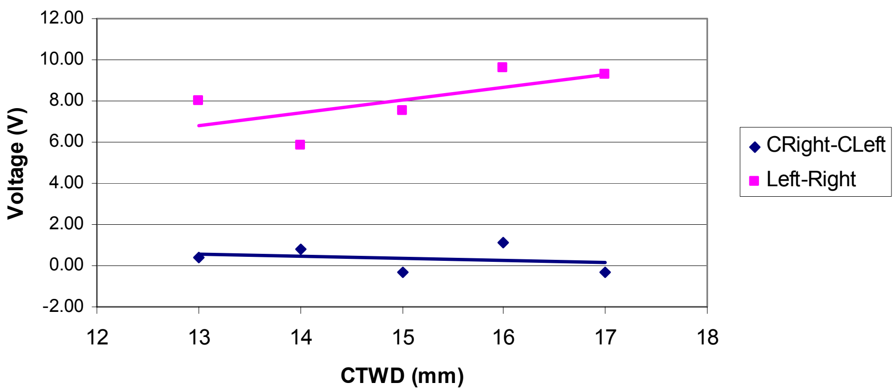


Figure 5.28 - Peak voltage variation versus CTWD for trial D4, 5 Hz oscillation frequency

5.3.3 Double sidewall (groove) trials with 5° preparation angle

Trials E1 to E8 were performed with a double 5° sidewall forming a groove (Figure 3.14 – p. 61) to evaluate arc behaviour for arc signal sensitivity analysis and weld metal penetration, using GMAW-P. Trials were conducted with variations in torch proximity to the sidewall and in torch oscillation frequency. Arc signal analysis, high speed image visualisation and metallographic images were used to extract the information demonstrated in this section. The objective of this study is to determine what influence torch oscillation frequency and proximity to the sidewall have on weld metal penetration and how this is reflected in peak voltages. Due to their asymmetrical values, the off-centre trials E1, E2 and E8 were not considered for this analysis. Only trials E3 to E7 were used.

The analysis performed for groove trials was similar to single sidewall trials and used the same automated procedure from the WeldData program. First, peak voltages were extracted at the four torch positions: left extreme, right extreme, centre coming from right and centre coming from left. This data along with fusion measurements were put into a spreadsheet and graphs were plotted. Due to the symmetrical sidewall configuration of the groove, left and right voltages were average as well as both centre voltages. Left and right weld metal penetrations were also averaged. Data in the spreadsheet was then arranged in two ways to examine the values as a function of oscillation frequency and as a function of sidewall proximity.

5.3.3.1 Oscillation frequency influence for groove trials

Figure 5.29 and Figure 5.30 show the influence of oscillation frequency on peak voltages and weld metal penetration. The graph shows the final values represented by coloured symbols and lines. The dark blue colour represents voltage with the torch at sidewall (Side) and the pink colour represents voltage with the torch at centre of oscillation (Centre). The green colour represents bottom weld metal penetration (Bottom P) and the light blue colour represents sidewall weld metal penetration (Side P).

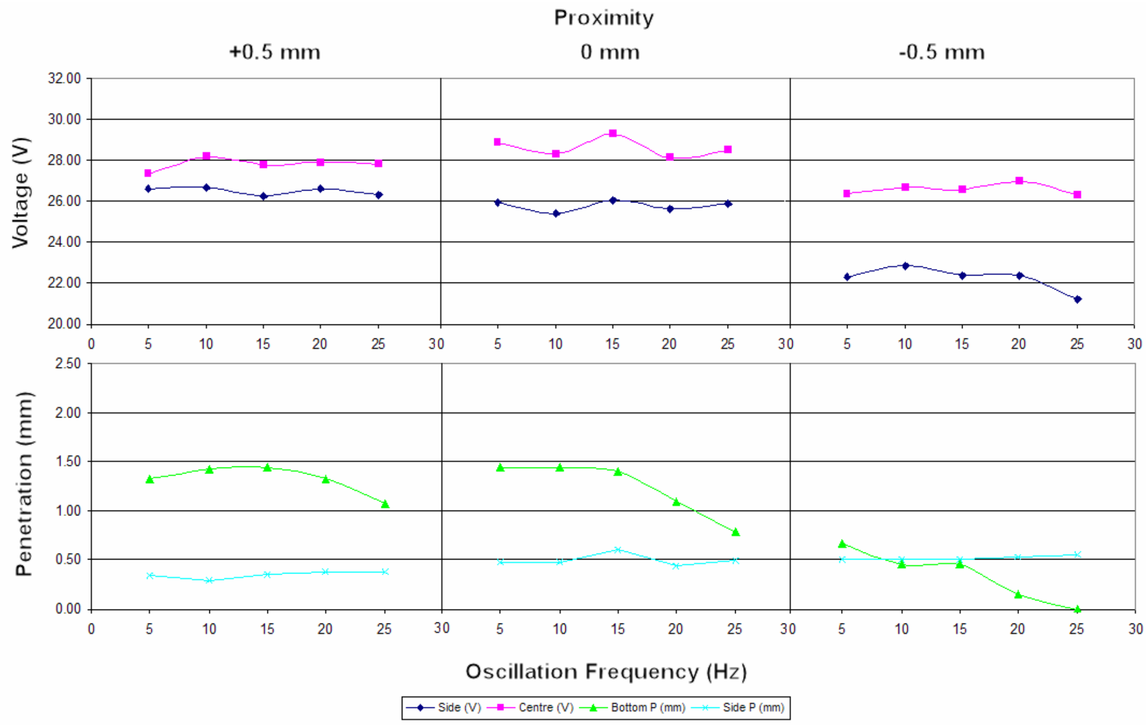


Figure 5.29 - Oscillation frequency influence on peak voltages and weld metal penetration for groove trials with 2.5 mm oscillation width, at different torch to sidewall proximities

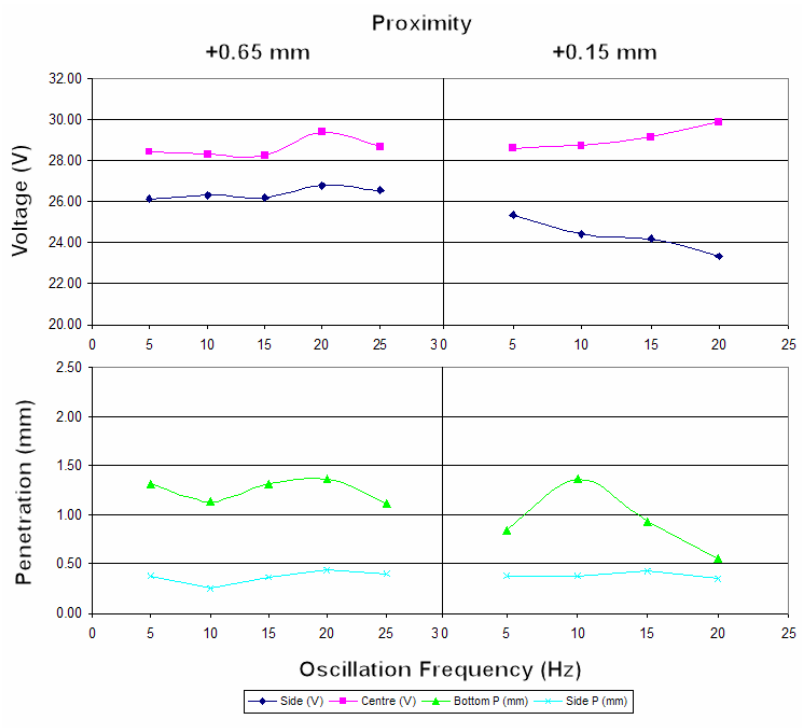


Figure 5.30 - Oscillation frequency influence on peak voltages and weld metal penetration for groove trials with 3.7 mm oscillation width, at different torch to sidewall proximities

From the graphs of Figure 5.29 and Figure 5.30 it is clear that the influence of oscillation frequency on peak voltages is small. The same cannot be said though in fusion values where bottom penetration is clearly influenced by oscillation frequency increase at close proximity to the sidewall. Peak voltage difference between Centre and Side torch positions along with sidewall penetration also show no influence with oscillation frequency change. According to the graphs, proximity rather oscillation frequency seems to be the major influencing factor for the four variables analysed. Oscillation width variation from 2.5 mm to 3.7 mm has also produced no significant variation in peak voltages and fusion values.

5.3.3.2 Sidewall proximity influence for groove trials

Figure 5.31 shows the proximity influence on arc peak voltage signal and weld metal penetration in groove trials. The colour scheme and representation of this graph is similar as the previous graphs.

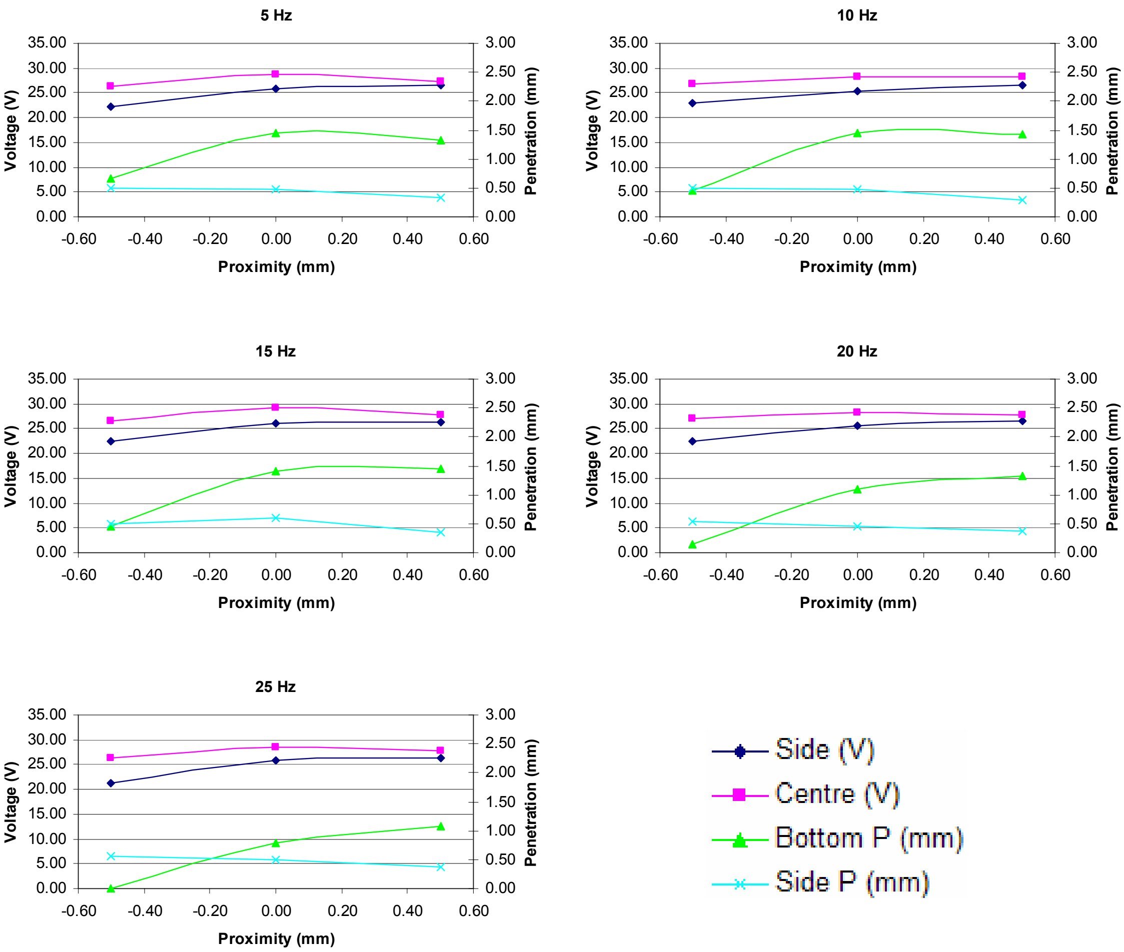


Figure 5.31 - Proximity influence on peak voltages and weld metal penetration for groove trials with 2.5 mm oscillation width, at different oscillation frequencies

The graphs of Figure 5.31 show the influence of wire to sidewall proximity on peak voltages and fusion values, for an oscillation width of 2.5 mm. It is apparent from the graphs that negative proximities affect all the four analysed variables. Voltage difference between Centre and Side positions are more evident with proximity to the sidewall, but reduce to almost zero as the torch moves away from the sidewall. Bottom penetration has a visible decrease for a negative proximity and tends to stabilise to a constant value for positive proximities. However, it is influenced by oscillation frequency of 20 Hz and over, more visible at 0 mm proximity. Sidewall penetration slightly increases with sidewall proximity moving from positive to negative values.

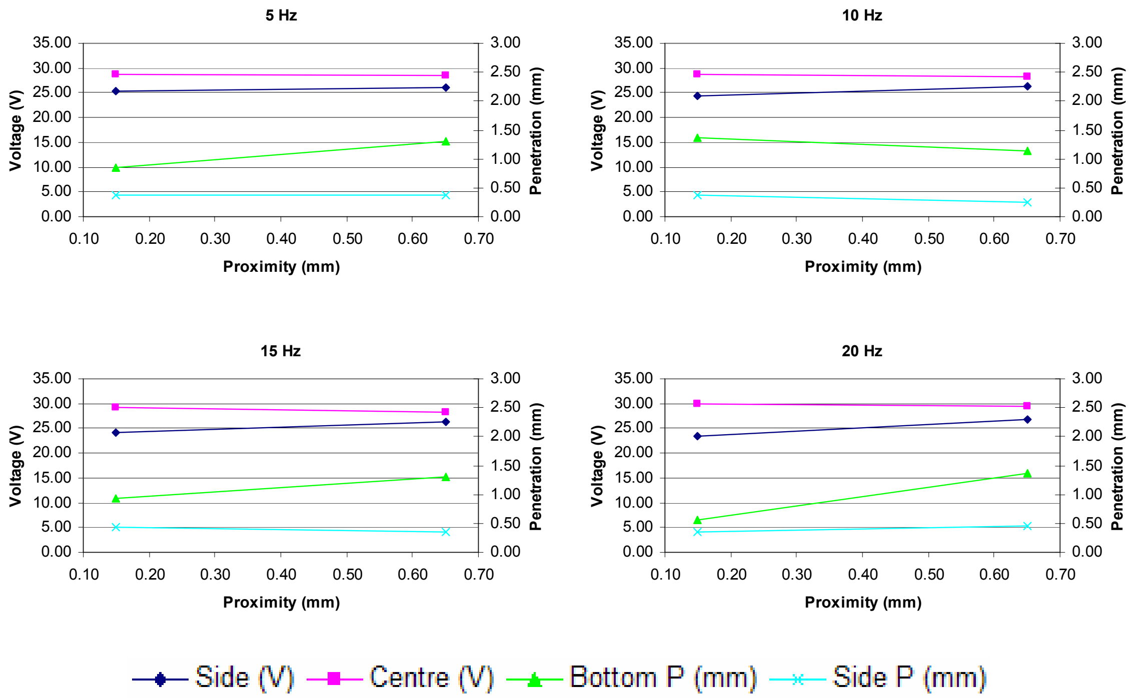


Figure 5.32 - Proximity influence on peak voltages and weld metal penetration for groove trials with 3.7 mm oscillation width, at different oscillation frequencies

The graphs of Figure 5.32 show the influence of torch to sidewall proximity on peak voltages and fusion values, for an oscillation width of 3.7 mm. As with 2.5 mm oscillation width, voltage difference between Centre and Side positions show an increase with sidewall proximity. Bottom penetration increases for positive values of proximity in most cases whereas sidewall penetration appears to be constant in the range tested, irrespective of proximity and frequency.

5.3.4 High speed video and metallographic analysis

High speed video images from the experiments were obtained (section 3.3.2) in order to analyse arc behaviour. In addition, macrosections were taken to relate the variation in experimental parameters to fusion zone geometry. This section reports the result of observations made with both arc images and macrosections.

5.3.4.1 Arc signal sensitivity comparison between CV GMAW and GMAW-P

This set of experiments (section 3.3.2.1) examined the comparison between CV GMAW and GMAW-P for the single side 45° preparation. The following figures show extracted images from the trials. The figures are divided into four arc images taken at specific torch positions as described.

Figure 5.33 shows arc images of trial C2.1. For this experiment, the arc is vertical at the longest arc length and deflects towards the plate at the shortest arc length. It should also be noted that the droplets are travelling from right to left, i.e., they are being deflected out of the arc.

Left	Centre	Centre	Right
	Coming from left	Coming from right	

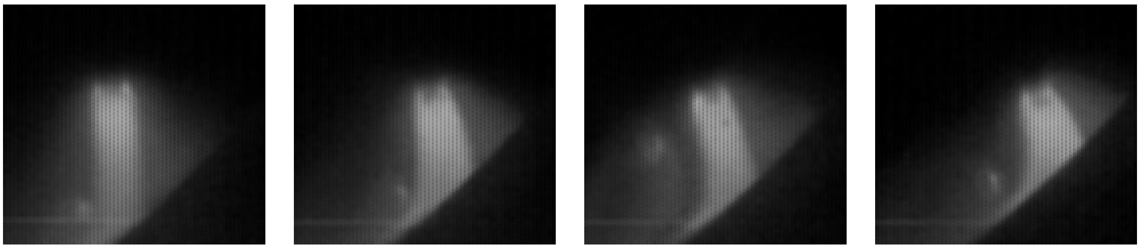


Figure 5.33 - Arc images from trial C2.1 (CTWD 15.4 – 17.6 mm @ 5 Hz; GMAW-P)

Figure 5.34 shows shorter arc lengths when compared to the arcs of Figure 5.33. CTWD in this trial was shorter than the trial of Figure 5.33. It is also possible to see that the left arc is not as vertical as the left arc of Figure 5.33.

Left	Centre	Centre	Right
	Coming from left	Coming from right	

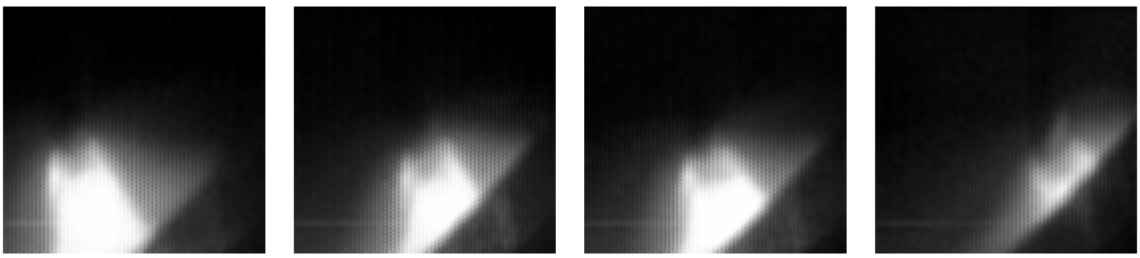


Figure 5.34 - Arc images from trial C3.5 (CTWD 13.2 – 15.7 mm @ 9 Hz; GMAW-P)

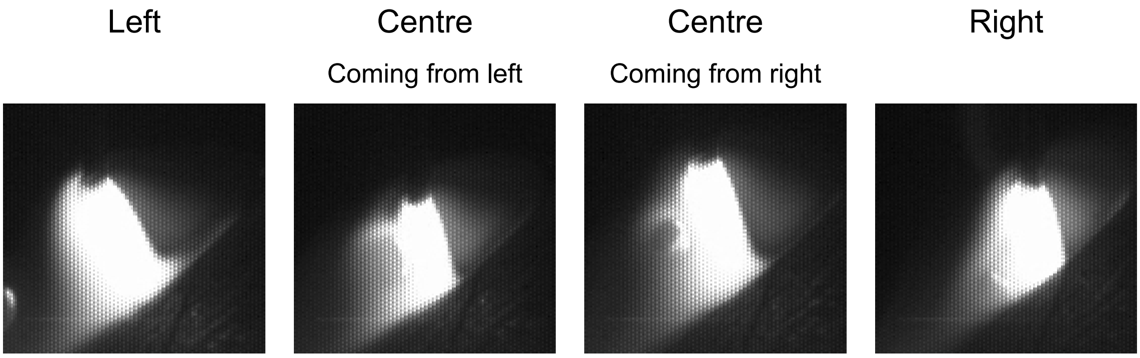


Figure 5.35 - Arc images from trial C4.1 (CTWD 12.4 – 14.7 mm @ 5 Hz; CV GMAW)

CV GMAW arc lengths are longer and show a different behaviour when compared with GMAW-P arcs. For GMAW-P, arc images were extracted from the high-speed film during the peak current period whereas for CV GMAW, at random intervals. Instability in arc position is also more visible in CV GMAW and the angle formed by the arc and the workpiece varies more than in GMAW-P. Droplets are also deflected out of the arc and are of similar size as in GMAW-P trials, i.e. approximately equal to the wire diameter. From the arc images of Figure 5.35, it is apparent that the arc at centre coming from left is shorter that at centre coming from right. The same can be seen in Figure 5.36 in another CV GMAW trial.

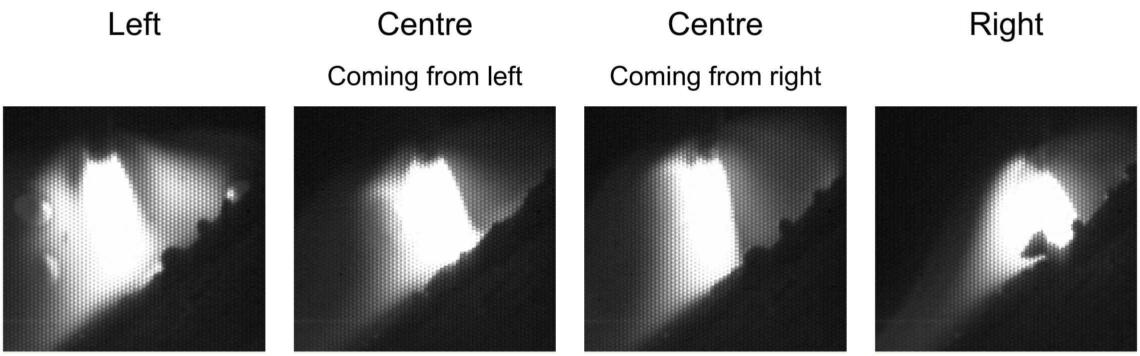


Figure 5.36 - Arc images from trial C5.3 (CTWD 12.9 – 15.3 mm @ 9 Hz; CV GMAW)

Figure 5.37 shows two image sequences of droplet detachment of the same trial. The top sequence is with the torch at extreme left and the bottom sequence is with the torch at extreme right.

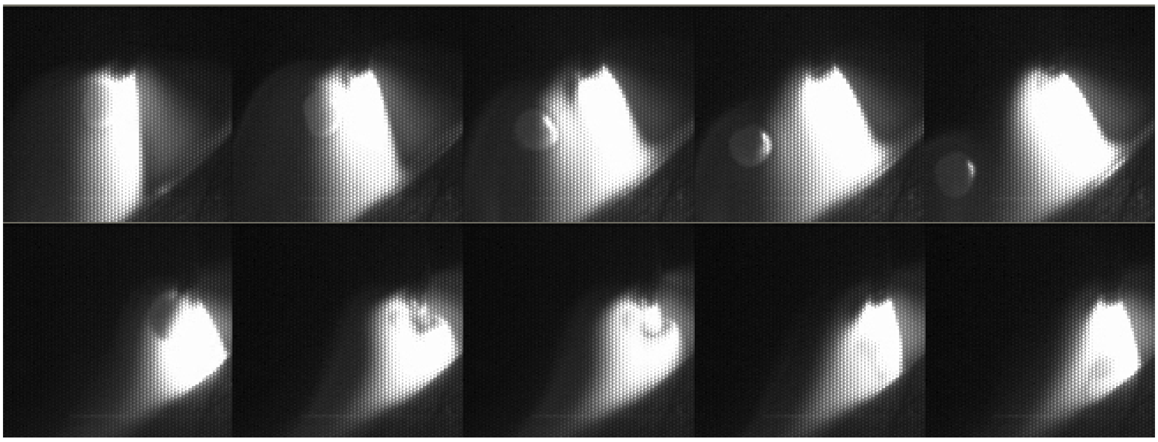


Figure 5.37 – Two droplet detachment sequences from trial C4.1 (CTWD 12.4 – 14.7 mm @ 5 Hz; CV GMAW)

It appears in the top sequence that with the longer arc length, the droplet of diameter approximately equal to the wire diameter is deflected out of the arc with almost no movement of the wire. On the other hand, the shorter arc length shows the droplet travelling within the arc column. This behaviour was observed in both CV GMAW and GMAW-P trials.

In conclusion, from the single side 45° preparation trials, the main observations were:

a) Vertical arc at left hand side for long arcs
b) Arc deflected to the plate for short arcs
c) Droplets deflected out of the arc for long arcs
d) Droplets travelled within the arc column for short arcs
e) Different arc length at Centre with arc coming from the left or from the right in CV GMAW

5.3.4.2 Single sidewall trials with 5° preparation angle

Figure 5.38 to Figure 5.41 show arc images taken at peak current from trials D1 to D4. Figure 5.43 to Figure 5.46 show macrosections from the same trials. Arc images were also sketched to better visualise arc behaviour near the sidewall. Each arc image was extracted with the torch position described in the top of the image. The columns in each figure were arranged in order to follow the natural torch weaving path.

Figure 5.38 shows arc images of trial D1.2 with 0 mm wire proximity to the sidewall. At the right position, the arc is fully established on the sidewall. At the centre, in both cases, vertical arc establishment is visible but with spreading on the sidewall. With the wire proximity of 1 mm (Figure 5.39), the arc spreading in the wall direction is only slightly visible with the torch at the right position. The opposite occurs in Figure 5.40 (wire proximity of -1 mm) with the arc almost constantly established on the sidewall.

It is clear in Figure 5.39 (1 mm proximity) that arc length is shorter than in Figure 5.38 (0 mm proximity) and in turn shorter than in Figure 5.40 (-1 mm proximity). This shows a clear relationship between wire proximity to the sidewall and arc length. In Figure 5.40 with the torch at the right, the wire touches the sidewall and the arc is almost extinguished.

Right	Centre	Left	Centre
	Coming from right		Coming from left

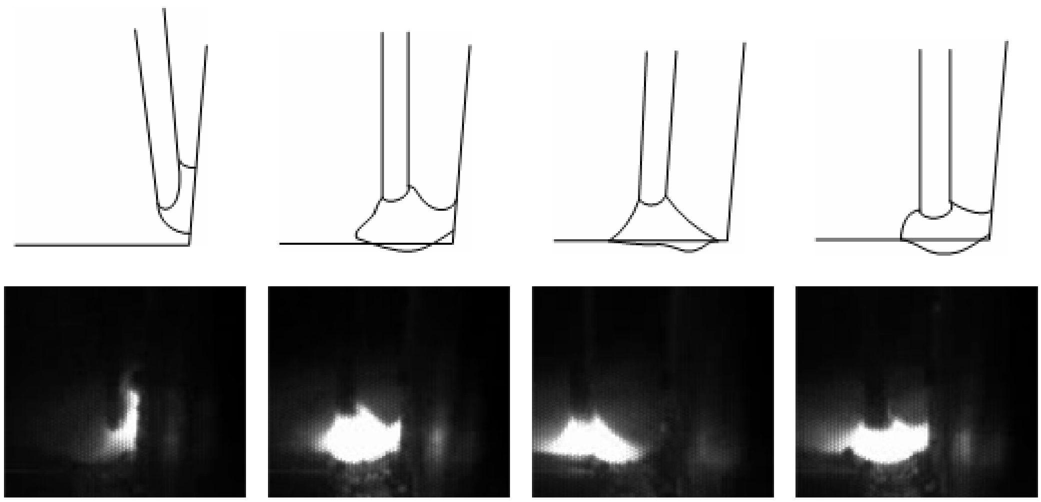


Figure 5.38 - Arc images from trial D1.2 (Osc. freq. 10 Hz – Proximity 0 mm)

Right	Centre	Left	Centre
	Coming from right		Coming from left

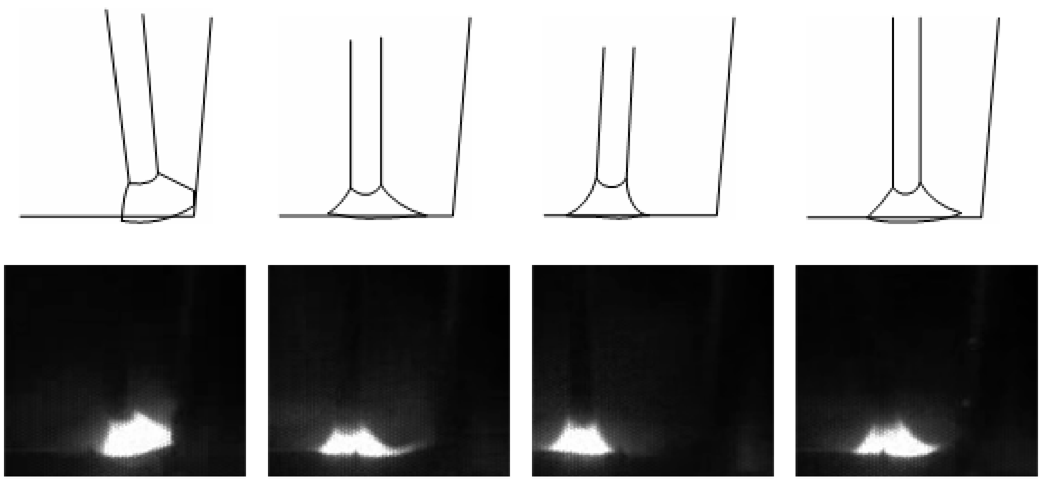


Figure 5.39 - Arc images from trial D2.2 (Osc. freq. 10 Hz – Proximity 1 mm)

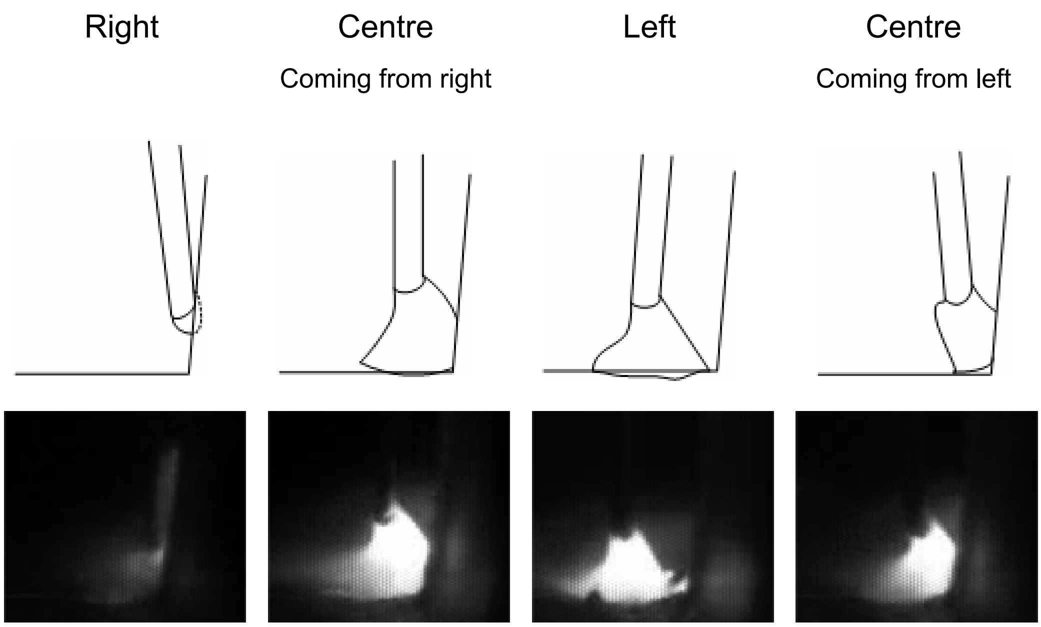


Figure 5.40 - Arc images from trial D3.2 (Osc. freq. 10 Hz – Proximity -1 mm)

Figure 5.41 shows arc images from trials D4.1 to D4.5 (CTWD variation trials). The images were extracted with the torch at maximum extreme in the left (away from the sidewall). It is apparent from arc images that arc length increases with the increase of CTWD. The sidewall proximity for trial D4 was 0 mm.

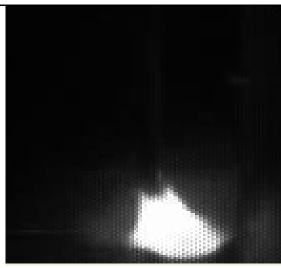
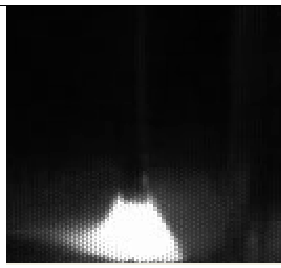
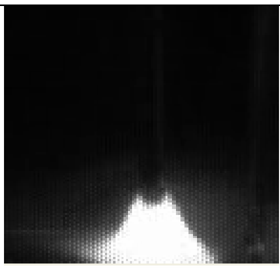
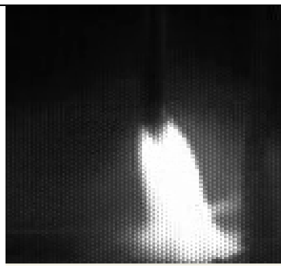
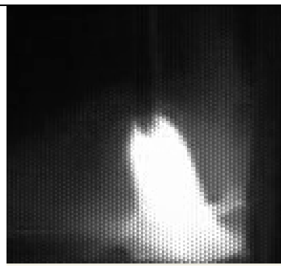
Trial D4 Proximity: 0 mm Osc. Freq. 5 Hz Trial D4.x : CTWD	 Trial D4.1 : 13 mm	 Trial D4.2 : 14 mm
 Trial D4.3 : 15 mm	 Trial D4.4 : 16 mm	 Trial D4.5 : 17 mm

Figure 5.41 - Arc images from trial D4 with the torch at maximum left extreme (Proximity 0 mm – Oscillation frequency 5 Hz)

Figure 5.43 to Figure 5.46 show bead profiles (macrosections) of trials D1 to D4. It should be noted that the torch centre of oscillation was vertical (perpendicular to bottom plate) and the right sidewall performs a 5° angle with the vertical. The angle of the beads was measured to relate to torch proximity to the sidewall and to oscillation frequency. The bead angle was measured as shown Figure 5.42. Bead angles show a gradually decrease with increase in oscillation frequency from 30° for trial D1.1 to 25° for trial D1.5. Also, undercut defects become more visible with oscillation frequency increase.

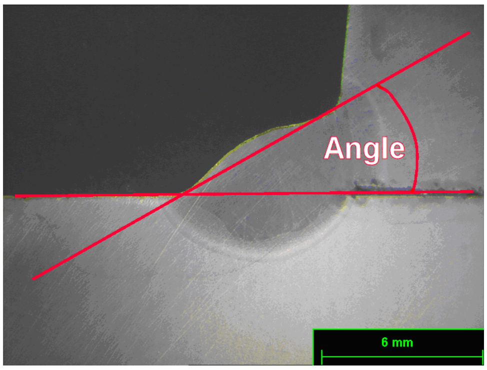


Figure 5.42 – Bead shape angle for macrosections of trial D1 to D4

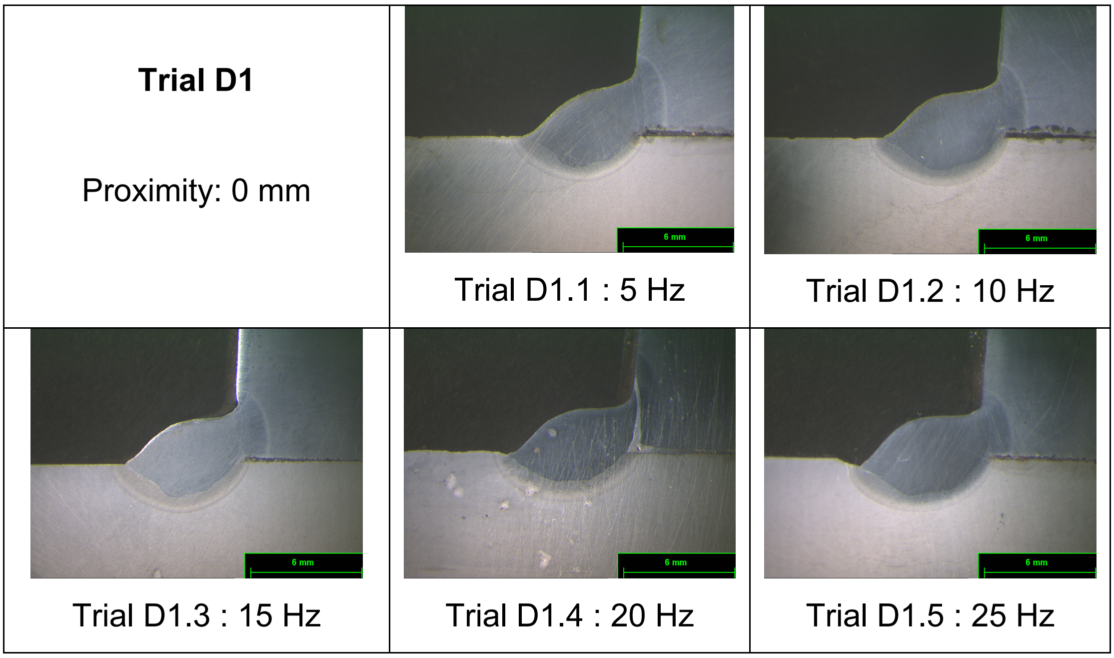


Figure 5.43 - Bead profile shape from trial D1 at each torch oscillation frequency

Figure 5.44 shows images of trial D2 bead profiles. Again, bead angles decrease with oscillation frequency increase between 28° for trial D2.1 and 20° for trial D2.5. Undercut defects are also more visible with higher oscillation frequencies. Trial D2.5 shows a visibly shallow bottom workpiece weld metal penetration.

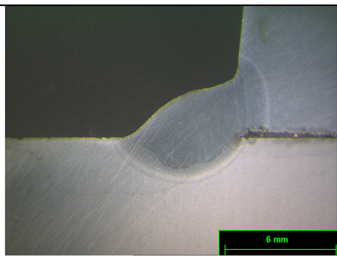
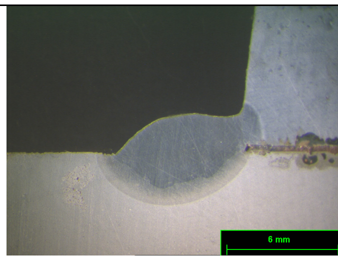
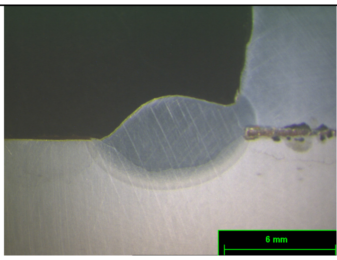
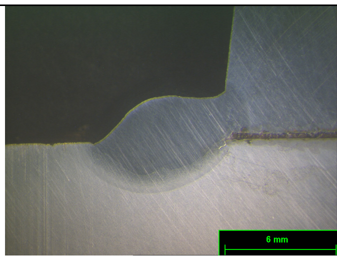
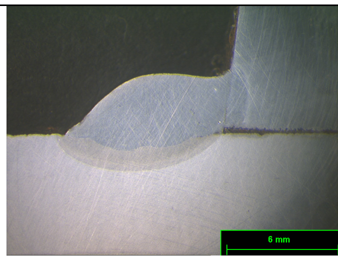
Trial D2 Proximity: +1 mm	 Trial D2.1 : 5 Hz	 Trial D2.2 : 10 Hz
 Trial D2.3 : 15 Hz	 Trial D2.4 : 20 Hz	 Trial D2.5 : 25 Hz

Figure 5.44 - Bead profile shape from trial D2 of at each torch oscillation frequency

Figure 5.45 shows images of trial D3 bead profiles. Bead angles decrease with oscillation frequency between 44° (trial D3.1) to 33° (trial D3.5). Undercut defects are visible in all trials but become stronger with the increase of oscillation frequency. There is a possible lack-of-sidewall fusion in the corner between sidewall and bottom workpiece in some specimens, although it is not possible to confirm because of the gap formed by the sidewall plate and the bottom plate. Trials D3.1 and D3.5 show shallow bottom workpiece weld metal penetration.

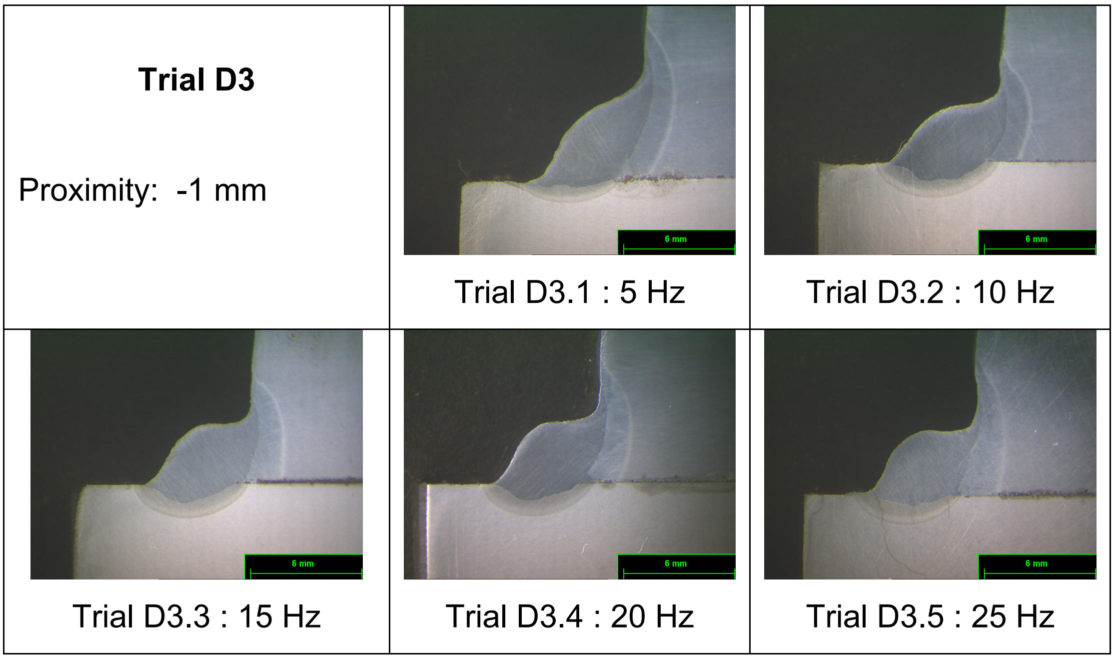


Figure 5.45 - Bead profile shape from trial D3 at each torch oscillation frequency

Figure 5.46 shows images of trial D4 bead profiles where CTWD was varied with a fixed oscillation frequency and sidewall proximity. Bead angles increased with CTWD from 33° at 13 mm (trial D4.1) to 40° at 17 mm (trial D4.5). A small undercut defect is visible only in trial D4.3. Bottom and sidewall penetration becomes shallower with increase of CTWD.

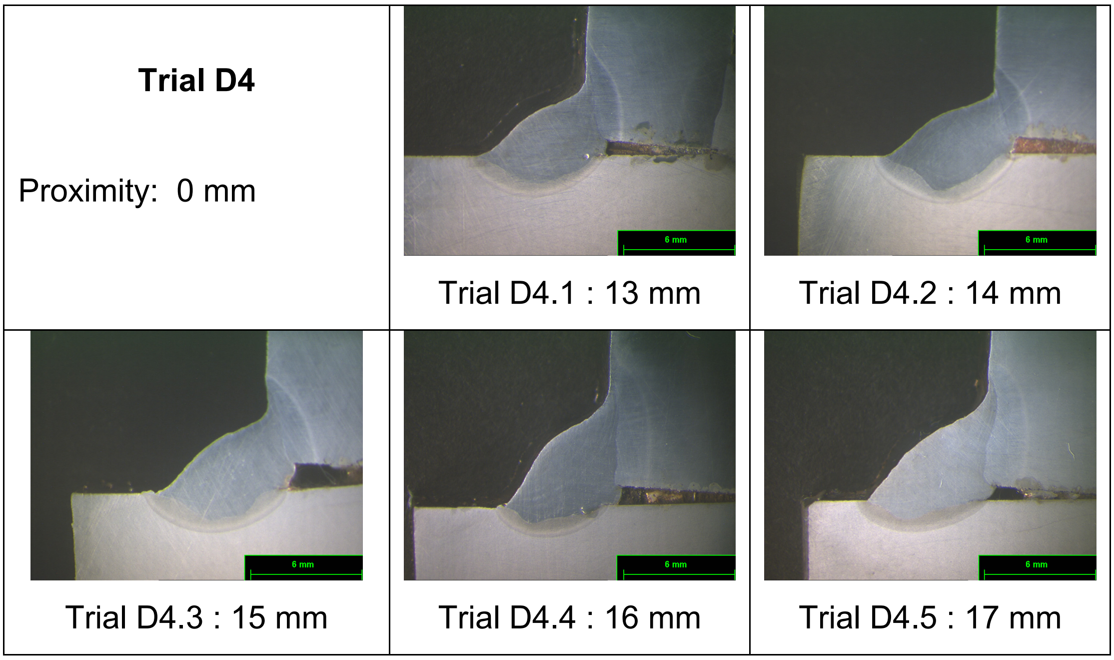


Figure 5.46 - Bead profile shape from trial D4 at each CTWD

In conclusion, from the single 5° sidewall trials, the main observations were:

a) Arc length increases with distance from the sidewall and with increase of CTWD
b) Bead angles are higher with closer sidewall proximity, and undercut defects also increase with closer sidewall proximity
c) Increased oscillation frequency results in increased undercut defects and shallower bottom penetration
d) Closer sidewall proximity increases sidewall penetration

5.3.4.3 Double sidewall (groove) trials with 5° preparation angle

Observation of arc images extracted from the high speed video and bead profiles of groove trials were also performed. This section describes important features found in the observations. Figure 5.47 shows arc images and bead profile of off-centred trial E1 and Figure 5.49 to Figure 5.51 show arc images of trials E2 to E8. Metallographic images are shown in Figure 5.53 to Figure 5.59, grouped and analysed by proximity for trials E1 to E8.

Figure 5.47 shows two arc images taken at centre and right side of torch oscillation and bead profile of trial E1.4. They are representative of the whole trial E1. The top arc image shows an arc established on both sidewalls not reaching the root plate. Although the arc is vertical, it is visible from the image that arc length is not equal to the distance from the wire tip to the bottom of the groove. The droplets travelled to the bottom of the groove. Also, the bottom arc image from the figure shows the arc established directly on the sidewall, with the torch in the extreme right position. At the same height from the groove bottom, the bead profile on the right image shows an undercut defect.

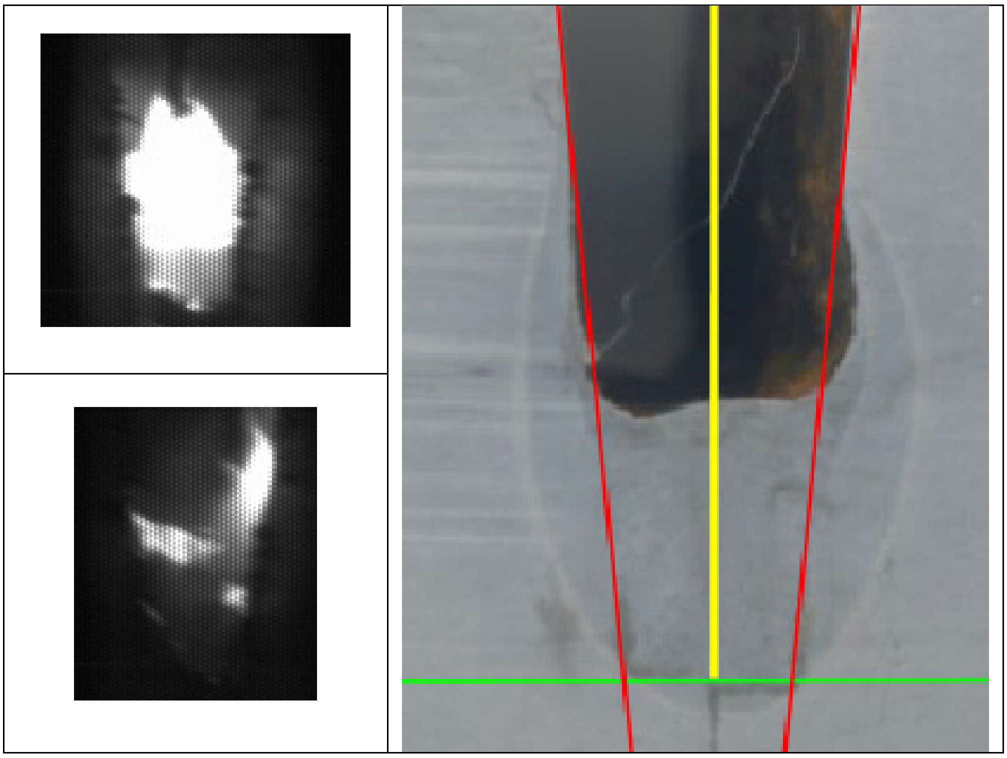


Figure 5.47 - Trial E1.4 arc images and bead profile – proximity -0.5 mm – 20 Hz oscillation frequency - 0.3 mm off-centre to the left

Figure 5.48 shows arc images from trials E2.1 and E2.4. Due to the torch off-centre path, it is not possible to show the left torch position arc image. The wire collided with the sidewall creating a short circuit (dip transfer) and no visible arc was formed, resulting in a black image. With the increase in oscillation frequency, arc length increased in the right side of oscillation as shown in the example of Figure 5.48 at 20 Hz.

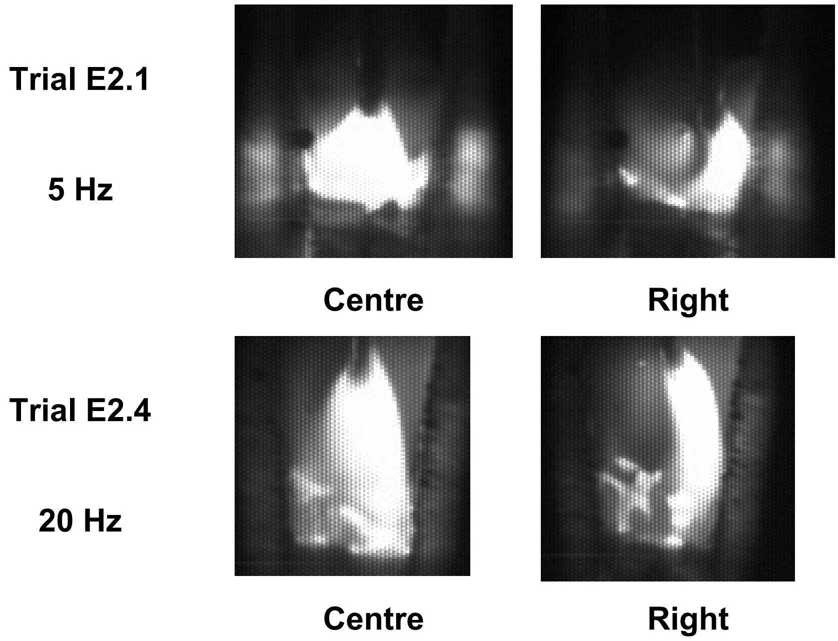


Figure 5.48 - Trials E2.1 and E2.4 arc images at each indicated torch positions– proximity 0 mm – 0.6 mm off-centre to the left

A common observation was found in all trials with closer proximity to the sidewall during torch weaving, although more visible with longer arcs. The arc establishes on the sidewall and then performs a descendant path to the bottom workpiece, during the rise and fall of a current pulse, as shown in the image sequence of Figure 5.49. The sequence is composed of 16 consecutive images (3.2 ms) of trial E6.1, during the period of a current pulse with the torch at maximum right extreme. It shows the arc behaviour starting on the sidewall and going down to the bottom workpiece. The droplet was released from the wire in the beginning of the background period, i.e. in the end of the image sequence.

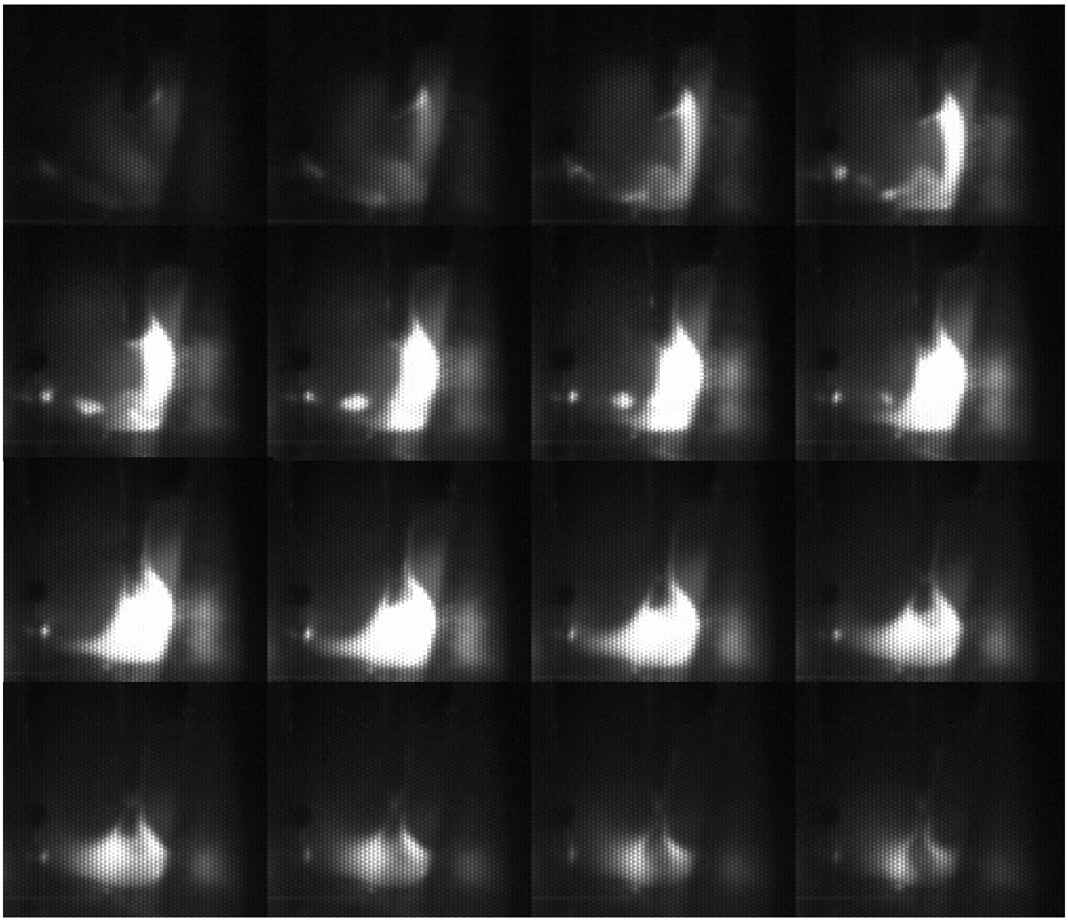


Figure 5.49 - Arc image sequence of one current pulse of trial E2.1 - proximity 0 mm, oscillation frequency 5 Hz, groove width 3.5 mm (0.2 ms between images)

Figure 5.50 shows arc images of trials E3.1, E4.1 and E5.1 (2.5 mm oscillation width trials) at each indicated torch position. It is visible from the images that the arc behaves differently for each sidewall proximity. In trial E3.1 (+0.5 mm sidewall proximity), the arc at maximum torch excursion establishes on the sidewall/bottom corner whilst in trials E4.1 and E5.1, the arc is established on the sidewall. At torch centre position, it is visible from the arc images that arc length is different on each trial. The arc length is longer with a closer proximity and shorter with longer proximity. The arc images shown in Figure 5.50 reflect the trend found in each torch oscillation frequency trial for the same sidewall proximity, and hence arc images from other trials are not presented here. In summary, side contact creates wire burn-back and consequently longer arc lengths.

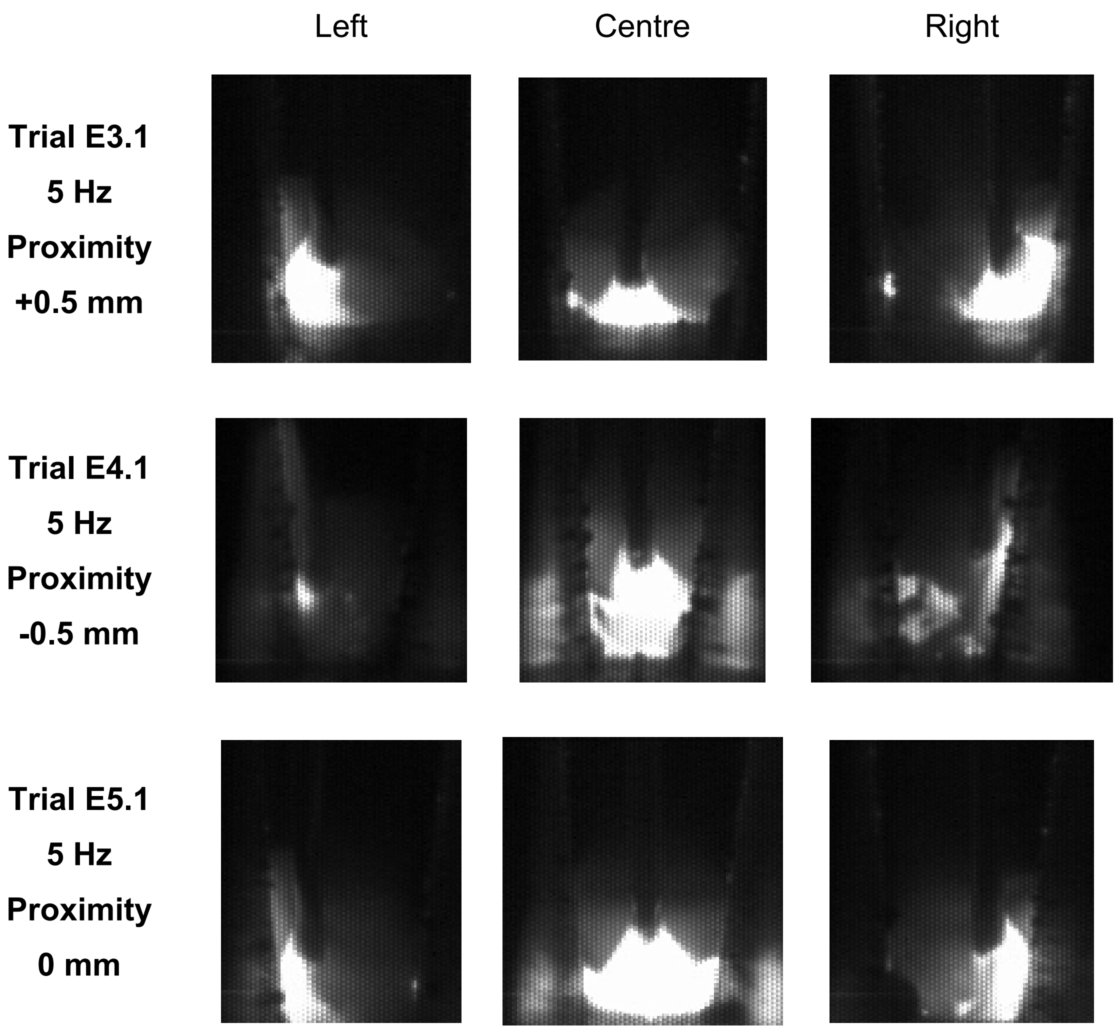


Figure 5.50 - Arc images of trials E3.1, E4.1 and E5.1, oscillation width 2.5 mm, at each indicated torch position

Figure 5.51 shows arc images from trials E6.1, E7.1, E7.4 and E8.1 (3.7 mm oscillation width trials), at each indicated torch position. As in the previous arc images for 2.5 mm oscillation width trials, different sidewall proximities create different arc behaviours in similar ways. Trial E7 (+0.65 mm proximity) also shows a visible arc length influence at maximum torch excursions with torch oscillation frequency (E7.1 and E7.4 of Figure 5.51), although with similar arc length at centre of oscillation.

Table 5.1 – Arc images from trials E6.1, E7.1, E7.4 and E8.1, oscillation width 3.7 mm, at each indicated torch position

Trial E6.1
5 Hz
Proximity
+0.15 mm

Trial E7.1
5 Hz
Proximity
+0.65 mm

Trial E7.4
20 Hz
Proximity
+0.65 mm

Trial E8.1
5 Hz
Proximity
+0.55 mm

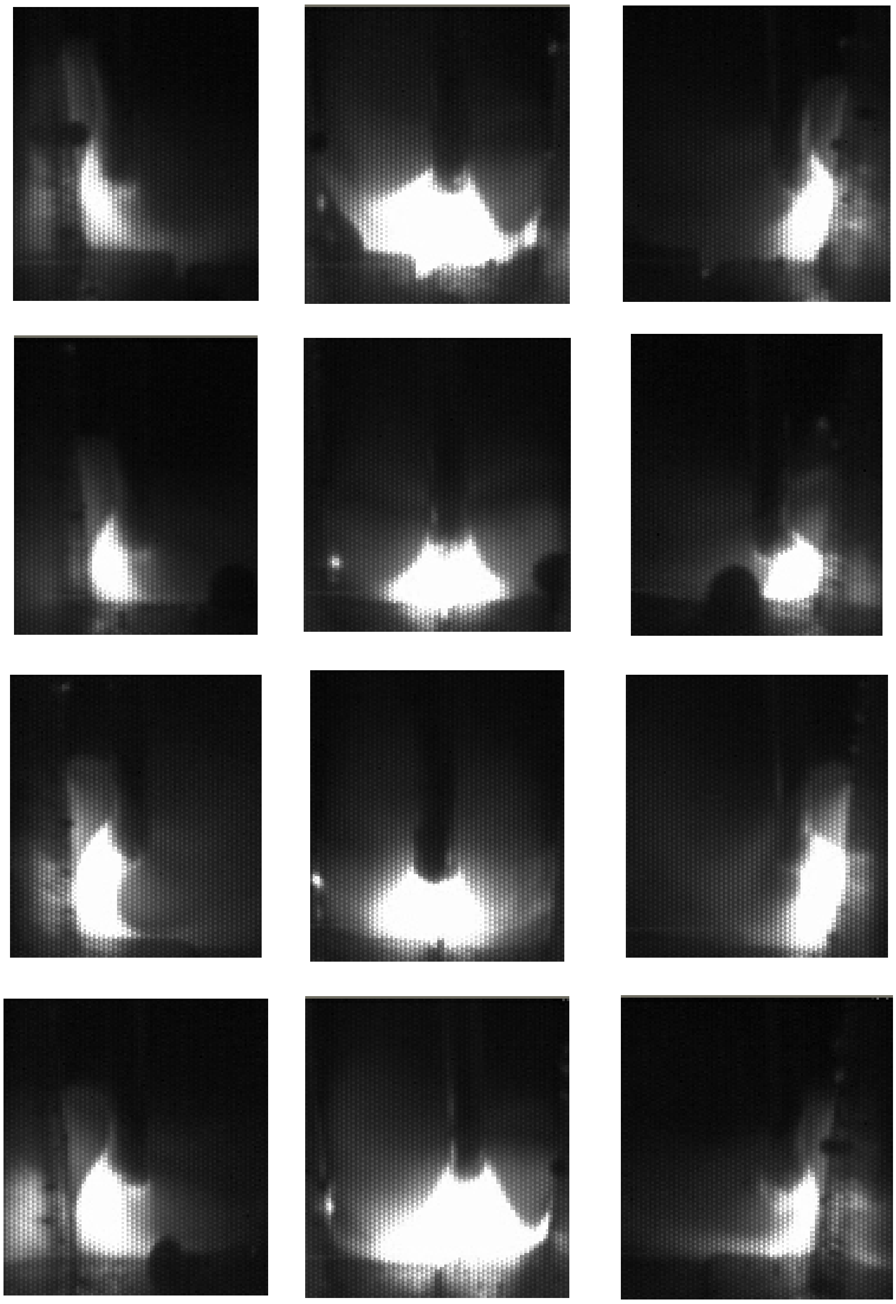


Figure 5.51 - Arc images from trials E6.1, E7.1, E7.4 and E8.1 (0.5 mm off-centre to the right), oscillation width 3.7 mm, at each indicated torch position

From Figure 5.50 and Figure 5.51, it is clear that the arc establishes to the sidewall for sidewall proximities equal and below 0.15 mm, and establishes to the sidewall/bottom corner for proximities over 0.15mm.

In conclusion from arc image observations of groove trials, it can be said:

a) Oscillation width variation from 2.5 mm to 3.7 mm does not show any visible influence on arc behaviour
b) Negative sidewall proximity creates wire burn-back and hence increasing arc length
c) When the arc establishes on the sidewall, it moves down during rise and fall period of the current pulse
d) For sidewall proximities below +0.15 mm the arc establishes on the sidewall and over +0.15 mm the arc establishes on the sidewall/bottom corner
e) For positive sidewall proximities, arc length shows more influence with oscillation frequency increase by an increase in arc length

The following analyses were performed on bead profiles of trials E1 to E8. Figure 5.52 shows a graph of bead Width/Depth ratio of the on-centre trials.

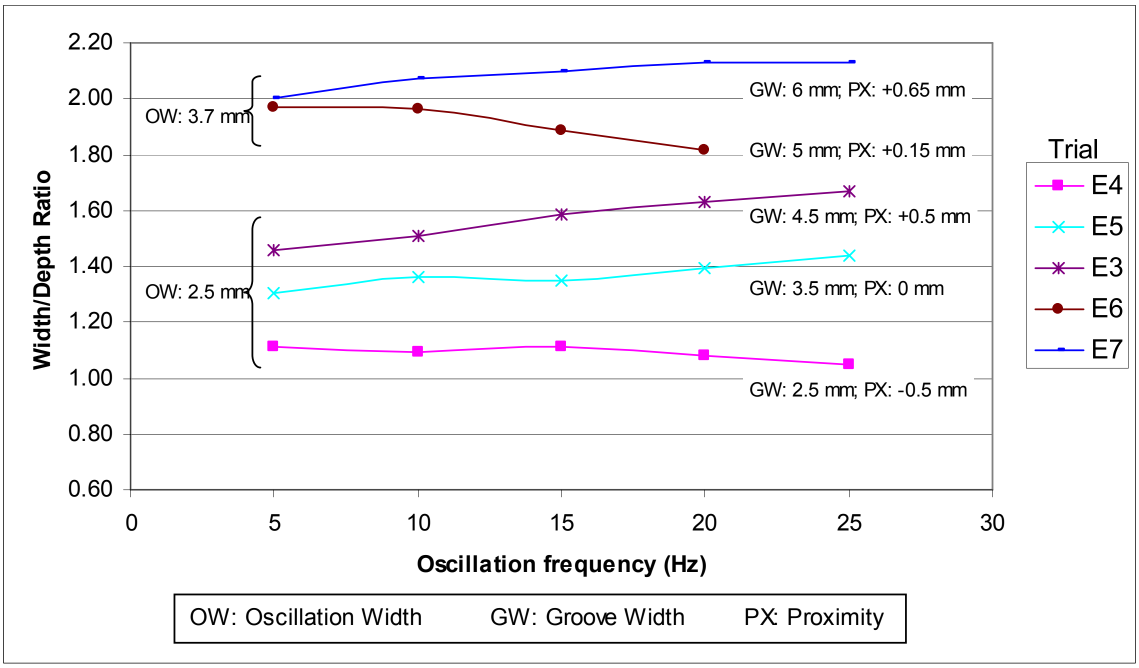


Figure 5.52 – Width/Depth ratio from bead profiles of E1 to E8 trials

It is clear from the graph of Figure 5.52 that beads become shallow with groove width increase. For each millimetre groove width increase, the ratio increases by 0.16, for the lowest oscillation frequency. An exception was found though between 4.5 mm and 5 mm groove widths. In this case, the variation was 0.5. Also, the graph shows some oscillation frequency influence on longer sidewall distances, such as in trials E3 and E7.

Figure 5.53 to Figure 5.59 contains the bead profiles of each set of trials conducted at different sidewall proximities. Each figure contains the bead profile of each trial with the oscillation frequency indicated under each image. The figures are grouped by sidewall proximity. Figure 5.53 to Figure 5.57 are from 2.5 mm oscillation width trials and Figure 5.58 to Figure 5.59 from 3.7 mm oscillation width trials.

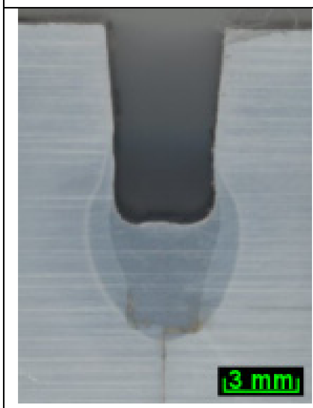
Trial E1 Proximity -0.5 mm		
	E1.1 : 5 Hz	E1.2 : 10 Hz
		
E1.3 : 15 Hz	E1.4 : 20 Hz	E1.5 : 25 Hz

Figure 5.53 - Bead profiles from trials E1 – sidewall proximity -0.5 mm – torch oscillation width 2.5 mm (0.3 mm off-centre to the left)

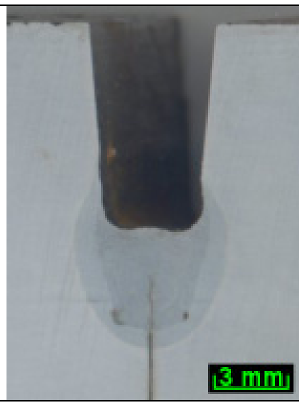
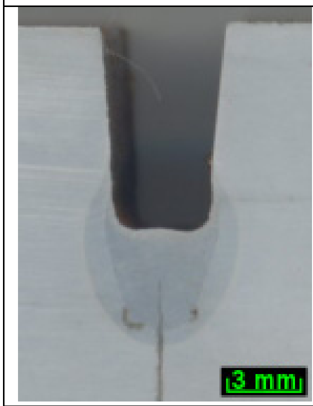
Trial E4 Proximity -0.5 mm	 E4.1 : 5 Hz	 E4.2 : 10 Hz
 E4.3 : 15 Hz	 E4.4 : 20 Hz	 E4.5 : 25 Hz

Figure 5.54 - Bead profiles from trials E4 – sidewall proximity -0.5 mm – torch oscillation width 2.5 mm

It is clear from Figure 5.53 and Figure 5.54 bead profiles that sidewall and bottom weld metal penetrations are shallow or non-existent. In fact, trial E1.5 shows no sidewall fusion from top to bottom, and the majority show sidewall penetration in the bead top with lack-of-fusion in both bottom corners. No bottom fusion is visible in all trials. Sidewall undercut defects are visible in all profile images, more evident in the right sidewall.

Central cracking was found in most specimens on trials E1 to E8, starting from the bottom of the bead and going up. In some cases the specimen broke in two while being cut in the saw. Both parts were joined and polished together and the profile image acquired. This did not affect the measurements taken for the purposes of the analysis.

Trial E2 Proximity 0 mm		
	E2.1 : 5 Hz	E2.2 : 10 Hz
		
E2.3 : 15 Hz	E2.4 : 20 Hz	

Figure 5.55 - Bead profiles from trials E2 – sidewall proximity 0 mm – torch oscillation width 2.5 mm (0.6 mm off-centre to the left)

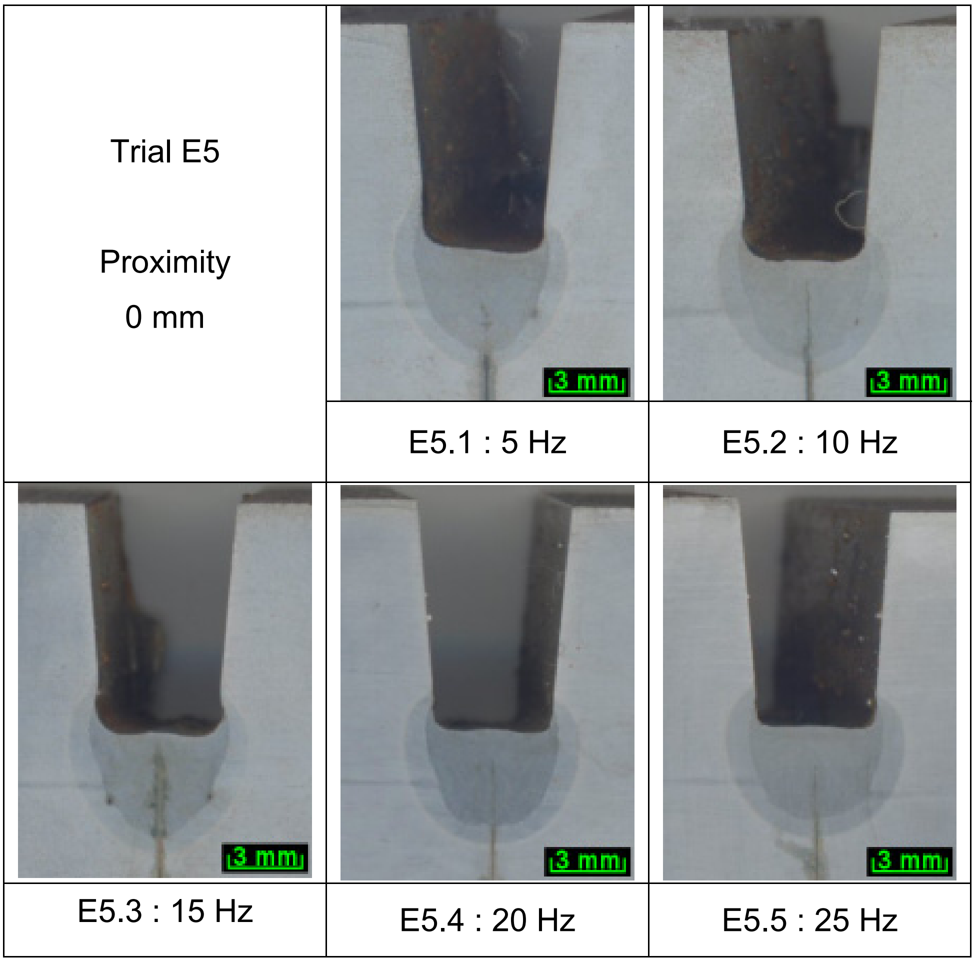


Figure 5.56 - Bead profiles from trials E5 – sidewall proximity 0 mm – torch oscillation width 2.5 mm

The bead profiles of Figure 5.55 and Figure 5.56 show a more even and round shape with fewer defects (0 mm proximity) than the previous beads of Figure 5.53 and Figure 5.54 (-0.5 mm proximity). It is apparent in Figure 5.55 that, as a result of a torch off-centre to the left of 0.6 mm, the bead top is higher on the left. Some lack-of-sidewall fusion in the corners are found in some cases.

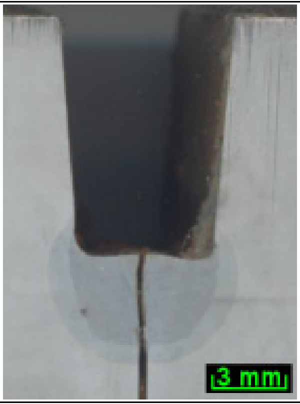
Trial E3 Proximity 0.5 mm		
	E3.1 : 5 Hz	E3.2 : 10 Hz
		
E3.3 : 15 Hz	E3.4 : 20 Hz	E3.5 : 25 Hz

Figure 5.57 - Bead profiles from trials E3 – sidewall proximity 0.5 mm – torch oscillation width 2.5 mm

Figure 5.57 bead profiles (+0.5 mm proximity) show fewer defects than previous beads. The bead shape has a round effect and fusion was successfully achieved in both groove corners. A small inclusion was found in the left corner of trial E3.4.

As a summary of bead profile analysis of groove trials with 2.5 mm oscillation width, it can be said that:

A) Moving from positive to negative proximities increase the number of corner lack-of-sidewall fusion and undercut defects
B) Negative proximities reflected no root fusion and in some cases also no sidewall fusion

C) Increasing sidewall distance creates rounded bead shapes

D) 0.6 mm off-centre on a 3.5 mm groove width produces a visible top bead inclination

The following figures are from 3.7 mm oscillation width trials.

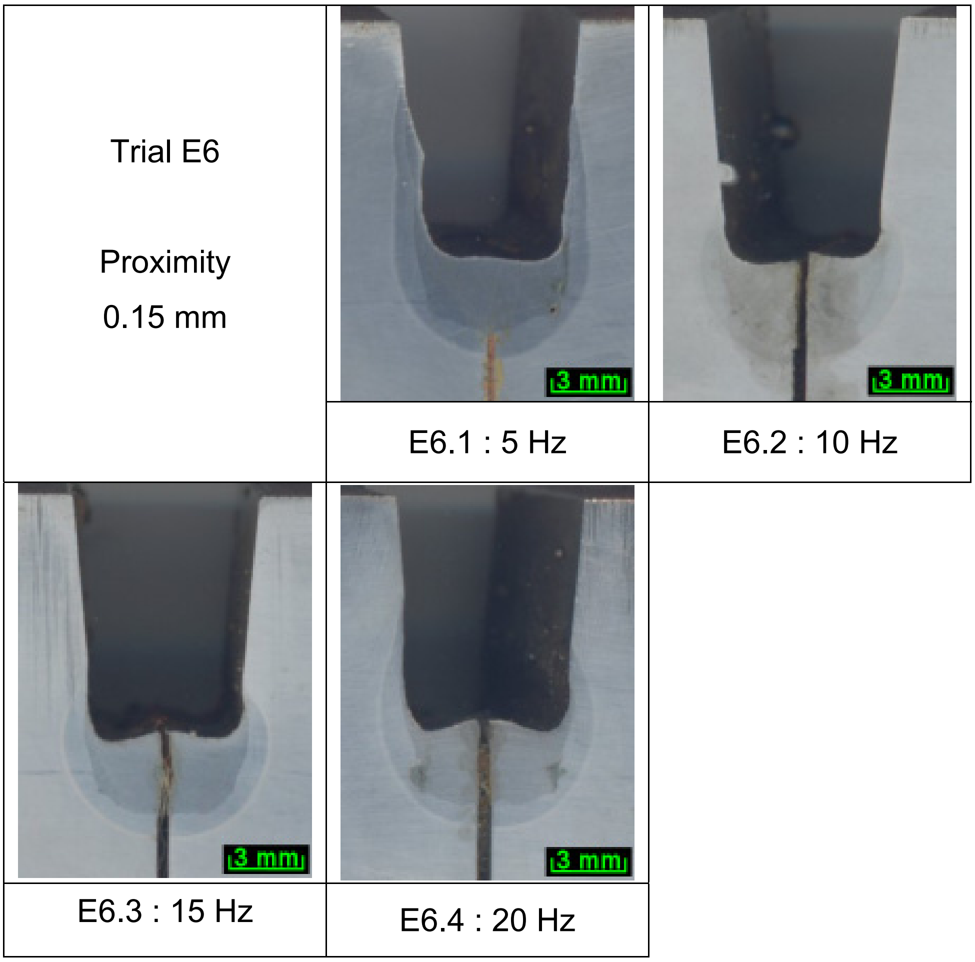


Figure 5.58 - Bead profiles from trials E6 – sidewall proximity 0.15 mm – torch oscillation width 3.7 mm

From the bead profiles of Figure 5.58, it is apparent that there is lack-of-sidewall fusion in the groove corners and undercut defects, visible in the trials. All beads show irregular shapes.

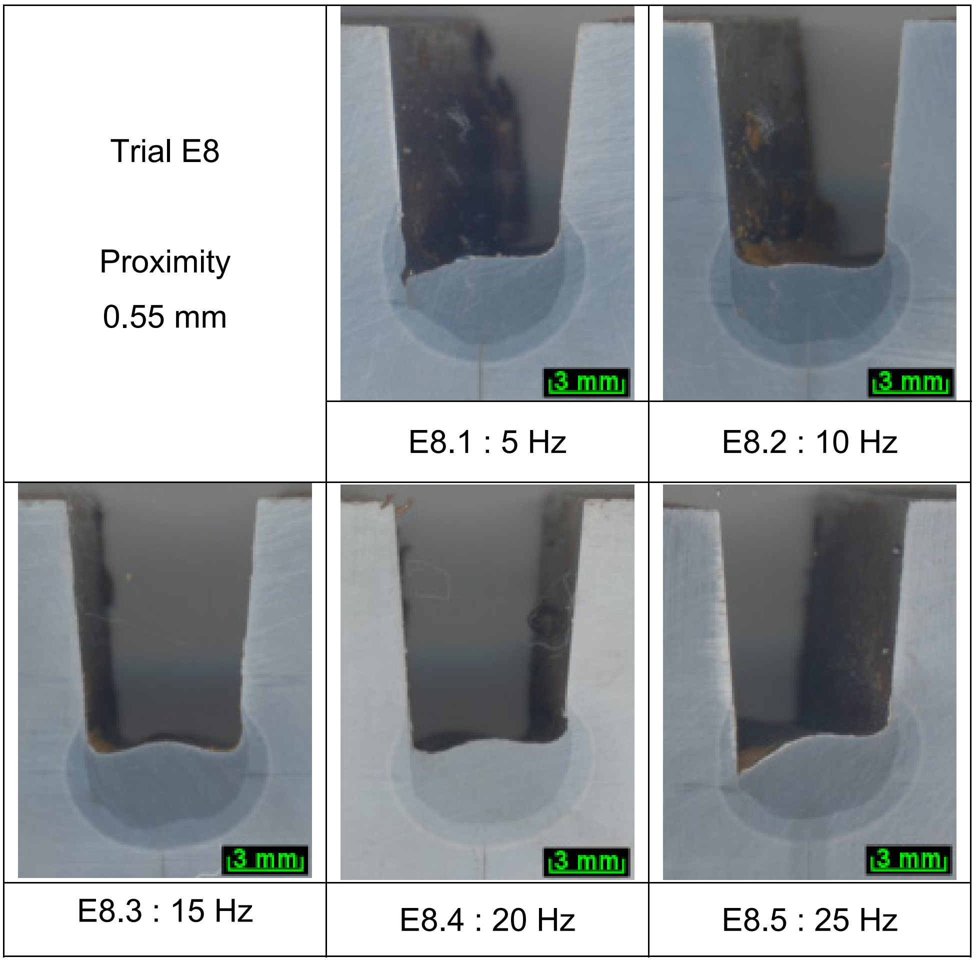


Figure 5.59 - Bead profiles from trials E8 – sidewall proximity 0.55 mm – torch oscillation width 3.7 mm (0.5 mm off-centre to the right)

Although with a round shape and no major defects, bead profiles of Figure 5.59 are slightly higher on the right sidewall, more visible in E8.1 and 8.5 trials. They are a result of 0.5 mm off-centre to the right.

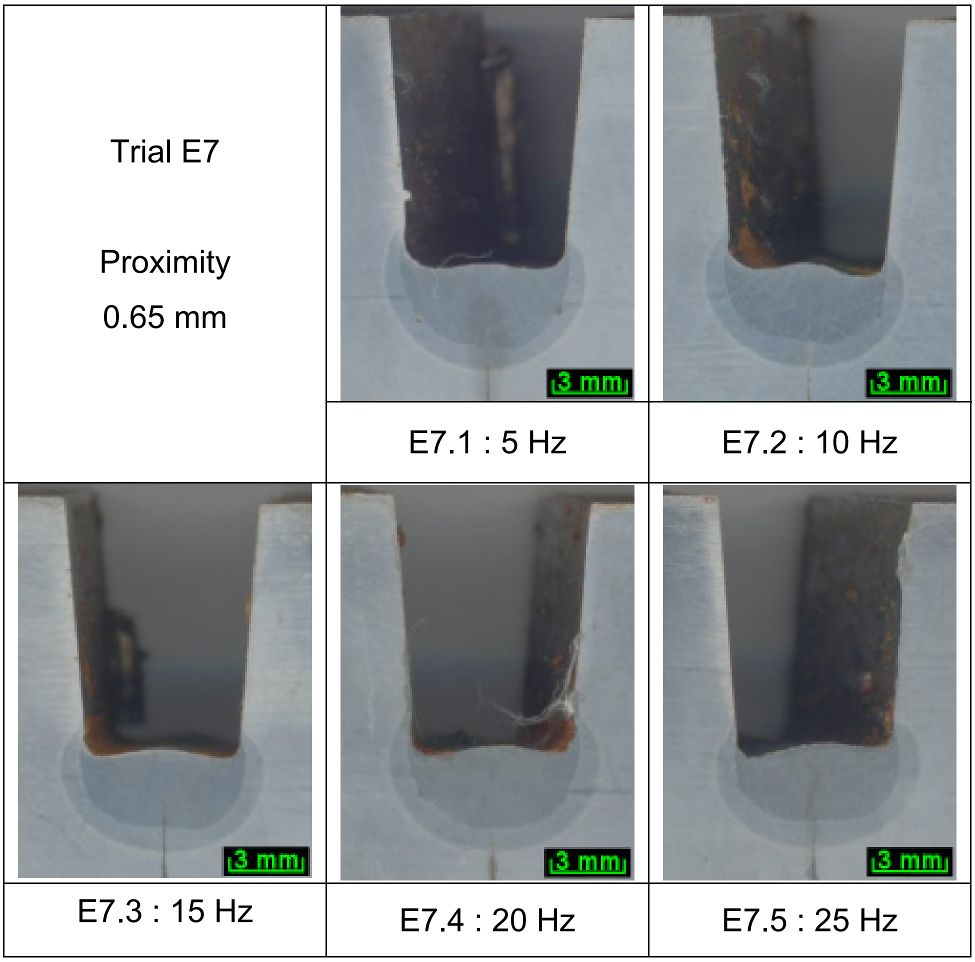


Figure 5.60 - Bead profiles from trials E7 – sidewall proximity 0.65 mm – torch oscillation width 3.7 mm

A rounder bead shape is apparent in the bead profiles of Figure 5.60. Small undercuts are visible in some specimens and a small inclusion is visible in the right corner of trial E7.1. In general, the number of defects is substantially lower than 0.15 mm proximity trials. With the increase of groove width, bead shapes are visibly shallower.

In conclusion, bead profile analysis of groove trials E1 to E8 has found the following general results:

a) Moving from a positive to a negative sidewall proximity increases the number of corner lack-of-sidewall fusion and undercut defects
b) Little root fusion is apparent for small groove widths, and in some cases also no sidewall fusion
c) Increasing sidewall distance creates rounded bead shapes with good weld metal penetrations and fewer defects
d) High sidewall distances reduces sidewall penetration
e) Torch off-centre produce inclined beads

6 Discussion

Through-the-arc sensing was evaluated for torch position control in GMAW-P narrow groove pipe welding. Experiments were performed to assess the arc as a sensor, based on arc voltage and current signals, to evaluate feasibility of achieving control of CTWD, cross seam position and groove width. Experiments were also conducted to assess the influence of oscillation frequency on arc signals and fusion characteristics. A set of initial trials (experimentation phase 1) was devised and the results provided key information for CTWD and cross seam control algorithms development. A test bed for the control algorithms was conducted to assess the performance and accuracy of the developed system (experimentation phase 2). Torch oscillation width and frequency was analysed (experimentation phase 3) in order to provide fundamental information for groove width control development.

The three phases of experimentation are here discussed in this chapter. The methodology followed for each experimentation phase can be found in chapter 3, software algorithms used for analysis of experiments and developed torch position control are in chapter 4; and experimentation analysis and results are in chapter 5.

6.1 Through-the-arc sensing for CTWD variations

Through-the-arc sensing is a method where the welding arc is used as a sensor to supply information of the welding process. The use of arc signals for control has been used by many researchers and by the welding industry for process quality control, process instability detection and torch path guidance. In the latter, through-the-arc sensing is being applied with success for GMAW and GMAW-P in wide grooves. No literature was found for CTWD control using GMAW-P narrow groove welding with a 5° bevel.

Researchers have explored arc signal variations to identify torch relative position to the seam, for various types of welding preparations. The method is to follow a signal pattern in current or voltage produced as a result of torch motion and the type of welding preparation in use. The key parameter that establishes a direct relationship between arc signal and welding preparation is the CTWD.

Welding parameters are controlled and monitored by the welding power supply. CTWD is a parameter controlled mechanically by torch positioning. Variation in its value changes circuit resistance and hence, welding current and voltage (arc signals). Depending on the type of welding power supply in use, one arc signal is more sensitive (has a higher variation) than the other to CTWD changes: For CV is the current which is most sensitive and for CC power supplies voltage is most sensitive. GMAW-P is most commonly used as a CC process and hence CTWD variations are more apparent in the voltage signal.

In GMAW, arc signal variations are sensed by analysing the signals in time. A CTWD variation causes an instantaneous change in the signal value and measurements can be performed at any instant of the waveform. In GMAW-P arc signals are pulsed and their values strongly vary in time. Measurements should be consistently taken in the same part of the waveform such as pulse peak or background period in order to achieve greatest sensitivity to CTWD changes. Averaging the pulsed signal may also be used as described in section 6.2.1.2. To simplify the terminology in this work, signal variations in GMAW-P are related to signal difference in the same part of the waveform. Also, the term "*peak voltage*" is used to mean "voltage at peak current" and "*background voltage*" means "voltage at background current".

6.2 *Initial trials*

Initial trials were performed aimed at understanding the variation of arc signals in the narrow groove with GMAW-P. This project was manly focused on the Lincoln Power Wave F355i GMAW-P power supply although trials were also conducted in experimentation phase 2 using Lincoln Power Wave 455M. The F355i was designed for constant current GMAW-P with the monitored pulse waveform accurately following the programmed waveform from the Lincoln Wave Designer software.

Good control is maintained over current as shown in the graph of Figure 5.1. This graph also shows a voltage signal with a periodic modulation created by the torch weaving effect producing cyclical variations in CTWD values. The effect is more clearly visible for longer periods in the waveforms of Figure 5.2. In these waveforms the relationship between torch oscillation frequency and wave shape is clear. For instance with no torch oscillation, no modulation is visible in the waveform. This arc signal behaviour indicates that, with the welding setup in use (power supply, wire, gas and parameters), the voltage signal clearly reflects torch behaviour and hence CTWD variations. It was then important to understand how this voltage signal could be related to torch positioning in order to develop a proper control.

6.2.1 Voltage signal signatures

In order to understand the relationship between torch positioning inside the groove and voltage values of a pulsating signal, off-line signal processing was performed. The voltage data signal captured in trials A1 to A6 was used and three voltage signatures were analysed for the control development:

a) Peak voltages (voltage at peak current)
b) Background voltages (voltage at background current)
c) Average voltage (moving average from the total voltage signal)

Peak voltages are considered here as the voltage obtained at the maximum current value at each pulse. This forms a new signal based in the peak to peak variations of the stream of pulsed voltage data. Background voltages are based in the same principle as peak voltages but considering the voltages at minimum current of each pulse.

6.2.1.1 Peak versus background voltages

The current and voltage waveforms of Figure 6.1 were extracted from trial A1. Peak and background values (green and light blue in the graph) are voltage measurements taken at maximum and minimum current values of each pulse. It is apparent from the voltage waveform that it is in the background period of the pulse where most irregularities are found, produced by short-circuiting events and droplet detachments. Short-circuits can occur by the droplet touching the weld pool while still attached to the wire, and transferring due to surface tension forces. According to the observations performed in all trials, short-circuiting events tend to occur in the down slope of the current pulse, not coinciding with the minimum current value of each pulse. This means that background voltages measured at the minimum current values are valid, even with short-circuiting in the same background period. However, the values might still be affected by the voltage drop produced by the short-circuit.

Spurious voltages in the background period were also observed as shown in the waveform of Figure 4.8 (p. 75). In general, background voltages were found to be less sensitive than peak voltages. In other words, for the same CTWD variation produced by torch oscillation, a smaller signal variation is observed using background voltages rather peak voltages. From Figure 6.1, the peak voltage variation during 80 ms of trial A1 was more than the double the background voltage variation (3.97 V against 1.72 V). For longer periods, it is clear that the same pattern is found as shown in the voltage waveform of Figure 6.2, in this case for trial A2. A higher voltage difference signifies greater sensitivity of the torch positioning control system.

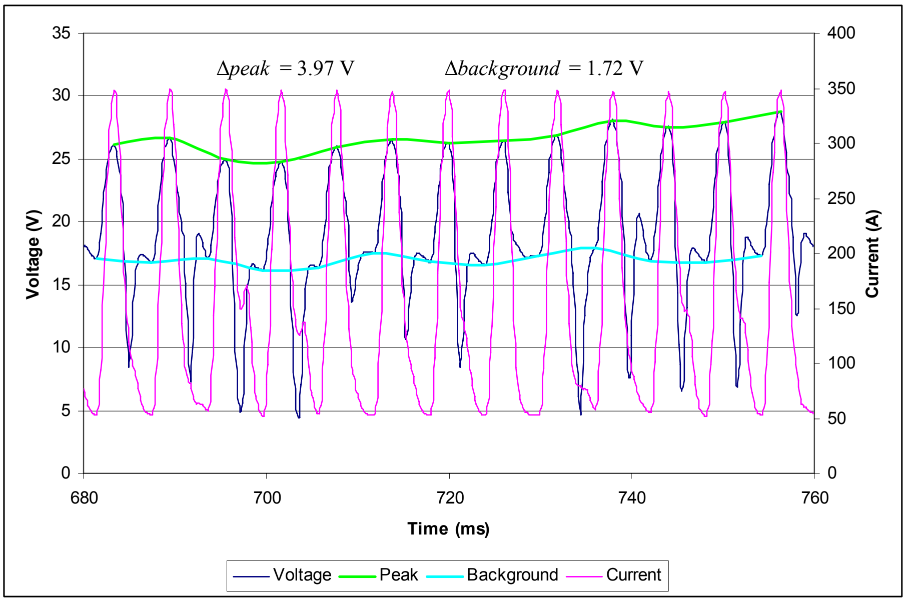


Figure 6.1 – 80 ms period of voltage data signal extracted from the raw data file of trial A1, 680 ms after the trial start – Oscillation frequency 3.33 Hz, Oscillation width 5 mm

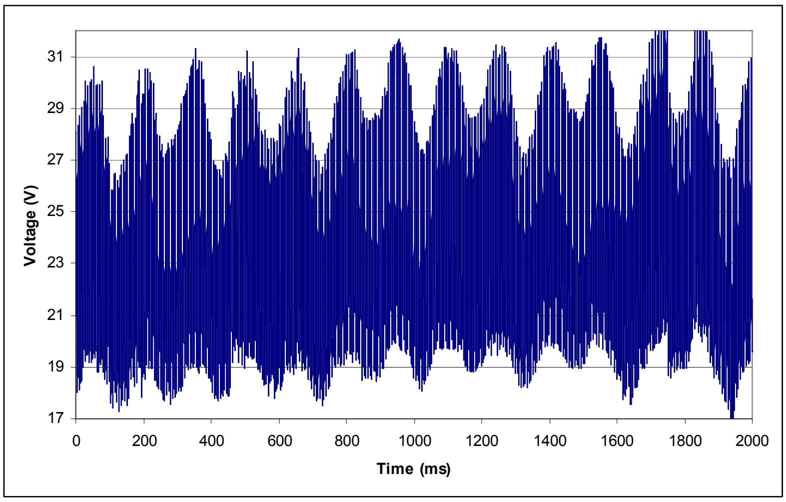


Figure 6.2 – Trial A2 voltage signal – Oscillation frequency 3.33 Hz, Oscillation width 6 mm

Reduced susceptibility to irregularities or spurious signals and increased signal sensitivity were the key factors in this comparison and hence peak voltages were found to be more suitable to be used for torch position control.

6.2.1.2 Peak versus moving average voltages

Moving average of arc voltages is a signal processing technique that smoothes the original signal according to an average window value. The main advantage of this technique is the continuous availability of a voltage value at any instant. Peak voltages are only available after the end of each peak pulse and may not coincide with the precise moment when the value is measured by the control system. In this case, the last available peak voltage shall be used instead. On the other hand, the measured value of a moving average does not reflect the exact instant of the original signal, due to the lag or time shifting created by the number of samples needed for averaging (average window). Figure 6.3 shows the raw voltage data of trial A1 plotted with four moving averages at different average windows.

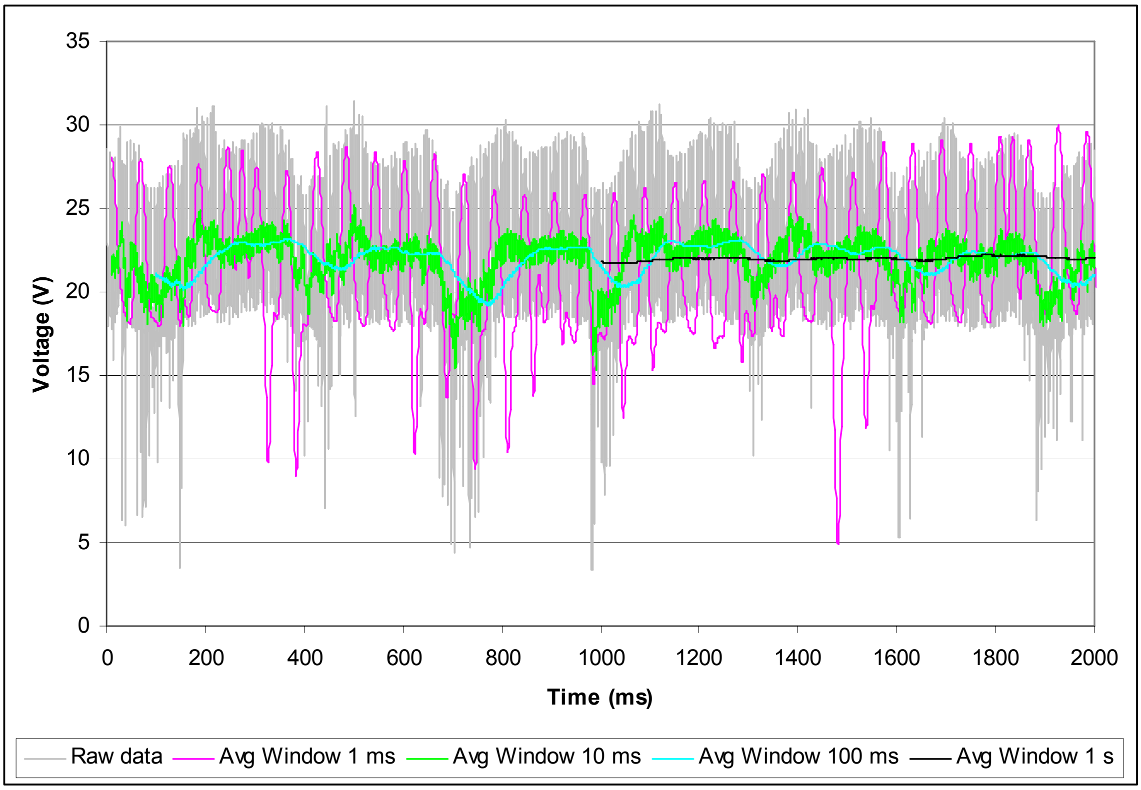


Figure 6.3 – Trial A1 raw voltage data and respective moving averages at different average windows

Several features were directly observed from the graph of Figure 6.3:

1) Signal displacement or time shifting from the events of the original signal, produced by the choice of the average window
2) Small average window (1 ms) produced an oscillatory and irregular signal or aliasing effect (pink line in the graph)
3) An average window of 10 ms still contains a large amount of noise but reflecting the original signal undulation
4) Short-circuiting visibly influences the moving average by lowering its value
5) An average window of 1 s produces a flat average effectively filtering any torch variation in short periods of time
6) An average window of 100 ms shows a smooth line, influenced by short-circuiting but still sensing torch variations in short periods of time
7) In all cases moving averages showed a smaller voltage sensitivity compared to peak voltages

Moving averages in the graph of Figure 6.3 were produced by average window values attributed with no relation with the original voltage signal. However, if the pulse frequency is known, the average window can be selected at a more appropriate value avoiding the aliasing effect created by the 1 ms average window of Figure 6.3. The graph of Figure 6.4 shows the raw voltage data from trial A1 and two moving averages with average windows of one and ten pulses respectively.

The result of using average windows related to the pulse length is a smoother average signal with a smaller time shifting. Also, voltage sensitivity has increased but is still lower than sensitivity using peak voltages. The values are approximately as follows, extracted around the B1 position (Figure 6.4) of the original waveform and the B2 position for the moving averages:

Δ *Peak voltages* = 3.55 V

Δ *Moving averages* (100 ms average window) = 1.09 V

Δ *Moving averages* (10 pulses average window) = 1.98 V

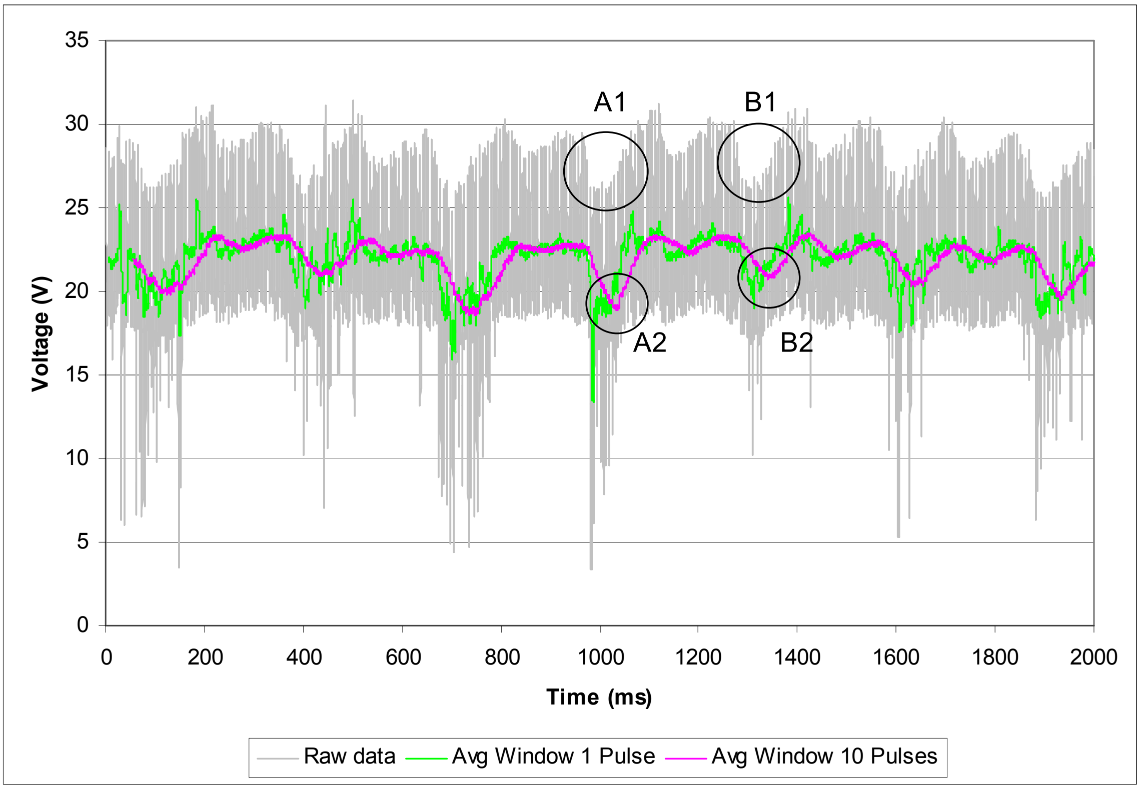


Figure 6.4 – Trial A1 raw voltage data and respective moving averages at different average windows

The application of the second method of moving average calculation, based on the pulse length, requires the knowledge of pulse period or frequency. Usually this value is not known in GMAW-P because it is internal to the power supply, and thus, it is necessary to implement pulse frequency measurements based on detection on arc signals. Since pulse frequency is not always constant with some power supplies, when an adaptive mode option is enabled, some care should be taken.

Another important issue on the use of moving averages in general is the influence of short-circuiting on the continuous average values. For instance, at positions A1 and B1 on the graph of Figure 6.4, peak voltages are similar in value. On the other hand, at positions A2 and B2 of the moving average the values are different. It is clear that the moving average at position A2 is more influenced by short-circuiting and thus it reflects a lower voltage value. Short-circuits are frequent when welding with the narrow groove technique. Usually short arc lengths are used to avoid arc deflections to the sidewalls as a result of the very narrow groove (4.5 mm in the root pass). If the arc is constantly deflected to the sidewalls, there is little base metal fusion.

It was concluded that peak voltages are more consistent, more sensitive, less susceptible to noise or short-circuiting influence and therefore more adequate for torch online positioning control, with the welding setup used (power supply, wire, gas and parameters). This result is also consistent with Jieyu et al [66] who stated that higher currents are more robust and less influence by noise than lower currents. Moving averages can also be used for torch positioning, although with less sensitivity than peak voltages and demanding for a well chosen average window. Also, two types of voltage average windows can be used: by fixed time or by pulse length, the latter being more sensitive and with less time-shifting. Short-circuiting affects both types of voltage averages.

6.2.1.3 Comparison with previous research

A similar study between the use of peak and background voltages to attain torch cross-seam control in GMAW-P was performed by Barnett et al [60]. These authors found peak voltages to be more noisy and irregular than background voltages. Unfortunately, the original waveforms of voltage and current from their experimentation were not provided by the authors in the published paper. The two filtered waveforms of peak and background voltages (Figure 6.5) indeed show peak voltages more irregular and noisy than background voltages. In contrast, it is clear from the waveforms that higher signal sensitivity is found with peak voltages rather than background voltages.

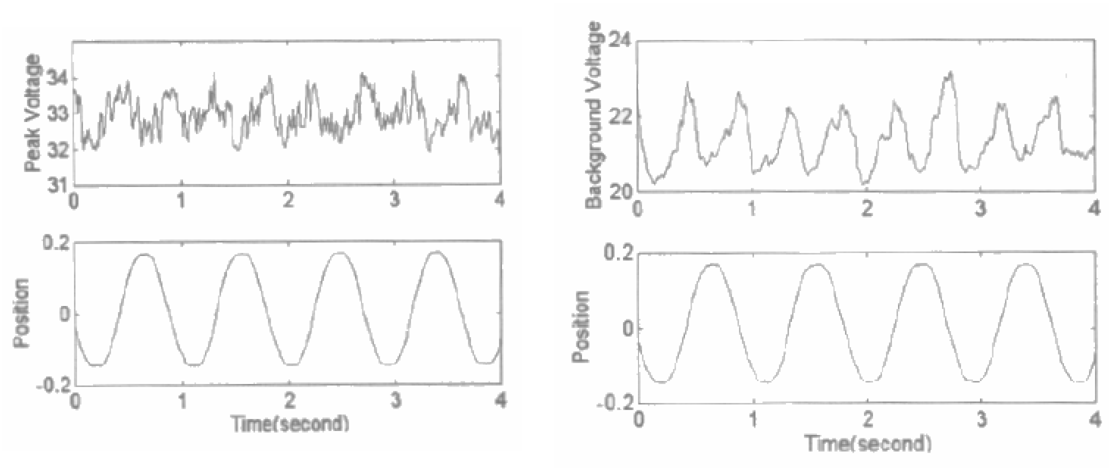


Figure 6.5 – Peak and background voltage comparison from Barnett et al [60]

Different GMAW-P power supplies have different characteristics with different pulse shapes and signal waveforms. An analysis of the current values of the original signal could supply the answer to the noisy peak voltages.

From the same university and with the same co-authors as in the previous research work, Rashid et al [85] developed a cross-seam control for V-groove with GMAW-P based on voltage averages. The author used a patented [84] GMAW cross-seam control algorithm where the frequency of short-circuits, created mostly at lower CTWD values when the torch achives the oscillation extreme, is analysed to define the torch oscillation dwelling at maximum excursions. The dwelling period is then the signal averaging period. In general, the author of the patent recognizes that a fixed sampling period of 100-200 ms is sufficient for most welding scenarios, although this value is not quoted by Rashid et al [85]. The voltage signal average as a function of torch oscillation position presented by Rashid et al [85] is shown in Figure 6.6. This waveform shows a continuum voltage average and not an average obtained only on torch maximum excursions. Also, voltage sensitivity is approximately 2 V, similar to the 1.98 V obtained by the 10 pulse average window discussed previously in section 6.2.1.2. It is important to realise that with different welding parameters, similar results were obtained.

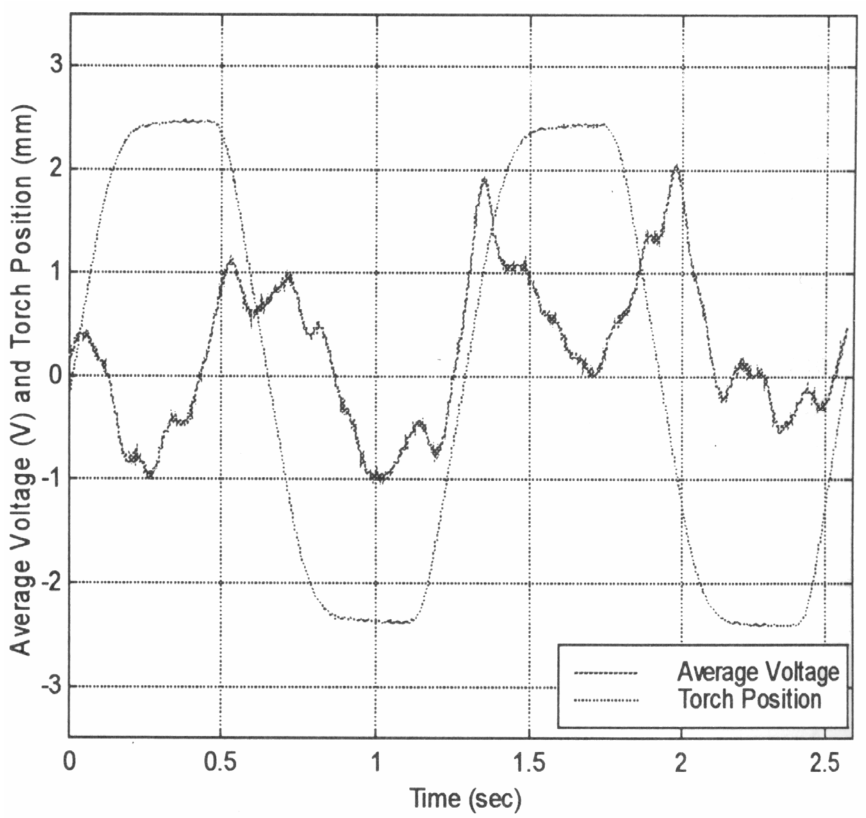


Figure 6.6 – Variations of voltage average in relation to torch oscillation position by Rashid et al [85]

6.2.1.4 Off-line signal analysis for torch position control

In order to establish the relationship between peak voltage values and torch position, initial algorithms were developed to analyse the acquired data from the initial trials A1 to A6. The algorithms isolated the torch positions and established the respective peak voltages (Figure 5.3 – p. 98 and Figure 5.4 – p. 99). It is apparent that in trial A1 the torch was off-centre, by the 2 V difference between the blue and red lines of Figure 5.4 (p. 99). This data is very consistent in the 2 seconds of data analysed. Peak voltages at centre of oscillation in trial A1 (green line of Figure 5.3 – p. 98) show a cyclical irregularity. As it can be seen in the waveform of trial A4 (Figure 6.7) these values also represented by the green line are more stable and the irregularities are not cyclical.

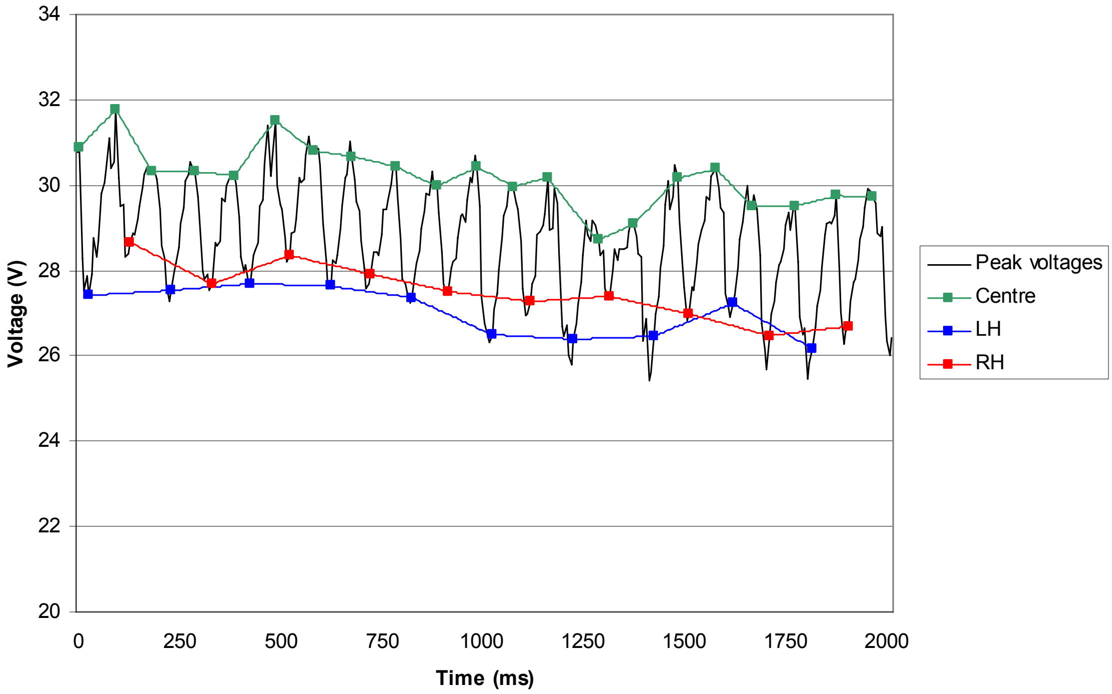


Figure 6.7 - Output data at the conclusion of the first and second pass algorithms from trial A4 (5 Hz torch oscillation frequency)

The difference between both trials is the off-centre value of trial A1. It is apparent from Figure 5.3 (p. 98) that the highest voltage values on the green line always occur after the lowest voltage values represented by the blue line. In other words, voltages were higher at the centre of oscillation when the torch was coming from the sidewall closer to the torch centre of oscillation.

It is important to note that the algorithms applied to the initial trials were only voltage based and searched for maximum and minimum voltages from the peak voltages signal. These values may not perfectly coincide with the three torch positions (left maximum excursion, centre and right maximum excursion). If this information was known, the obtained voltage values at each torch position might have shown lower variations in time. This has shown how important is to know the torch position during oscillations to achieve control. The control system has then to relate arc signals (voltage in GMAW-P) with torch oscillation positions to detect torch displacement inside the groove.

Three important conclusions were also reached from this off-line signal analysis. The first is related to the importance of torch oscillation for torch displacement detection. As shown in Figure 6.8 of trial A6 (0 Hz torch oscillation frequency), the result of first and second pass algorithms could not detect proper torch positions in the 2 s of sampled data. The LH, RH and Centre data is overlapped. With no torch oscillation, any displacement of the torch in relation to the groove cannot be related to left or right off-centre errors and thus it is not possible to correct the torch to the right path. For instance, if the torch is heading towards one of the sidewalls, voltage signal starts to decrease but the control system is not able to distinguish which movement direction must be used to correct the path.

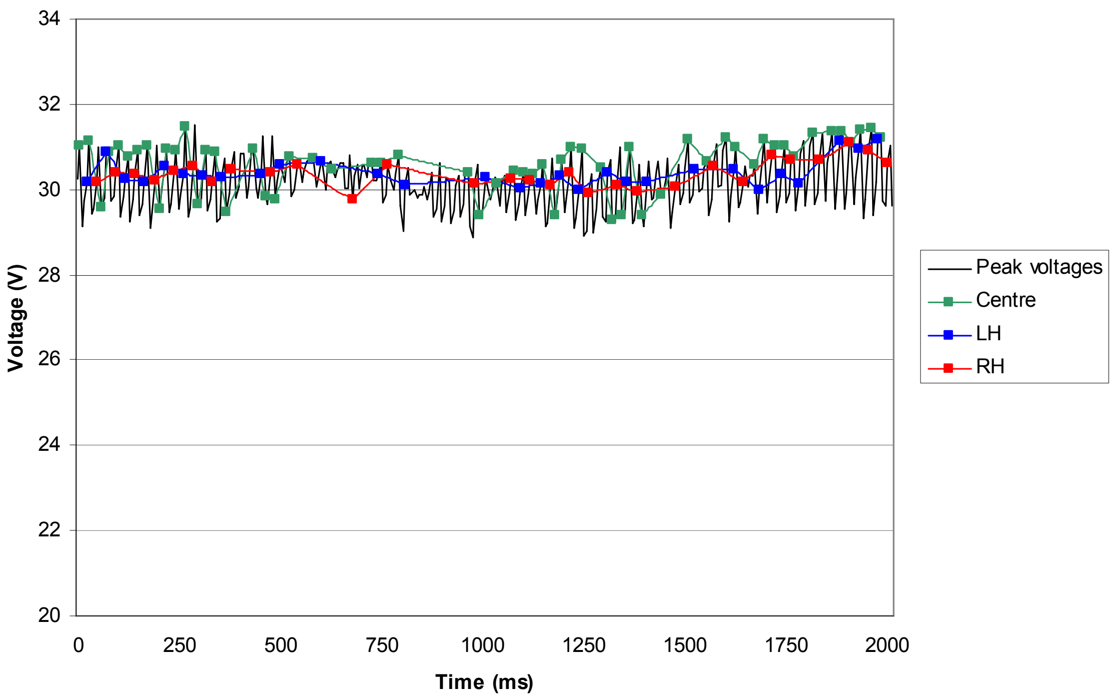


Figure 6.8 – Output data at the conclusion of the first and second pass algorithms from trial A6 (0 Hz torch oscillation frequency)

A second conclusion from the analysis was drawn about the use of peak voltages for control and the importance of pulse frequency versus torch oscillation frequency. From the earlier discussion, it was stated that a peak voltage may not coincide with the control system measurements for the particular torch oscillation positions. For instance, when the torch reaches one extreme excursion, voltage measurements at that particular point may not coincide with a peak voltage or voltage at peak current. In this case, the last acquired peak voltage value should be used and that voltage may have happened some milliseconds before. For low torch oscillation frequencies that may not be a problem but for high torch oscillation frequencies, some milliseconds may correspond to a voltage value taken far from the torch extreme excursion.

For instance, trials A1 to A6 pulse frequency was 181 Hz and torch frequencies varied from 0 Hz to 5 Hz. In the worst case (5 Hz), there are 36.2 pulses for a complete oscillation, or 18.1 pulses for each excursion. If Rashid et al [85] pulse parameters were used instead (100 Hz pulse frequency) with torch frequencies increased to 25 Hz, only two arc pulses (two peak voltages) would be available for one torch excursion. These two pulses could also coincide with any part of the torch position and their voltage difference would have no valid information on the torch position in relation to the groove.

As a third conclusion, a relationship between oscillation width and groove width was also found in peak voltage signal. The graph of Figure 5.5 (p. 99) clearly shows a relationship between voltages (peak voltage values difference between torch at centre of oscillation and torch at maximum excursion) and torch oscillation width, for the same groove width. A wider oscillation produces a higher voltage difference. This was an indication of torch oscillation width control feasibility using peak voltages. This control is important to detect groove width variations produced by pipe bevelling or metal distortions created by the heat in each welding pass. To accurately develop this control, more experimentation was needed and it was included in experimentation phase 3.

In conclusion, from the off-line signal analysis it can be said that:

1) Peak voltage values and variation has clearly shown a relation to the three torch positions
2) Torch off-centring can be found by voltage difference at maximum torch excursions
3) Torch off-centring can only be detected by arc signals (voltage in GMAW-P) if the torch oscillates across the seam
4) The information of torch position at any instant is important for an accurate relation with arc signals
5) Peak voltages for torch position control have a limited use dependent in the ratio between pulse frequency and torch oscillation frequency
6) Peak voltage difference between torch at centre of oscillation and torch at maximum excursion can be used to detect groove width variations

6.2.2 Cross-seam and CTWD control

Torch position control using through-the-arc sensing has evolved with technology. Initial devices were made of discrete electronic components with limited capabilities for updates and changes. Recently, the application of Digital Signal Processors (DSP) has brought huge capabilities to welding automation in general and through-the-arc sensing applications in particular.

For this project, this was not an exception. The availability of a programmable device able to perform arc sensing and torch positioning facilitated the whole project. Machine code programs were then needed to create an intelligent link between arc sensing and torch positioning, based on well defined algorithms to implement the automated control for torch cross-seam guidance and CTWD stability control.

From the previous conclusion of the initial trials and off-line algorithms, two separate control algorithms were devised for online cross seam position and

CTWD control. The following sections are the discussion of the two sets of algorithms.

6.2.2.1 CTWD control algorithms

From the initial trials, it was concluded that peak voltages produce a better signal to be used in torch position control. However, to develop a control system ready to be applied in the field, it would be necessary to create or revise welding procedures in order to determine the value of optimum peak voltage for each welding pass.

The welding procedure is a quality control tool based on a set of norms and parameters created in order to facilitate the welder's work in adjusting the right welding conditions for the job in hand. The welder has the freedom to perform the necessary adjustments in order to sustain a good and consistent weld quality. Voltage values in actual pipeline GMAW-P welding procedures are the resultant average voltage obtained at the end of each welding pass when the procedure was first produced, based on a predefined CTWD.

Before the welding starts, one of the last parameters adjusted by the welder is the CTWD, to certify that the same value is used as is described in the welding procedure. Changes in groove height produce changes in CTWD and in turn changes in the voltage signal (peak and averages). If a defined CTWD value in the welding procedure has generated a certain average voltage, using the same welding conditions, the opposite is also true. This means that by following the same voltage average, the original defined CTWD in the welding procedure should be achieved.

Using this premise, and in the absence of a peak voltage definition in the welding procedure, CTWD control was developed following the average voltage value defined in the welding procedure. As it was discussed earlier, the average voltage is much better used for long periods of time. CTWD changes in the groove passes are not abrupt and thus CTWD control can be implemented by using existent welding procedures.

The two types of moving averages (fixed time or pulse length) discussed in section 6.2.1.2 were then assessed. The fixed time type with an average window of 100 ms was chosen since it is less demanding in terms of algorithm processing, with apparent good results in long periods and low variations in time.

6.2.2.2 Cross-seam control algorithms

As with CTWD, torch off-centring variations in pipeline welding are not abrupt but with the use of narrow groove techniques, a small torch off-centre variation may produce lack-of-sidewall fusion or undercut defects. To achieve a proper cross-seam control, signal analysis and processing should be attained in every torch oscillation cycle. Cross-seam control algorithms hence need a fast response from the controller to process arc signals and torch positions, and to feedback torch position correction commands. With new high travel speed dual tandem systems (up to 2 m/min), torch oscillation frequencies also need to increase in order to perform correct and consistent sidewall fusion. Projected oscillation frequencies are around 10 Hz, several times higher than the ones used by Rashid et al [85] (2 Hz) and Barnett et al [60] (1.2 Hz) systems. In this case, peak voltages are the key signal for the implementation of a fast cross-seam control, due to the availability of sensitive signals that reflect instantaneous torch positions.

6.3 Control algorithms test bed

The trials conducted in experimentation phase 2 served to assess the feasibility and evaluate the robustness of the developed arc sensing and processing algorithms for torch cross-seam and CTWD control, using GMAW-P in the 5° narrow groove. The algorithms were divided in two parts:

1) CTWD control by using voltage moving average with a fixed average window of 100 ms
2) Cross-seam control by differentiating peak voltages at torch maximum excursions

Both algorithms worked simultaneously implemented in assembly language and C code on a Motorola DSP. Arc signals and exchanged messages between the DSP and the welding head were monitored and analysed and the results presented in section 5.2.

6.3.1 Tandem welding with enabled adaptive mode

Trials were performed with a different welding setup from the one used for trials A1 to A6 and algorithm development. The previous setup used a single wire with a GMAW-P Lincoln Power Wave F355i power supply, with disabled adaptive mode. The new setup used two synchronised GMAW-P Lincoln Power Wave 455M power supplies in tandem welding with enabled adaptive mode. Pulse parameters and welding procedure were also different. Tandem welding has different characteristics to single wire welding due to the interaction of both arcs. Also, the GMAW-P power supply adaptive mode enables self-regulation of the arc length by changing the pulse parameters. This process counter-acts the corrections made by the automation system. Both tandem welding and adaptive mode were used as part of the system's robustness test.

6.3.2 Expected trials accuracy

According to Bould et al [124], in a clear and steady environment and with the help of a digital calliper the human eye accuracy is 0.26 mm reproducible to ±0.1 mm. It is reasonable to say that in a pipeline welding environment with a moving welding head and an oscillating torch, the accuracy of manual torch position adjustments performed by the welder is much lower than 0.26 mm. In most cases, the initial setup of adjusting torch oscillation width is performed by calliper comparisons and hence, according to Bould et al [124], it might produce an oscillation width error of 0.26 mm. Runtime adjustments of oscillation width, torch alignment with groove centre and CTWD are also performed by the welder following his expertise and intuition. In this way, welding consistency is difficult to be achieved at a high level of accuracy.

Eichhorn et al [38] found that a tracking accuracy better than ±0.5 mm produced perfect welded seams with no fusion problems or pores, for 0° narrow gap GMAW with 40 mm workpiece thickness and 12 mm groove width. This accuracy value was then defined as the maximum acceptable tolerance to be achieved by the developed system for both CTWD and cross-seam control.

6.3.3 CTWD control

Trials B1, B3 and B4 have shown that CTWD control algorithms performed well within the intended maximum tolerance of 0.5 mm, with standard deviations of 0.05 mm, 0.14 mm and 0.1 mm respectively. Standard deviation in millimetres was calculated based on the voltage standard deviation and its relationship to CTWD obtained from the graph of Figure 5.6 (p. 100).

CTWD control was obtained using a feedback loop controller where measured voltage averages are compared with a voltage reference and the difference results in proportional (PID controller) torch height correction sent to the welding head. The aim of the controller is to produce a constant voltage average, as discussed earlier in section 6.2.2.1.

Small fluctuations of voltage average can be seen in the graphs of Figure 5.7 (p. 101) to Figure 5.9 (p. 102), but essentially voltage remained steady for large induced variations such as trial B3 and B4 down step, thus revealing good CTWD control. In normal operation, pipeline welding groove height changes are progressive and not abrupt as performed in trials B3 and B4. The fast torch height recovery achieved by the developed system has shown a high level of sensitivity and response, not necessary for the pipeline welding application. An average window of 1 s may have a more progressive response and can be used instead.

6.3.4 Cross seam control

Trials B1, B3 and B4 have shown good cross-seam control consistency inside an intended tolerance of 0.5 mm, with standard deviations of 0.14 mm, 0.39 mm and 0.18 mm respectively. The tracking was consistent in all trials as shown in the graphs of Figure 5.13 (p. 105) to Figure 5.15 (p. 105).

Cross-seam control was obtained by a feedback loop controller where peak voltage values at maximum torch oscillation excursions were successively compared and the difference generated a 0.1 mm horizontal torch movement towards the correct side (section 6.2.2.2).

Trial B3 showed the worst standard deviation value of the three trials (0.39 mm), visible in the blue trace variation on Figure 5.11 (p. 104). For this trial, torch oscillation width was kept at 2.5 mm as used for trial B1, when ideally it should have been 4.5 mm. In comparison, torch oscillation width in trial B4 was set to a more appropriate value for the groove width and the resultant standard deviation is half of trial B3 (0.18 mm).

6.3.5 Control summary

The two control algorithms performed well inside the required tolerance. Their implementation was separated as discussed previously and although both used feedback loop methods, each controller was implemented differently. With CTWD control, a PID controller was used because the risk of strong variations of voltage average is low and thus three extreme situations might happen in case of abnormalities, strong signal instabilities or wrong setup:

1) Torch is constantly sent up due to constant low voltage indication with arc extinction on a too high CTWD, stopping the whole process
2) Torch is constantly sent down due to an abnormal high voltage measurement as a result of a momentarily process instability. The system recovers after the instability due to the low voltage value. For a prolongued instability, the arc voltage will drop at some stage and hence the system elevates the torch to the right position
3) If the external voltage reference is set too low, the torch is moved down to obtain the intended voltage average value, hence the contact tip might collide with the root material

For horizontal corrections with the cross-seam control, the risk of sidewall collision is high should any disturbance in arc signals occur. The contact-tip distance from the sidewall during oscillation cycles in the 5° narrow groove is sometimes tenths of a millimetre. Signal instabilities due to droplet detachment, welding pipe position, short-circuiting, groove width changes or inappropriate torch oscillation width may interfere with a correct control action and move the torch in the wrong direction. For this reason, it was decided that in each torch half oscillation cycle, peak voltage measured at maximum torch excursion is compared to the peak voltage from the opposite maximum excursion and the torch is moved 0.1 mm towards the correct position. In case of signal disturbance, if the difference value moves the torch in the wrong direction, the system can recover in the next oscillation cycle. Sometimes the error can be recurrent as is visible for instance in the 2^{nd} and 7^{th} seconds of trial B4 (Figure 5.12 – p. 104).

In conclusion of experimentation phase 3, the feasibility of through-the-arc sensing for torch position control using GMAW-P in the 5° narrow groove was verified. The use of voltage moving average for CTWD control and peak voltages differential on torch maximum excursions for cross-seam control was assessed and evaluated showing good vertical and horizontal tracking results, better than the intended 0.5 mm accuracy. The algorithms have shown robustness to welding setup changes such as tandem welding with synchronised GMAW-P power supplies and enabled adaptive mode. From the conducted trials, the use of an appropriate torch oscillation width with groove width was found to be fundamental, otherwise tracking inaccuracy and fusion defects may occur. This reinforces the importance of developing an automated torch oscillation width control working in conjunction with the developed control system. For the overall system and algorithms evaluation, this was considered an excellent result as the system performed better than expected.

A production demonstration trial for the sponsors of this research was also conducted under normal operating conditions from top to bottom of the pipe. This hot pass trial was performed on the 0.9 m diameter (36 inches) X80 line pipe, with 5º bevel narrow groove using single wire GMAW-P. The welding head band was deliberately misaligned with the groove centre by 25 mm at half way from the pipe top and an initial part of the run was performed over a previous hot pass bead to demonstrate the system’s ability to control torch height. The trial was performed successfully demonstrating good performance and accuracy of the cross-seam and CTWD control algorithms.

6.4 Torch oscillation width and frequency

As discussed in the previous section, torch oscillation width control is important to achieve good tracking control and to attain optimum fusion profiles to avoid defects. Also, according to many authors, torch oscillation frequency in GMAW produces different arc signal sensitivities as a function of the working oscillation frequency.

From the initial trials, a basic degree of torch oscillation width detection in GMAW-P by peak voltages was performed. The experimentation was not conclusive about what is the optimum torch oscillation width for a particular groove width due to lack of essential data such as torch oscillation position. Experimentation phase 3 was devised to acquire key data in order to find optimum torch oscillation width values and also to assess the influence of torch oscillation frequency on sensitivity to peak voltages. As discussed earlier, torch oscillation width control can be achieved by peak voltage comparison between torch at centre of oscillation and torch at maximum excursion. Thus, the analysis performed in experimentation phase 3 for the proper development of this control is based on the premise of finding the right peak voltage reference value.

To simplify terminology, torch at centre of oscillation was abbreviated to T@C and torch at maximum excursion or at sidewall to T@S.

The discussion of this experimentation phase is divided in three sections:

d) Arc signal sensitivity comparison between CV GMAW and GMAW-P
e) Single sidewall proximity trials
f) Double sidewall (groove) proximity trials

6.4.1 Arc signal sensitivity comparison between CV GMAW and GMAW-P

Arc signal sensitivity for welding automation using through-the-arc sensing technology is important because it determines the degree of response of the automation system. A higher sensitivity usually reflects an increase of system's accuracy. However, the magnitude of the signal may change with the welding process parameters creating variations on signal sensitivity. This is important when signal reference values are used for control. Arc signal sensitivity is known to be higher with the increase of torch oscillation frequency, on CV or CC GMAW. This means that for each oscillation frequency, a different reference value must be applied to attain proper control. It is therefore important to understand the effects of using different torch oscillation frequencies on arc signal sensitivity using GMAW-P.

6.4.1.1 GMAW trials

According to many authors, in CC GMAW voltage sensitivity is affected by torch oscillation frequency. Higher oscillation frequencies produce higher voltage sensitivity. In CV GMAW it is the current sensitivity that is most affected. For GMAW-P this influence is unknown. Trials C1 to C5 were devised to generate the needed data in order to understand and quantify this oscillation frequency influence on arc signal. The experiments were conducted with both CV GMAW and GMAW-P processes, using a welding setup similar to the used by Pan [7] (section 3.3.2.1).

From the trials performed with the CV GMAW process, the results obtained as shown in Figure 6.10 were similar to Pan's results in the graph of Figure 6.9.

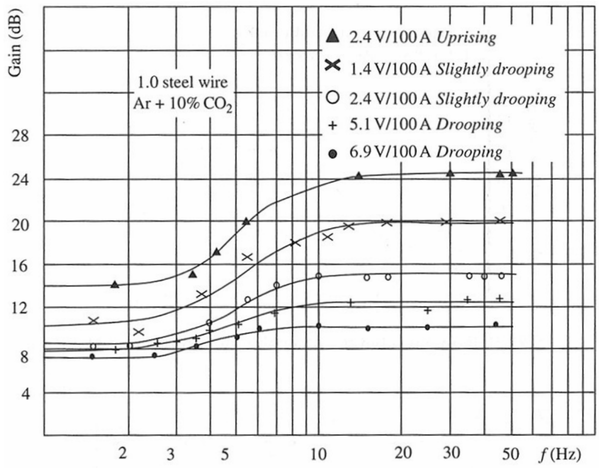


Figure 6.9 – Pan's magnitude-frequency (signal sensitivity as a function of torch rotation frequency) for different power supply characteristics in GMAW [7]

There is no published information available about the drooping characteristic of the CV GMAW power supply used for these trials to do a direct comparison between graphs. However it is clear from both graphs that similar trends are followed at approximately the same frequencies. This information is important to validate the welding setup used on trials C1 to C5 in order to compare CV GMAW and GMAW-P process results.

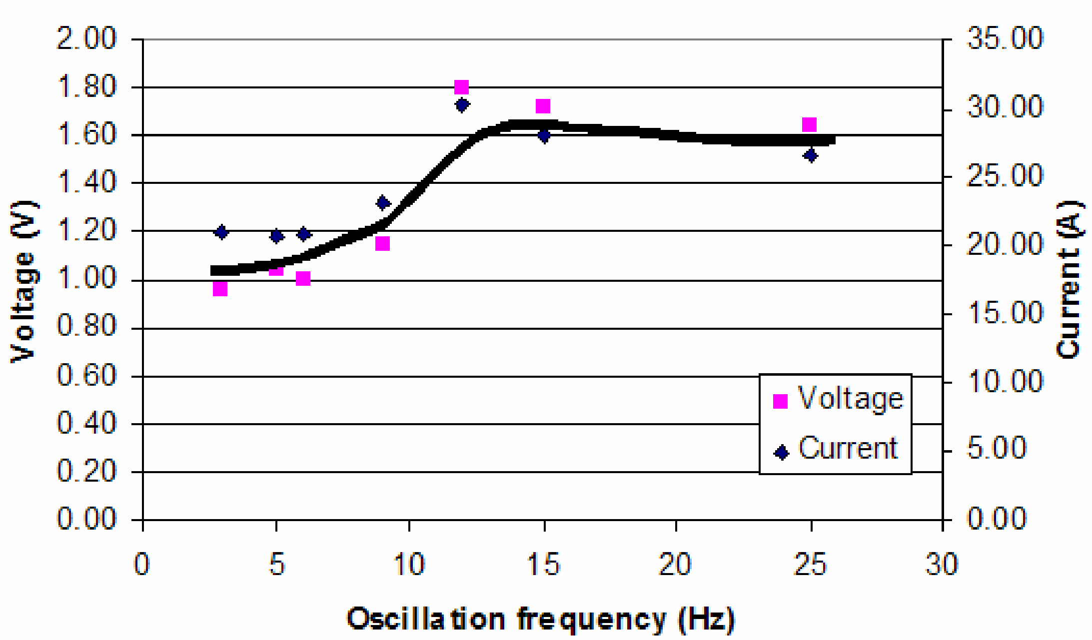


Figure 6.10 – Voltage and current absolute difference between left and right torch excursions, at different torch oscillation frequencies, in CV GMAW trials C4 and C5

6.4.1.2 GMAW-P trials

From the graphs of Figure 5.17 (p. 108) and Figure 5.18 (p. 108) for GMAW-P trials, there is no significant change of voltage sensitivity with oscillation frequency. There is a slightly indication mainly on Figure 5.18 (p. 108) that it might have some change but the data is not conclusive. Compared with CV GMAW where voltage variation has a factor of 2:1 (Figure 5.19 – p. 109 and Figure 6.10), GMAW-P voltage variation with oscillation frequency is close to zero and shows no significant advantage to be considered for torch position control.

It was then concluded that torch oscillation frequency produces no apparent influence on peak voltage sensitivity in GMAW-P, with the 45° plate inclination with the vertical welding setup. Similar sensitivity analysis was performed for both single sidewall and groove trials as discussed further.

6.4.2 Single sidewall trials with 5° preparation angle

Trials D1 to D4 were conducted using GMAW-P with a single 5° sidewall, in order to understand and evaluate peak voltage variations and weld metal penetration for changes in:

1) Torch oscillation frequency
2) Torch to sidewall proximity
3) CTWD

Arc signal sensitivity analysis was also performed for this set of trials.

6.4.2.1 Torch oscillation frequency influence

The results of this analysis are shown in the graphs of Figure 5.20 (p. 111). It is clear from the graphs that sidewall proximity produces different influences of torch oscillation frequency on peak voltages and weld metal penetrations. At +1 mm from the sidewall, the graph shows a clear reduction of bottom penetration with the increase of oscillation frequency although no explanation was found for this situation. At 0 mm and -1 mm proximities, metal penetration variation with oscillation frequency is not conclusive showing some scattered information.

Peak voltage variation also shows different behaviours as a function of sidewall proximity. Some of the lines on the graphs show slightly upward trend slopes but in general there is no consistent variation that may lead to a conclusive result. For the purpose of torch oscillation width control development, no conclusion can be taken from the analysed data.

6.4.2.2 Torch to sidewall proximity influence

The results of this analysis are shown in the graphs of Figure 5.21 (p. 113). Torch to sidewall proximity has clearly shown an increase in side penetration moving from positive to negative values of sidewall proximity but strongly reducing bottom penetration. This behaviour is caused by the longer arcs created near negative proximities due to the wire burn back. This matter is discussed later. Longer arc lengths produce shallower penetrations.

The analysis also shows that proximity to the sidewall increases peak voltage difference between T@C and T@S. From the graphs, it is possible to see that T@C peak voltage is similar in the three proximities and for different torch oscillation frequencies. On the other hand the T@S peak voltage decreases with sidewall proximity. This is caused by arc deflections to the sidewall shortening CTWD values, as discussed later. Shorter CTWD values produce short arcs and hence lower voltage values.

It is also important to note that 1 mm difference in sidewall proximity creates large differences in voltage signals and essentially in weld metal penetration, supporting the necessity of torch position control with low tolerances as has been achieved by the developed algorithms. The trials also demonstrated that at +1 mm distance the sidewall has minimal effect in the arc signal, and thus it cannot be used for control. In this case the cross-seam control moves the torch erratically inside the groove seeking for sidewalls, as was seen for trial B3 in experimentation phase 2.

For the purpose of torch oscillation width control development, it was concluded that the best compromise between good fusion values and peak voltage difference is at 0 mm proximity, for single sidewall trials. This value is further confirmed in groove trials.

6.4.2.3 CTWD change influence

The results of this analysis are shown in the graphs of Figure 5.22 (p. 114). The data from the graph for both side and bottom weld metal penetrations show a clear reduction in fusion with the increase of CTWD. It is known from GMAW welding in general that long arc lengths produce shallow penetrations. Also these results are consistent with those reported by Nagesh et al [125] who states that “too long” or “too short” arc lengths may result in poor weld metal penetrations. Recent work from Liratzis [126] has also found that different arc lengths in orbital welding produces different weld metal penetrations. Liratzis’s experiments were conducted at different angles around the pipe. For each angle several experiments were performed using different arc lengths obtained by changing the trim value of the power supply and thus changing the pulse parameters to create different arc lengths. The angle change shows a clear pattern on the bead shape varying from slightly concave on the flat position, strongly concave in the vertical down position and almost flat for overhead welding. Arc length variation produced different levels of depth of weld metal penetration inside each pattern of the tested angles. Longer arc lengths produced shallower penetrations.

A clear conclusion from the graph is that the value of 13.5 mm CTWD often used for pipeline welding is a good value. From the graphs, CTWD values between 14 mm and 15 mm produced better fusion values but it should be noted that experiments were taken in flat position. As discussed earlier, short arcs are more important for pipeline orbital welding application.

Peak voltages on the other hand show a consistent increasing trend and difference between T@C and T@S with the increase of CTWD, as expected. This consistency shows that the developed CTWD control algorithms are valid for a broad range of reference voltages in respect with CTWD values. Since the trials were performed with the same sidewall proximity, it can also be concluded that the consistency in voltage difference between T@C and T@S can increase the reliability of torch oscillation width control for various CTWD values.

6.4.2.4 Arc signal sensitivity analysis for single sidewall trials

The previous two analyses have shown the overall evolution of peak voltages between T@C and T@S for torch oscillation width control. The objective of this analysis is to understand how these peak voltages evolve for different proximities to the sidewall and at different torch oscillation frequencies. The analysis is based on the graphs of Figure 5.26 (p. 117) to Figure 5.28 (p. 118) and compares peak voltages from extreme torch oscillation excursions instead from T@C and T@S as previously.

Sidewall proximity has been shown to influence peak voltage evolution with oscillation frequency (Figure 5.26 – p. 117). This pattern was already found in the previous analysis of section 6.4.2.1. The higher peak voltage variations with torch at maximum excursions are found for the lower sidewall proximities although there is no consistent pattern on voltage evolution with oscillation frequency between different proximities. This analysis reinforces the previous conclusion that since oscillation frequency does not significantly affect arc voltage sensitivity, it is not necessary to consider different frequencies in torch oscillation width control development.

Sidewall proximity for Left-Right analysis follows a similar trend as for T@C and T@S analysis. The graph of Figure 5.27 (p. 118) shows a clear evidence of peak voltage difference between torch maximum excursions on single sidewall trials, when using closer sidewall proximities. The difference becomes more accentuated for negative proximities. The wire at this distance from the sidewall produces very short arcs and constant short-circuits, visibly lowering peak voltage values at the sidewall side and hence increasing the voltage difference between torch maximum excursions.

6.4.2.5 Summary

As a conclusion of single 5° sidewall trials, it can be stated that:

1) Increasing of torch oscillation frequency produces shallower beads and does not show significant and evident influence on peak voltage sensitivity
2) Torch proximity to the sidewall revealed that 1 mm change in proximity produces large differences in peak voltages and metal fusion
3) The best compromise between arc signal differentiation with torch position and metal fusion was found at 0 mm sidewall proximity and thus this value is followed for torch oscillation width control
4) The 13.5 mm CTWD value produces good weld beads and metal fusion although for the flat position the values of 14 mm to 15 mm would also be appropriate.

6.4.2.6 Arc recovery analysis for single sidewall trials

"According to the minimum voltage theory, the arc current flows into the weld pool along the shortest path" [127]. In other words, the arc current always follows the shortest ionisation path or resistance path between two potential poles, considering that the arc was already ignited. In GMAW or GMAW-P after ignition, an arc is formed between the wire tip and the nearest point on the base metal through the lowest resistance in the ionisation path. With an oscillating torch, arc changes contact location both at the target (workpiece) and at the wire (Figure 5.49 – p. 136). In most trials, the arc at T@C was established vertically with the bottom workpiece and at T@S with the sidewall. In Figure 6.11, it can be seen that at T@S, the shortest path to create an arc is l_w because this value is smaller than l_t,. The opposite occurs for T@C.

Where:

l_t is wire tip to bottom workpiece length

l_a is arc length

l_s is wire extension or stick-out

l_w is the wire tip distance to the sidewall

$CTWD$ is contact-tip--workpiece-distance

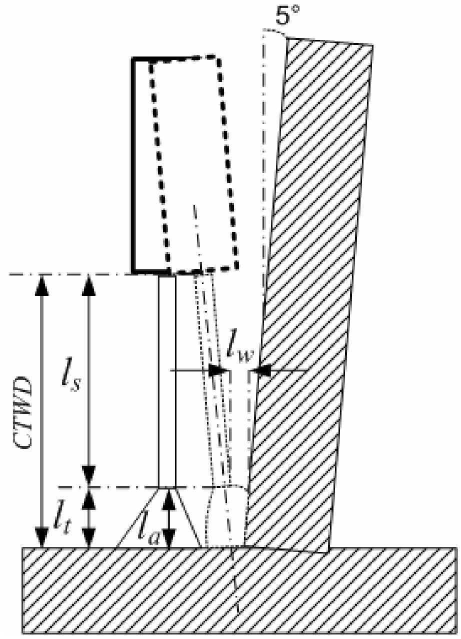


Figure 6.11 – Single sidewall length definitions

Generically, CTWD is assumed as the value starting at the end of the contact-tip and following a straight line to the workpiece, as adjusted by the welder (first definition). Also CTWD is known as the sum of two lengths: wire extension and arc length (second definition). The first and second definitions are coincident if the arc performs a straight line with the wire but they are not coincident when the arc is deflected to a nearby sidewall. Two CTWD values can be defined to simplify the calculations, with an oscillating torch in a single sidewall: one CTWD for T@C defined by the welder and one CTWD for T@S.

The new CTWD becomes:

$$CTWD = CTWD_0 - l_t + l_w \tag{6.1}$$

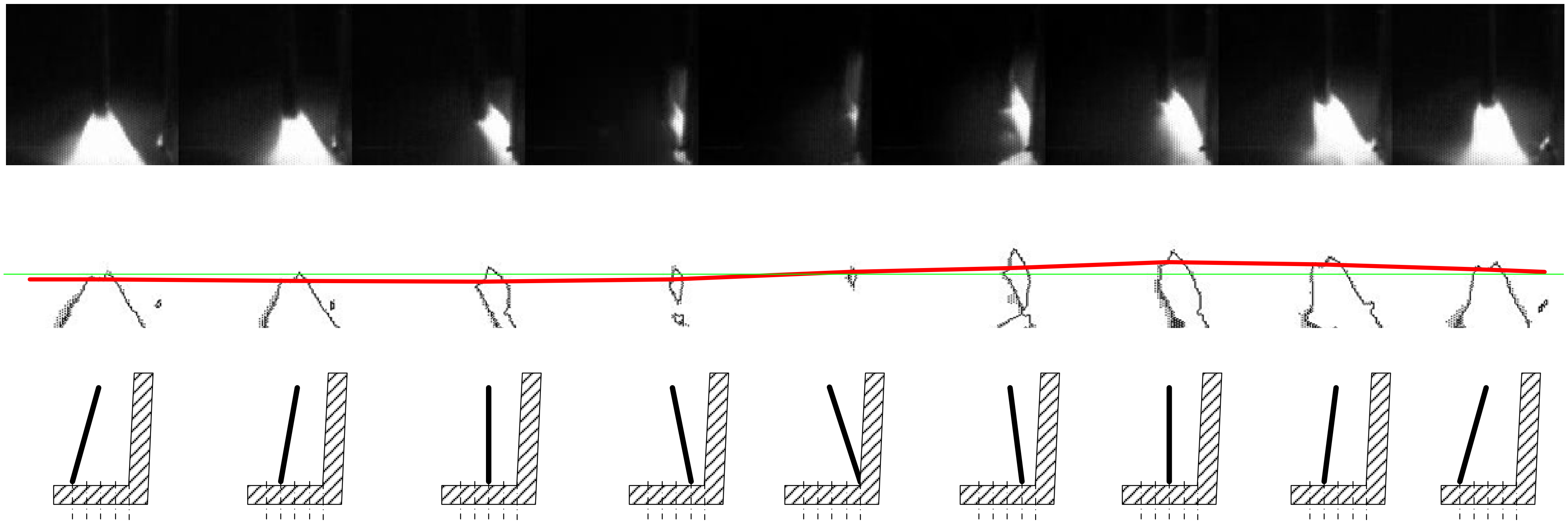
or

$$CTWD = l_s + l_w \tag{6.2}$$

where $CTWD_0$ is the value defined for T@C.

To better explain the process, it is important to consider the single sidewall example of Figure 6.11 with the torch coming from left and that at T@C when the system is balanced or in equilibrium state between arc length and wire extension ($l_a = l_t$). In other words, melting rate is in equilibrium with wire feed rate. The torch continues the oscillation towards the sidewall and at a certain point l_w becomes smaller than l_t. and the arc starts deflecting to the sidewall. This changes the CTWD and hence the equilibrium state by altering the balance conditions and forcing a melting rate change. More wire has to be melted to re-establish the process balance with the new CTWD and the wire extension l_s starts to decrease (burn back) also increasing l_t. When the torch moves away from the sidewall, l_t continues to increase until l_w becomes smaller than l_t and the arc establishes vertically with the bottom workpiece. When this happens, a new CTWD if found changing again melting rate. The system initiates a re-equilibrium process increasing l_s and decreasing l_t that is now equal to the arc length (l_a). This process continues until the sidewall is again reached and the process repeats itself.

Figure 6.12 shows this behaviour using arc images extracted at different torch positions during one complete oscillation cycle. The figure shows the arc images on the top. From the arc images, an edge finding algorithm extracted the arc shape leaving the wire tip shape visible for each arc image, as shown in the middle of the figure. The bottom images in the figure show the wire position in relation to the sidewall where the image was extracted. The red line joins the wire tip positions and the green line is the average point of the red line. It is clear from the figure that the red line comes from below and goes above the green line showing different l_t distances for different wire oscillation positions.



Top – real arc images

Middle – edge finding of **Top** arc images

Bottom – wire position of the oscillation where the arc image was extracted

Red line – wire tip position variation

Green line – average point of red line

Figure 6.12 – Image sequence or arc length variation at each torch position for a complete oscillation cycle of trial D4.3 (osc. freq. 5 Hz; prox. 0 mm; CTWD 15 mm)

Figure 6.13 shows a schematic explanation of this process. The number inside the circle on each picture is the relative torch weaving position within a complete oscillation. Number 1 is maximum excursion at the left side and number 5 is maximum excursion at the right side or at wall side.

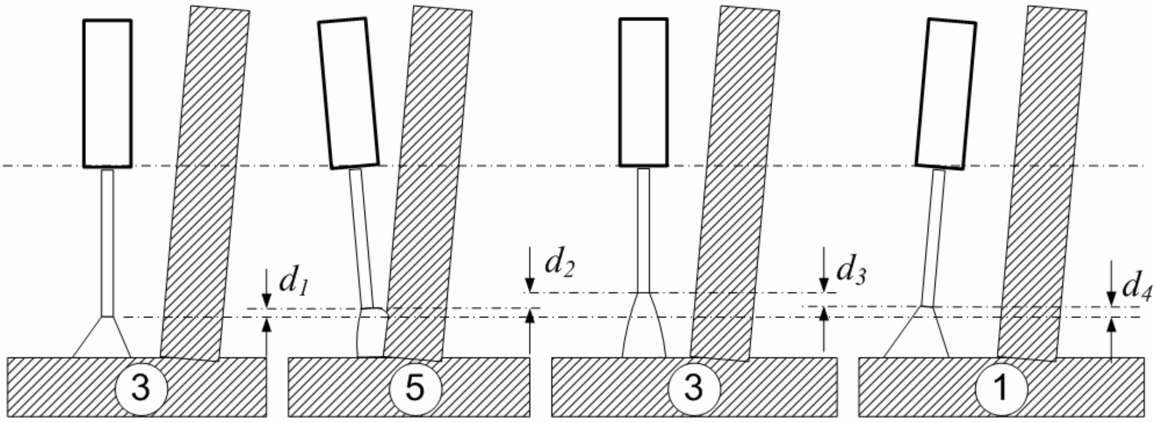


Figure 6.13 – Arc length and wire extension behaviour in single sidewall process

The graph of Figure 6.14 shows l_t behaviour extracted from a complete oscillation cycle of trial D4.3. The green arrows in the graph show the torch oscillation direction.

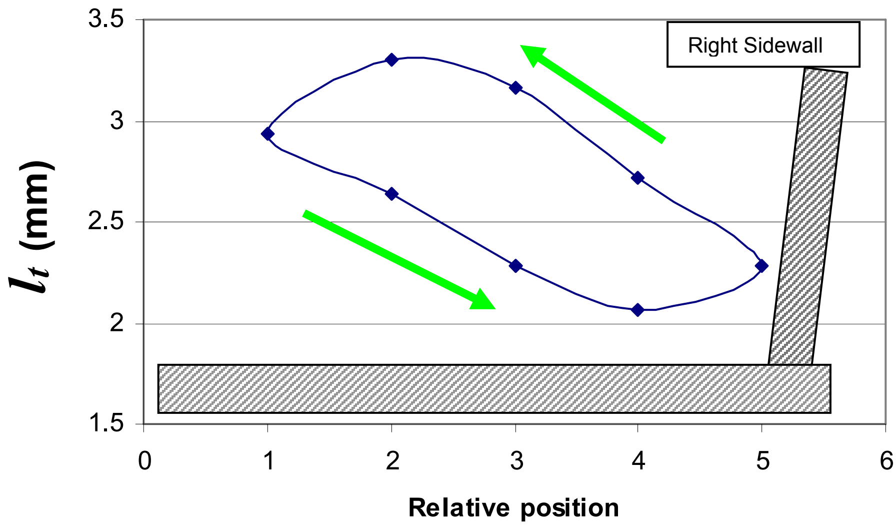


Figure 6.14 - l_t variation at each torch position for a complete oscillation cycle of trial D4.3 (osc. freq. 15 Hz; prox. 0 mm; CTWD 15 mm)

From the graph of Figure 6.14, two slopes can be noted following the torch oscillation direction:

1) Down slope from position 2 to position 4
2) Up slope from position 4 to position 2

From the graph of Figure 5.25 – p. 116, it was found that arc length varies at different rates according to CTWD. This was assumed to be a function of power supply behaviour and characteristics. The values were measured with the arc always established vertically with the bottom workpiece and thus, arc length variations can be directly related to l_t variations. Considering then time and voltage ratios from linear regressions of the graph, it can be said that:

$$V_r = (0.1968 * CTWD) - 1.5289 \tag{6.3}$$

and

$$t_r = (16.141 * CTWD) - 155.04 \tag{6.4}$$

where V_r is V/mm and t_r is ms/mm. Using the above expressions, for the down slope recovery with a CTWD of 13.5 mm, the rates are:

$$t_r = (16.141 * 13.5) - 155.04 = 62.86 \text{ ms/mm} \tag{6.5}$$

with a recovery peak voltage of:

$$V_r = (0.1968 * 13.5) - 1.5289 = 1.12 \text{ V/mm} \tag{6.6}$$

This means that if the arc needs to perform a length recovery at a CTWD of 13.5 mm, the process is performed at a rate of 62.86 ms/mm. It is clear that the complete recovery might happen if the torch oscillation is not sufficiently fast. In other words, if torch oscillation frequency is high, the system may not recover its balance and works always outside the ideal equilibrium values for each CTWD along the torch trajectory. This also means that voltage recovery may not be the same for each torch oscillation frequency.

To express l_t variation (Δl_t) as a function of torch oscillation frequency, one of the slopes can be considered and it represents half of a complete oscillation cycle. To ease calculations, both recovery slopes are considered symmetrical. Dividing the half time of a complete oscillation cycle (t_o) by the recovery speed for a particular CTWD (t_r), the result is the amount of l_t variation or Δl_t that may occur in that half cycle.

$$\Delta l_t = \frac{\left(\frac{1}{2} t_o\right)}{t_r} = \frac{t_o}{2t_r} \tag{6.7}$$

Since frequency (f) is the inverse of time ($1/t_o$), and to express the results in millimetres (10^3), expression (6.8) represents Δl_t as a function of torch oscillation frequency.

$$\Delta l_t = \frac{1}{2ft_r} * 10^3 \tag{6.8}$$

As an example, for a torch oscillation frequency of 5 Hz and a CTWD of 13.5 mm, l_t may recover:

$$\Delta l_t = \frac{1}{2ft_r} * 10^3 = \frac{10^3}{2 * 5 * 62.86} = 1.59 \text{ mm} \tag{6.9}$$

and with a peak voltage recovery of:

$$\Delta V = \Delta l_t * V_r = 1.59 * 1.12 = 1.78 \text{ V} \tag{6.10}$$

In the measured data of the graph of Figure 6.14, the parameters were: 15 mm CTWD and 5 Hz oscillation frequency. The calculated Δl_t is 1.14 mm and the measured data from the graph is 1.24 mm showing a good model accuracy of 0.1 mm.

For the same range of oscillation frequencies as used in the trials, peak voltage recovery values were plotted in the graph of Figure 6.15.

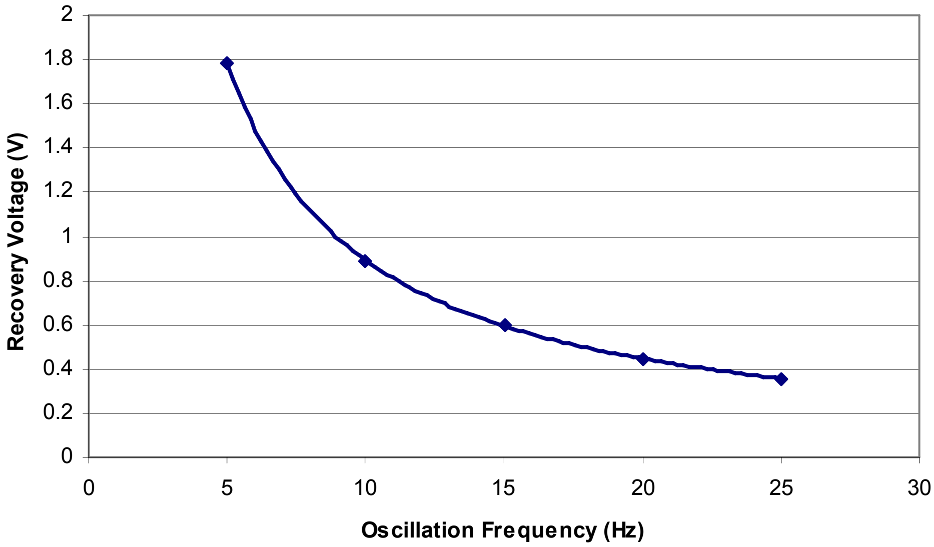


Figure 6.15 – Recovery voltage as a function of oscillation frequency for single sidewall welding

The data from the graph of Figure 6.15 is consistent with the results from experimentation. For instance, for trials C2 with the 45° sidewall angle and a CTWD of 17.6 mm (Figure 5.18 – p. 108), the model shows a voltage recovery difference between 5 Hz and 25 Hz of 1.19 V. From the results of trial C2 it was 1.15 V. The graph of Figure 6.15 also confirms that in fact the influence of torch oscillation frequency in voltage sensitivity is small.

It is now known that CTWD varies with different wire positions while weaving caused by the arc deflections with the sidewall. The initial CTWD adjustment is performed with the premise that the arc is always vertical but it is now evident that it is not. It is also known that different CTWD values have different recovery speeds with different voltage values. This complete mechanism produces a constant melting rate rebalance in every cycle of torch oscillation with the arc and stick-out not necessarily reaching a dynamic equilibrium in every point. Torch oscillation frequency in fact produces differences in this rebalance behaviour but not as big as expected initially.

6.4.3 Double sidewall (groove) trials with 5° preparation angle

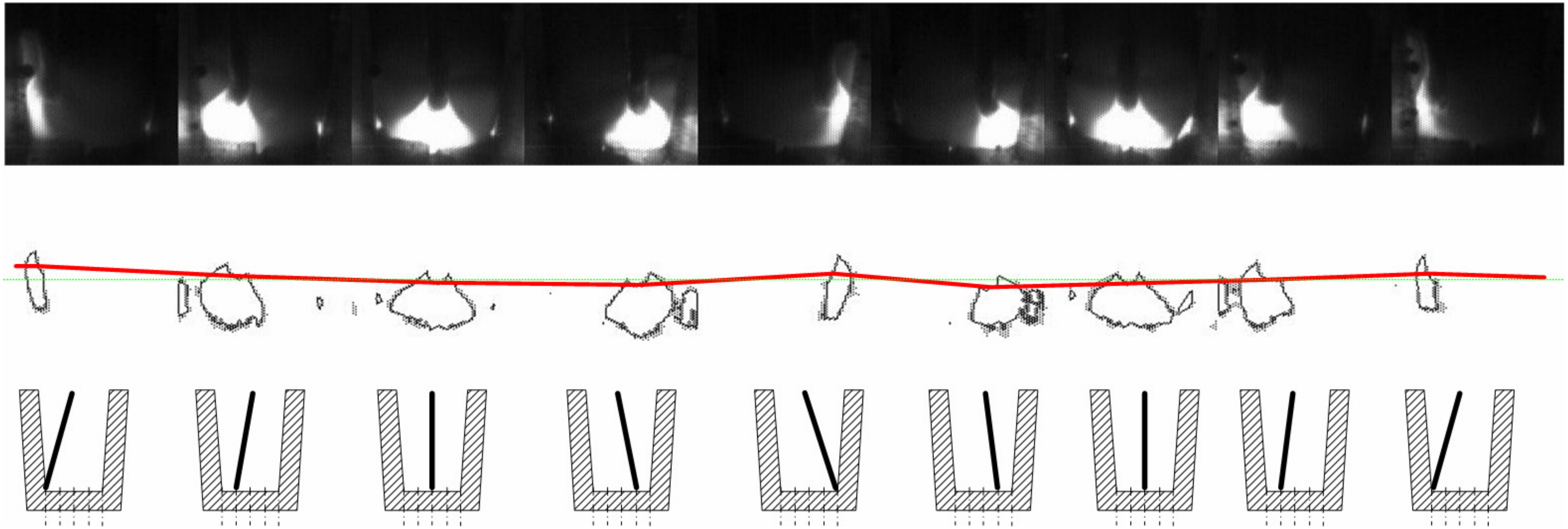
This set of trials was developed to understand the influence of torch oscillation frequency and wire proximity to the sidewalls when welding with GMAW-P inside a 5º narrow groove, for groove width control development. The results obtained from this set of trials revealed similarities with single sidewall trials. This section is divided in two parts:

a) Torch oscillation frequency influence
b) Proximity to the sidewall influence

6.4.3.1 Torch oscillation frequency influence

The influence of torch oscillation frequency in peak voltage variation and weld metal penetration is shown in the graph of Figure 5.29 (p. 120). It is clear that voltage variation is even lower and more consistent than with single sidewall trials. It reinforces the conclusion that torch oscillation frequency does not produce significant influence in the peak voltage signal.

Using the same analysis performed earlier, arc images and wire tip evolution (l_t) in groove welding were analysed as shown in Figure 6.16. Two distinct cycles are now observed when compared with the single cycle in single sidewall trials.



Top – real arc images

Middle – edge finding of **Top** arc images

Bottom – wire position of the oscillation where the arc image was extracted

Red line – wire tip position variation

Green line – average point of red line

Figure 6.16 - Image sequence or arc length variation at each torch position for a complete oscillation cycle of trial E6.1 (osc. freq. 5 Hz; prox. +0.15 mm)

The value of l_t was measured from a complete torch oscillation cycle of trial E6.1 and shown in the graph of Figure 6.17. The pattern is duplicated compared to single sidewall trials with the wire tip (l_t) varying in just half torch oscillation and producing half of the total amount of variation.

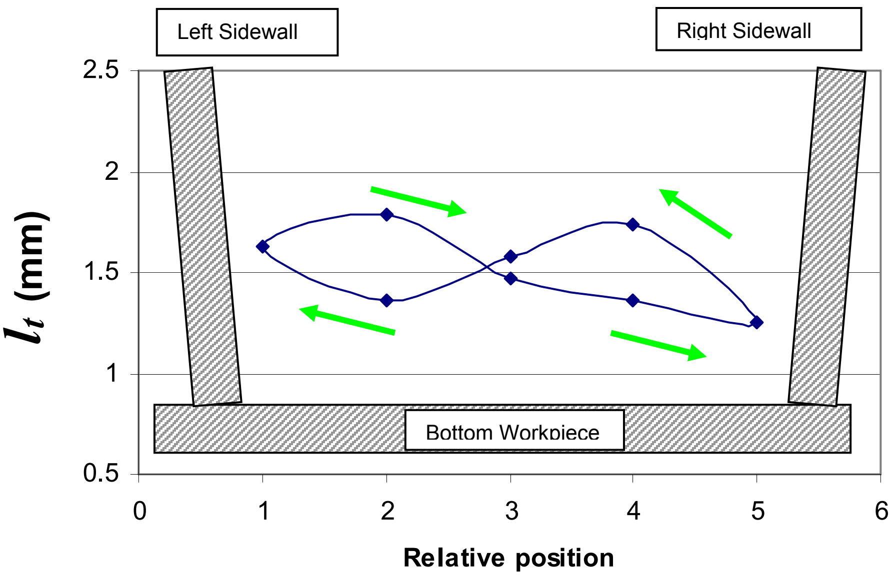


Figure 6.17 - l_t variation at each torch position for a complete oscillation cycle of trial E6.1 (osc. freq. 5 Hz; prox. 0.15 mm; 13.5 mm CTWD)

In order to show a more visible l_t variation, the Y-scale of the graph is different from the graph of Figure 6.14. From this new graph and following the previous discussion for single sidewall, four different CTWD values can be found associated with two up slopes and two down slopes of l_t. From expression (6.8), half of a complete oscillation cycle was considered because only two slopes were found. In the case of four slopes, the resultant expression is:

$$\Delta l_t = \frac{1}{4 f t_r} * 10^3 \qquad (6.11)$$

From the acquired data of trial E6.1 (Figure 6.17), torch oscillation frequency of 5 Hz and CTWD of 13.5 mm, the amount of l_t recovery in each slope is:

$$\Delta l_t = \frac{1}{4 f t_r} * 10^3 = \frac{10^3}{4 * 5 * 62.86} = 0.79 \text{ mm} \qquad (6.12)$$

and with a peak voltage recovery of

$$\Delta V = \Delta l_t * V_r = 0.79 * 1.12 = 0.89 \text{ V} \quad (6.13)$$

The measured Δl_t from the graph of Figure 6.17 was 0.54 mm showing a difference of 0.25 mm with the calculated value. More factors might be influencing the model accuracy but in general it seems fairly accurate. A similar peak voltage recovery graph as a function of oscillation frequency is shown in Figure 6.18.

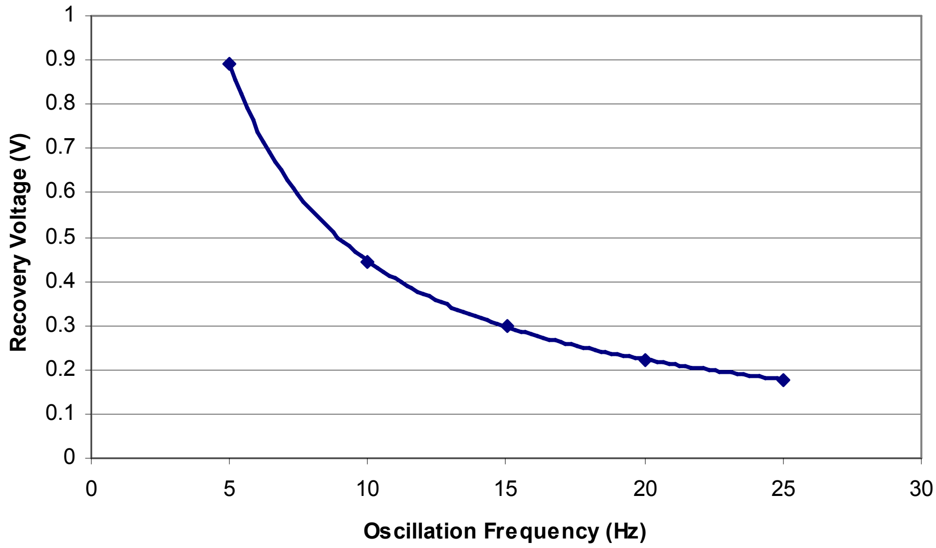


Figure 6.18 - Recovery voltage as a function of oscillation frequency for groove welding

The graph of Figure 6.18 demonstrates a lower torch oscillation frequency influence in arc peak voltages in groove welding when compared to single sidewall welding. The model is consistent with the data from experimentation. For instance with trial E5 (Figure 5.29 – p. 120) (13.5 mm CTWD), voltage difference between 5 Hz and 25 Hz is 0.36 V whereas the calculated value is 0.79 V. It was then concluded that torch oscillation frequency is not significant for torch position control in GMAW-P due to the relatively small influence in arc signal.

As in single sidewall welds, weld metal penetration decays with the increase of oscillation frequency by the same reasons in both cases: lower localised energy intensity with increase of oscillation frequency due to arc deflections to the sidewall. The lower localised energy intensity is created by less arc time established vertically with the groove bottom. The arc deflects to the sidewall even at considerable distances from it, thus reducing the number of arc pulses established vertically. For instance in trial E5.1 (proximity 0 mm, oscillation frequency 5 Hz, groove width 3.5 mm), although one torch seam traverse has 18 arc pulses, only two are established vertically with the groove bottom (Figure 6.19). With the increase of oscillation frequency, the amount of vertical established arcs is highly reduced. In trial E5.5 (proximity 0 mm, oscillation frequency 25 Hz, groove width 3.5 mm), a vertical arc is sometimes established with the groove bottom only after two complete oscillations. Sidewall proximity makes the arc to slightly deflect to the sidewall as it is seen in the left and right images from the sequence of Figure 6.19, thus reducing the heat to the groove bottom in the centre of the groove.

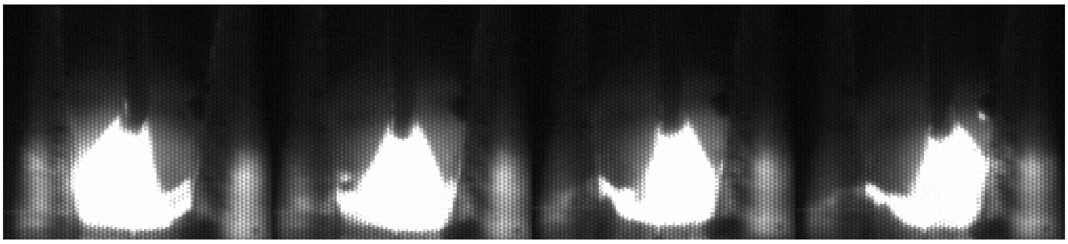


Figure 6.19 – Four consecutive arc pulse images in a sequence for one torch seam traverse of trial E5.1 (proximity 0 mm, oscillation frequency 5 Hz, groove width 3.5 mm)

Sometimes, the arc is established with the sidewall when the wire is still further away reducing even more the number of vertical arcs (Figure 6.20). The reason is the failure of droplet detachment in one arc pulse. Droplet detachment in GMAW-P is theoretically one droplet per pulse. However, there are occasions where the forces in the process do not result in the detachment of the droplet. The wire tip becomes larger with the droplet wobbling around the wire tip as is visible in Figure 6.20 reducing the distance to the sidewall. The arc in next current pulse establishes with the sidewall that burns back the wire increasing the droplet size and forming a big globule. This globule starts then to move down changing the arc's path during a single current pulse. The gravitational force and base metal surface tension becomes the prevalent forces and the globule finally detaches from the wire.

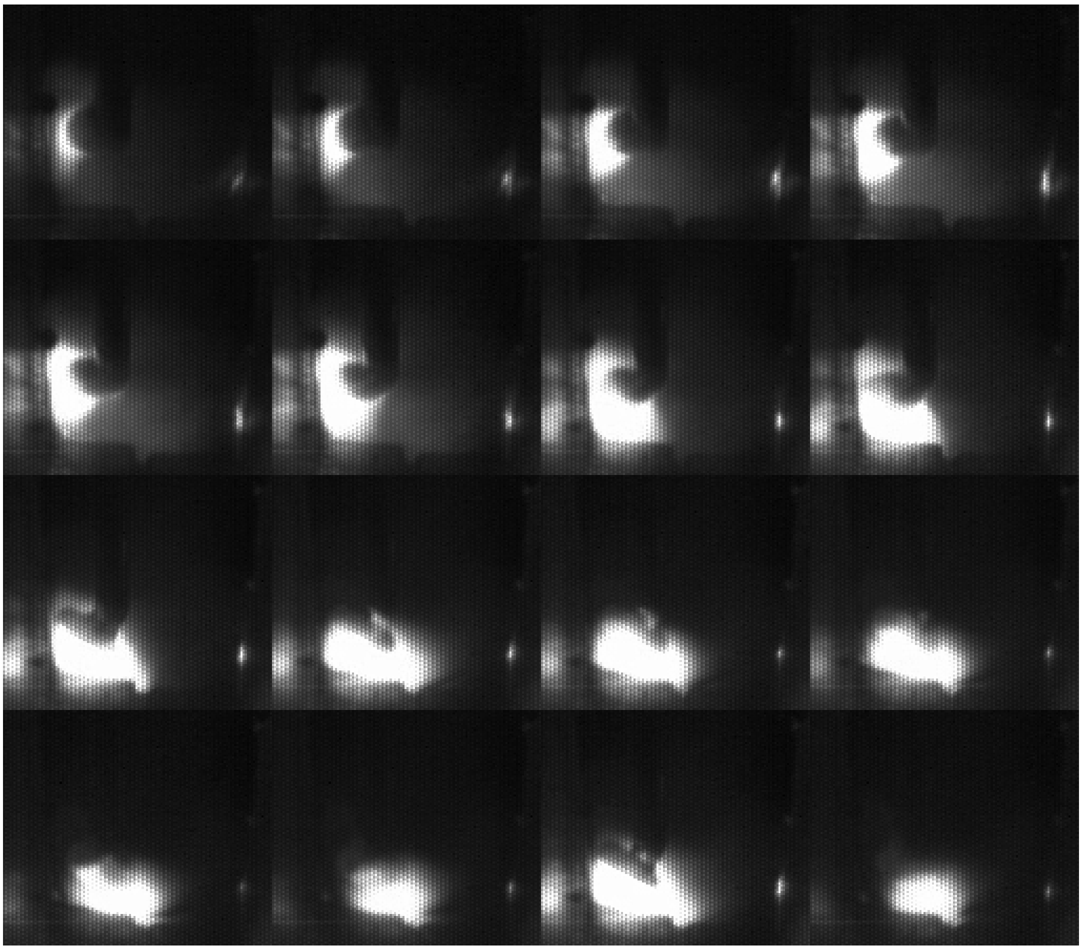


Figure 6.20 – Arc image sequence for trial E6.1 (proximity 0.15 mm, oscillation frequency 5 Hz, groove width 5 mm)

It is also important to note that sidewall weld metal penetration inside the groove varied around the same values as with single sidewall, whereas bottom penetration is visibly lower inside the groove (Figure 5.29 – p. 120 and Figure 5.20 – p. 111 respectively). This is a result of the higher arc lengths found in groove trials compared with single sidewall trials. The presence of the second sidewall produces more burn back and hence the wire does not have time to recover sufficiently between torch excursions, as discussed earlier. Longer arc lengths produce greater radiative heat losses being less efficient in the amount of heat transferred to the base metal [128, 129].

6.4.3.2 Sidewall proximity influence

This analysis has shown a similar and consistent variation as with single sidewall trials. In fact, it confirms that a closer proximity with the sidewall (within positive values) benefits all aspects such as:

a) Increase of bottom weld metal penetration
b) Increase of side weld metal penetration
c) Increase of arc peak voltage difference between sidewall and groove centre improving control sensitivity
d) Less corner lack-of-sidewall fusion defects
e) Less undercut defects

Peak voltages at 0 mm proximity have similar values for both groove and single sidewall trials. For torch oscillation width control development, the reference value found was 2.8 V ±0.2 V of voltage difference between peak voltage at T@C and T@S. In fact, this value is similar to the value obtained in the initial trials as shown in the graph of Figure 5.5 (p. 99) at 5 mm oscillation width.

For weld metal penetration at 0 mm proximity, the similarity with single sidewall trials is only in sidewall penetration. For bottom weld metal penetration the difference is approximately 30% less penetration for groove trials. It is the same order of penetration difference found with oscillation frequency influence analysis. The reasons are the same presented earlier: the number of vertical arc pulses established to the bottom workpiece is less for groove than for single sidewall welds because the arc is more deflected to the sidewalls.

An important observation was found at low sidewall proximities as shown in Figure 6.21 from trial E6.1. With the arc establishing with the sidewall, the wire tends to melt just in the side of the sidewall creating a pointy shape as is visible in the images. In this case, when the torch moves away from the sidewall the arc rapidly establishes with the groove bottom due to the lower distance with it, not performing the normal cycle of l_t as described before.

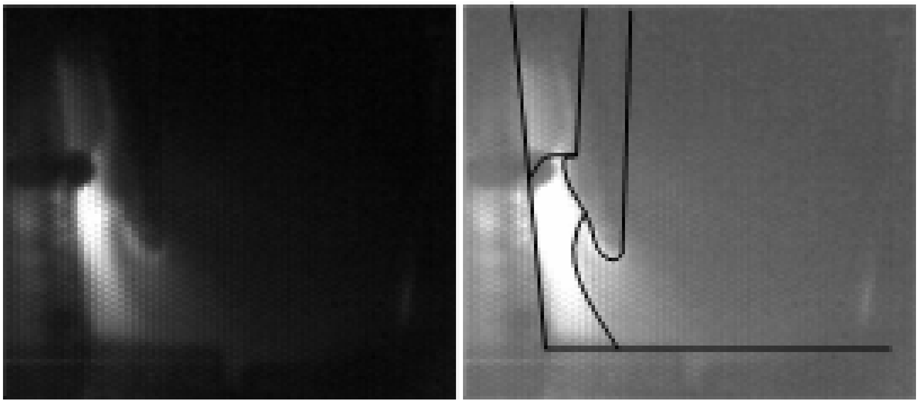


Figure 6.21 – Wire tip melting process when near the sidewall from trial E6.1 (proximity +0.15 mm, oscillation frequency 5 Hz, groove width 5 mm)

It was concluded from groove trials that torch oscillation frequency produced no significant influence in peak voltage sensitivity as confirmed by the developed model. It was also concluded that 0 mm proximity should be used to perform good depth of weld metal penetration and to avoid lack-of-sidewall fusion and undercut defects. At this sidewall proximity, the voltage difference for torch oscillation width control between peak voltages at T@C and T@S has a value of 2.8 V ±0.2 V.

6.4.4 Torch oscillation width and frequency summary

In conclusion, experimentation phase 3 supplied key data for the development of torch oscillation width control. The initial hypothesis was based on the premise of peak voltages differential between T@C and T@S indicating groove width changes. The analysis has shown that this voltage difference is 2.8 V with good weld metal penetrations with no visible defects. This voltage reflects a 0 mm wire proximity to the sidewall. The analysis also revealed that torch oscillation frequency variation in the range of 5 Hz to 25 Hz in GMAW-P has produced little influence in peak voltage difference between T@C and T@S. Also, the value of 14 mm CTWD (peak voltage of 28.96 V) was revealed by the analysis as showing the best fusion results from the tested range of 13 mm to 17 mm. This voltage value should be used improving the developed CTWD control.

6.5 Summary

Through-the-arc sensing in the 5º narrow groove pipe welding using GMAW-P was the main objective of this work. Arc behaviour inside the narrow groove was studied by high speed imaging observations of the arc synchronised with arc signals. Specific analysis software was developed for this purpose.

The GMAW-P power supplies used for this work are true constant current and thus arc length changes are only reflected in voltage values. The use of through-the-arc sensing for pipe welding applications poses specific challenges due to the steep groove sidewalls and the use of short arc lengths, producing very different behaviour compared to V-groove arc sensing techniques. Tandem welding is also quite different from single wire techniques with both wires concurrently share a close arc producing mutual interferences in arc signals.

The use of GMAW-P in pipe welding applications is performed with short arc lengths to achieve good weld profiles in all positions around the pipe. This technique creates frequent short-circuiting and thus influencing the arc voltage signal. An in depth analysis of this signal revealed that short-circuiting produces more interference during the background period of the pulse and hence to voltage average calculation. Different approaches were evaluated to determine ideal values for the average window size. From this analysis it was possible to conclude that voltages at peak current (peak voltage) were found to have higher sensitivity and produce larger variations, less noise and more consistent values for torch position control.

According to the wire proximity to the sidewall analysis performed in this work, optimum proximities were found between 0 mm and +0.2 mm measured from the outer edge of the wire to the sidewall corner. Accurate control is required since +1 mm proximity produced poor sidewall fusion and no signal differentiation for control recognition of groove width. This work showed that negative proximities or wire proximity beyond the sidewall produce wire burn back and hence very long arc lengths, resulting in poor depths of penetration and shallower beads, with major undercut defects.

As a result of this investigation, a CTWD and cross-seam control system was developed and tested for single and tandem GMAW-P, inside the 5º narrow groove. The test results were better than expected revealing accuracies for both controls below 0.2 mm. CTWD control was developed by following the existent welding procedure voltage average and cross-seam control by peak voltage comparison between maximum torch excursions. It is however necessary in the development of future welding procedures to incorporate the peak voltage values obtained at the torch centre of oscillation and at the sidewall, for further improvements in the system accuracy.

In addition, this work has also shown the importance of torch oscillation width control, in order to produce accurate cross-seam control. A method was proposed to achieve torch oscillation width control by a continuous peak voltage comparison between centre and sidewall torch positions. This control will also provide a clear indication of actual groove width. Clearly this data can also be used to implement a system which adapts welding parameters to groove width.

Torch oscillation frequency was also analysed in order to understand its influence in peak voltage sensitivity. Some researchers referred to have observed clear voltage sensitivity increase with the increase of oscillation frequency. Similar experimentation in GMAW was conducted and similar results were obtained. However, in GMAW-P no increase in sensitivity was found with the increase of oscillation frequency with the welding setup used. Different recovery speeds and recovery voltages were found at different CTWD values produced by the GMAW-P power supply. It seems possible that the constant rebalance of arc equilibrium produced by variations on CTWD with the constant arc deflections to the sidewall during torch oscillation may be the cause of a small variation in sensitivity.

7 Conclusion and recommendations for further work

7.1 Conclusions

The following conclusions were drawn from this research work.

1. The voltages at peak current were identified as the optimum strategy for through-the-arc control of torch position inside the 5º bevel narrow groove using GMAW-P.
2. Algorithms were developed for CTWD control based on voltage average and cross-seam position control by comparison of voltage at peak current at torch maximum excusions.
3. Tests on the developed system determined that both cross-seam position and CTWD control were achieved with an accuracy better than 0.2 mm.
4. Mechanisms for arc deflections and heat distribution were presented based on a detailed study of arc behaviour in the 5º bevel narrow groove with GMAW-P using a high speed oscillating torch at different oscillation frequencies.
5. Torch weave width must be closely controlled, both to achieve good fusion and to generate sufficiently strong signals for through-the-arc control.
6. Optimum oscillation width in relation to groove width was found with the wire tip distancing the sidewall by 0 mm to 0.2 mm at maximum torch excursions. The wire tip should be measured from the closest side of the sidewall and touching the groove bottom at the intended CTWD.
7. A torch oscillation width control strategy has been proposed based on sustaining a reference voltage value difference between torch at centre of oscillation and torch at maximum excursion.

8. Dynamic behaviour of GMAW-P was analysed and it was found that torch oscillation frequency does not produce the same magnitude of influence for the pulsed voltage signal sensitivity as produced for GMAW.
9. Mathematical models were devised showing the voltage signal sensitivity evolution for the experimental oscillation frequencies.
10. The final system was demonstrated in a circumferential pipe weld, and some of the methods developed have already been adopted by a pipeline contractor.

7.2 Recommendations for further work

The recommendations for further work are:

1. Development and implementation of the torch oscillation width control
   - Development of the control algorithms and integration with the existing VISENSE system, using the proposed control mechanism
2. Through-the-arc sensing of bead shape
   - Bead shape varies at different angles of the pipe. The idea is to sense and correct the welding parameters and torch positioning in order to attain a consistent bead shape along the pipe circumference
3. Adaptive filling
   - The torch positioning control system developed and proposed in this work already permits the system to perform adaptive filling but the whole process needs to be adapted to achieve it
4. Watchdog measures in existing control algorithms
   a. Development of watchdog measures to react to welding instabilities, mechanical failures and program input errors based on a predefined knowledge database, threshold values or any other available mechanisms
5. Inclusion of defect detection algorithms
   a. Some authors already performed through-the-arc sensing for welding errors and defect detection. These algorithms can be incorporated to build a more robust and complete automation system
6. Contact-tip wear control
   a. Through-the-arc sensing of changes in contact-tip performance and development of respective control algorithms

8 References

1. Administration, E.I. *Natural Gas Consumption timeline*. 2003 [cited 12/10/2005]; Available from: http://www.eia.doe.gov/iea/.
2. BS-EN287, *Qualification test of welders — Fusion welding —*. 2004, British Standards Institution.
3. BS-EN1418, *Welding personnel - Approval testing of welding operators for fusion welding and resistance weld setters for fully mechanized and automatic welding of metallic materials*. 1998, British Standards Institution.
4. Lukkari, J., *Light mechanisation. Easy and cost-efficient with ESAB.* Svetsaren - The Esab welding and cutting journal, 2005. **60**(1): p. 7-10.
5. Widgery, D.J., *Mechanised welding of pipelines.* Svetsaren - The Esab welding and cutting journal, 2005. **60**(1): p. 23-26.
6. Berge, J.M., *Automating the Welding Process - successful implementation of automated welding systems*. 1994, New York 10016-4078, USA: Industrial Press Inc. 198.
7. Pan, J.L., *Part 1: Dynamic behaviour of arc welding.* In Book: Arc Welding Control. Publ: Abington, Cambridge CB1 6AH, UK, 2003.
8. Ushio, M. and W. Mao, *Modelling of an arc sensor for d.c. MIG/MAG welding in open arc mode: study of improvement of sensitivity and reliability of arc sensors in GMA welding. First report.* Welding International. Vol., 1996. **10**(8): p. 622-631.
9. Ushio, M. and W. Mao, *An analysis of the dynamic characteristics of an arc sensor for dc MIG/MAG welding in open arc mode: study of improvement of sensitivity and reliability of arc sensors in GMA welding. II.* Welding International. Vol., 1996. **10**(7): p. 543-551.
10. Bingul, Z., G.E. Cook, and A.M. Strauss, *Dynamic model for electrode melting rate in gas metal arc welding process.* Science and Technology of Welding and Joining. Vol., 2001. **6**(1): p. 41-50.
11. Bingul, Z., G.E. Cook, and A.M. Strauss, *Dynamic modelling of electrode melting rate in the GMAW [MIG/MAG welding] process.* NIST [National Institute of Standards and Technology] Special Publication, 1999. **949**: p. 28-30.
12. Bingul, Z., G.E. Cook, and A.M. Strauss, *Comparison of constant current and pulsed gas metal arc welding processes on basis of electrode resistive dissipation.* Science and Technology of Welding & Joining, 2003. **8**: p. 19-28.
13. Bingul, Z., G.E. Cook, and A.M. Strauss. *Dynamic Modeling of Electrode Melting Rate in the GMAW Process*. in *International Conference on Computer Technology in Welding*. 1999. Detroit, MI: [Boulder CO] (325 Broadway Boulder 80303-3328).
14. Bingul, Z. and G.E. Cook, *Dynamic Modeling of GMAW Process.* Ieee International Conference on Robotics and Automation, 1999(4): p. 3059-3064.

15. Suban, M. and J. Tusek, *Dependence of melting rate in MIG/MAG welding on the type of shielding gas used* Journal of Materials Processing Technology, 2001. **119**(1-3): p. 185-192.
16. Adolfsson, S.L., *Automatic quality monitoring in GMA [MIG/MAG] welding using signal processing methods*. 1998, Lund University: Lund, Sweden. p. 269.
17. Norrish, J., *Advanced Welding Processes*, ed. J. Wood and D. Apelian. 1992, Bristol, Philadelphia and New York: Institute of Physics Publishing. 389.
18. Hsu, C., P. Soltis, and J. Hearn, *Optimum arc length and arc voltage in GMAW [MIG/MAG welding].* Trends in Welding Research. Proceedings, 6th International Conference, Pine Mountain, GA, 2002.
19. Pan, J.L., *Part 3: Arc sensors and seam tracking.* In Book: Arc Welding Control. Publ: Abington, Cambridge CB1 6AH, UK, 2003.
20. Philpott, M.L., *Direct arc sensing for robot MIG welding.*
21. Cho, C.H., et al., *Experimental analysis of rotating arc behaviors in a rotary arc gap switch for a 500 kJ capacitor bank.* IEEE Transactions on Magnetics. Vol., 2001. **37**(1): p. 358-361.
22. Liao, B.J., S.D. Wu, and J.L. Pan, *Theoretical model and signal processing of [welding] arc sensor.* Transactions of the China Welding Institution. Vol., 1996. **17**(4): p. 263-271.
23. Kim, C.H. and S.J. Na, *A study of an arc sensor model for gas metal arc welding with rotating arc. Part 2: Simulation of an arc sensor in mechanically rotating gas metal arc welding.* Proceedings of the Institution of Mechanical Engineers. Part B: Journal of Engineering Manufacture. Vol., 2001. **215**: p. 1281-1288.
24. Halmoy, E., *Simulation of rotational arc sensing in gas metal arc metal.* Science and Technology of Welding and Joining. Vol., 1999. **4**(6): p. 347-351.
25. Pan, J.L., *Advance in arc sensing system for automatic seam tracking.* Technology Advancements and New Industrial Applications in Welding, Proceedings, TIWC'98, Taiwan International Welding Conference, Taipei, Taiwan, 1998.
26. Dilthey, U. and J. Gollnick, *Through-the-arc sensing in GMA-welding with high speed rotating torch.* 5th International Conference: Trends in Welding Research, 1998.
27. Sugitani, Y., N. Tamaoki, and M. Murayama, *Development of lattice welding robot with arc sensing control.* The Role of Welding Science and Technology in the 21st Century, Proceedings, 6th JWS International Symposium, 1996. **1**: p. 19-21.
28. Ushio, M., M. Tanaka, and W. Mao, *Sensor for automatic arc welding: advantages and limitations.* Welding Science and Technology, Proceedings, International Welding Conference, Japan Slovak Welding Symposium, Tatranska Lomnica Matliare, Paper, 1996. **2**: p. 5-7.
29. Dilthey, U., L. Stein, and M. Oster, *Through the arc sensing - a universal and multipurpose sensor for arc welding automation.* International Journal for the Joining of Materials. Vol., 1996. **8**(1): p. 6-12.

30. Dilthey, U., L. Stein, and M. Oster, *Through the arc sensing - a multipurpose low-cost sensor for arc welding automation (Lichtbogensensor - ein vielseitiger, preisgunstiger Sensor fur das automatische Lichtbogenschweissen).* DVS Berichte, no, 1995. **170**: p. 13-15.
31. Sugitani, Y. and W. Mao, *Automatic simultaneous control of bead height and back bead shape using an arc sensor in one-sided welding with a backing plate.* Welding International. Vol., 1995. **9**(5): p. 366-374.
32. Ai, Y.Y. and L. Yao, *Arc sensor used in MIG/MAG weld tracking.* Transactions of the China Welding Institution. Vol., 1991. **12**(3): p. 155-160.
33. Kim, C.H., W.S. Yoo, and S.J. Na, *Development of an arc sensor with mechanized rotation of electrode.* Materials Science Forum. Vol., 2003.
34. Wu, S.D., B.J. Liao, and J.L. Pan, *High speed rotating arc sensor.* Transactions of the China Welding Institution. Vol., 1997. **18**(1): p. 61-66.
35. Nomura, H., et al., *Development and application of arc sensor control with a high speed rotating arc process.* Sensors and Control Systems in Arc Welding, Chapter, 1994. **28**: p. 247-256.
36. Sugitani, Y., Y. Nishi, and T. Satoh, *An intelligent arc welding robot with simultaneous control of penetration depth and bead height.* Sensors and Control Systems in Arc Welding, Chapter, 1994. **44**: p. 390-399.
37. Pan, J.L. and B.J. Liao, *Fuzzy control of multi-freedom welding machine.* China Welding. Vol., 1993. **2**(2): p. 125-133.
38. Eichhorn, F. and J. Borowka, *Process-oriented welding head guidance system for gas-shielded metal-arc narrow-gap welding (Prozessorientiertes Schweisskopffuhrungssystem ...).* Schweissen und Schneiden. Vol. 42, 1990(11, pp): p. 564.
39. Sugitani, Y., et al., *Intelligent arc welding robot with simultaneous control of penetration depth and bead height.* NKK Technical Review , no, 1990. **58**: p. 93-99.
40. Sugitani, Y., et al., *Development of multi-electrodes automatic fillet welding equipment with high speed rotating arc.* NKK Technical Review , no, 1989. **56**: p. 73-79.
41. Sugitani, Y. and M. Murayama, *Bevel profiling control method for arc welding*. 1989, Nippon Kokan, K. K.: Japan. p. 9.
42. Nomura, H., Y. Sugitani, and M. Murayama, *Development of automatic fillet welding process with high speed rotating arc.* Transactions of the Japan Welding Society. Vol., 1987. **18**(2): p. 116-124.
43. Nomura, H. and Y. Sugitani, *Sensing and control of arc welding.* Advances in Welding Science and Technology, Proceedings, International Conference, Gatlinburg, TN, USA, 1986.
44. Nomura, H. and Y. Sugitani, *Narrow gap MIG welding process with high speed rotating arc.* Quality and Reliability in Welding, Proceedings, International Conference, Hangzhou, 1984.
45. Sugitani, Y. and Y. Kobayashi, *Rotary arc-welding apparatus*. 1985, Nippon Kokan, K. K.: Japan. p. 8.
46. Sugitani, Y., M. Murayama, and K. Yamashita, *Development of articulated arc welding robot with high speed rotating arc process.*

Advanced Technology in Welding, Materials Processing and Evaluation, Proceedings, 5th JWS International Symposium, Tokyo, Vol.I, Paper III, 1990. **14**: p. 17-19.
47. Nomura, H., *Recent advances in GMAW [gas metal arc welding] processes in Japan.* Advanced Technology in Welding, Materials Processing and Evaluation, Proceedings, 5th JWS International Symposium, Tokyo, Vol.I, 1990.
48. Sugitani, Y., et al., *Development of articulated arc welding robot with high speed rotating arc process.* NKK Technical Review , no, 1989. **57**: p. 111-117.
49. Nomura, H. and Y. Sugitani, *Further improvement of narrow gap welding techniques.* Joining of Metals, 1984. **2**: p. 15-18.
50. Nomura, H., Y. Sugitani, and Y. Kobayashi, *Narrow gap welding process with high speed rotating arc*, in *Narrow Gap Welding (NGW), The State-of-the-Art in Japan*, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 74-80.
51. Nakajima, H., A. Nagai, and S. Minehisa, *Rotating arc narrow gap MIG welding process*, in *Narrow Gap Welding (NGW), The State-of-the-Art in Japan*, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 65-73.
52. Jeong, S.-K., et al. *Development of high speed rotating arc sensor and seam tracking controller for welding robots*. in *IEEE International Symposium on Industrial Electronics*. 2001.
53. Yoo, W.S., C.H. Kim, and S.J. Na. *Detection of end point of fillet joint by using mechanised rotating arc sensor*. in *Computer Technology in Welding and Manufacturing. Proceedings, 14th International Conference*. 2004. Sheffield, UK: Cambridge, CB1 6AI, UK; TWI Ltd; 2004.
54. Norrish, J. and J. Nixon, *A history of pulsed MIG welding.* Joining and Materials, 1989. **2**(6): p. 264-266, 268.
55. Ma, J. and R.L. Apps, *Analysing metal transfer during MIG welding.* Welding and Metal Fabrication, 1983. **51**(3): p. 119-122, 124-126, 128.
56. Amin, M. and P.V.C. Watkins, *Synergic Pulse MIG Welding*. 1977, TWI.
57. Quintino, L., *Fusion characteristics in P-GMAW of mild steel*, in *WERC*. 1986, Cranfield University: Bedford, UK.
58. Kenney, K.L., K.S. Miller, and H.B. Smartt. *Heat transfer in pulsed gas metal arc welding*. in *Trends in Welding Research, Proceedings, 5th International Conference*. 1998. Pine Mountain, GA.
59. Di Pietro, P., et al. *Through arc adaptive control of GMAW: Requirements for high productivity girth welding of pipe*. in *2nd International Conference on Recent Developments and Future Trends in Welding Technology*. 2003. Cranfield University, UK.
60. Barnett, R.J., et al., *Through-the-arc sensing and control in pulsed gas metal arc welding.* 5th International Conference: Trends in Welding Research, 1998.
61. Cook, G.E., et al., *Electric arc sensing for robot positioning control.* Robot Welding,, 1987: p. 181-216.
62. Laing, B., *Enhanced narrow gap pipeline welding using computer control and through-the-arc tracking technology.* Pipeline Reliability,

Proceedings, International Conference, Calgary, Canada, Vol, 1992. **2**(14): p. 2-5.
63. Rajagopalan, T.S. and P.A. Tews, *Through the arc tracking of 5G narrow gap pipe welds.* Eleventh International Conference on Computer Technology in Welding, 2001.
64. Cook, G.E., *System considerations in welding automation : impact on the production chain.* Automation and Robotics for Welding, Proceedings, Conference, Indianapolis, 1983.
65. King, F.J. and P. Hirsch. *Seam tracking systems with the arc as sensor*. in *'Advances in welding processes', proceedings, 4th international conference*. 1978. Harrogate, Yorks: Abington Hall, Abington, Cambridge CB1 6AL; The Welding Institute.
66. Jieyu, J.W., et al., *Development of a Closed-loop Through-the-Arc-Sensing Controller for Seam Tracking in Gas Metal Arc Welding*. 2000, The Welding Institute: Cambridge, UK.
67. Suzuki, Y., *Method and apparatus for determining seam tracking control of arc welding*. 2002, Central Motor Wheel Co., Ltd.: Japan. p. 15.
68. Sugitani, Y. and N. Tamaoki, *Automatic groove tracing control method for arc welding*. 1987, Nippon Kokan, K. K.: Japan. p. 8.
69. Webster, J.G., *The Handbook of Measurement, Instrumentation, and Sensors*. 1998: CRC Press. 2608.
70. Wang, J. and A. Kusiak, *Computational intelligence in manufacturing handbook*. 2000, Boca Raton - Florida - USA: CRC Press LLC. 576.
71. Murakami, S., et al., *Weld-line tracking control of arc welding robot using fuzzy logic controller.* Fuzzy Sets and Systems, 1989. **32**(2): p. 221.
72. Fujimura, H., E. Ide, and S. Murakami, *Arc sensing using fuzzy control.* Sensors and Control Systems in Arc Welding, Chapter, 1994. **27**: p. 238-246.
73. Kim, Y.J. and S.H. Rhee, *Development of an arc sensor model using a fuzzy controller in gas metal arc welding.* Measurement Science and Technolology. Vol., 2001. **12**: p. 534-541.
74. Kim, J.W. and S.J. Na, *A self-organising fuzzy control approach to arc sensor for weld joint tracking in gas metal arc welding of butt joints.* Welding Journal. Vol., 1993. **72**(2).
75. Eguchi, K., et al., *Application of neural network to arc sensor.* Science and Technology of Welding and Joining. Vol., 1999. **4**(6): p. 327-334.
76. Wang, J.Y., Y.Q. Le, and H.H. Pan, *A welding joint tracking detection algorithm for arc sensor based on neuro clustering.* International Journal for the Joining of Materials. Vol., 1999. **11**(2): p. 42-48.
77. Ohshima, K., et al., *Controlling of torch attitude and seam tracking using neuro arc sensor.* The, 1995. **2**(of 2): p. 06-10.
78. Wells, A.M.J., *Through-the-arc sensing and control methods in robotic arc welding*. 1988, Vanderbilt University: Nashville, TN 37240, USA. p. 202.
79. Halmoy, E. *Wire melting rate, droplet temperature and effective anode heating potential*. in *Arc Physics and Weld Pool Behaviour*. 1980. London: The Welding Institute, Cambridge.

80. Lesnewich, A., *Control of Melting Rate and Metal Transfer in Gas Shielded Metal-Arc Welding: Part I – Control of Electrode Melting Rate.* Welding Journal, 1958. **37**(8): p. pp. 343s-353s.
81. Dilthey, U. and J. Gollnick, *Keynote address - through-the-arc-sensing in GMA-welding with high speed rotating torch.* Trends in Welding Research, Proceedings, 5th International Conference, Pine Mountain, GA, 1998.
82. Nomura, H., Y. Sugitani, and M. Murayama, *Development of automatic fillet welding process with high speed rotating arc.* Quarterly Journal of the Japan Welding Society. Vol., 1986. **4**(3): p. 502-507.
83. Pan, J., *Arc sensing system for automatic weld seam tracking. II - Signal processing.* Science in China. Series E: Technological Sciences. Vol., 2001. **44**(4): p. 389-397.
84. Cook, G.E. and M. Wells, *Sidewall-matching adaptive control system for welding*. 1984, CRC Welding Systems, Inc.: USA. p. 34.
85. Rashid, K., et al., *Seam tracking control by fuzzy logic in pulsed gas metal arc welding.* 5th International Conference: Trends in Welding Research, 1998.
86. Laing, B., *A digital revolution.* World Pipelines, 2006: p. 17-18.
87. Halmoy, E., H. Olsen, and S. Samuelsen, *Adaptively controlled MIG narrow gap welding.* Developments and Innovations for Improved Welding Production, Proceedings, 1st International Conference, Birmingham, England, Session, 1983. **5**(8): p. 13-15.
88. Kang, Y.H. and S.J. Na, *Characteristics of welding and arc signal in narrow groove [narrow gap] gas metal arc welding using electromagnetic arc oscillation.* Welding Journal, vol, 2003. **82**(5).
89. Araya, T., et al., *General review - The state-of-the-art of narrow gap welding (NGW) in Japan*, in *Narrow Gap Welding (NGW), The State-of-the-Art in Japan*, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 13-33.
90. Modenesi, P.J., *Statistical modelling of the narrow gap gas metal arc welding process*. 1990, Cranfield University: Cranfield, Bedfordshire, MK43 OAL, UK. p. 336.
91. Nakamura, T. and K. Hiraoka, *Ultranarrow GMAW [MIG/MAG welding] process with newly developed wire melting control system.* Science and Technology of Welding and Joining. Vol., 2001. **6**(6): p. 355-362.
92. Lassaline, E., B. Zajaczkowski, and T.H. North, *Narrow groove twin-wire GMAW [gas metal arc welding] of high-strength steel.* Welding Journal, 1989. **68**(9): p. 53-58.
93. OKUDA, N., T. KASHIMURA, and H. SAITA, *Narrow gap GMA welding process "twist arc welding process"*, in *Narrow Gap Welding (NGW), The State-of-the-Art in Japan*, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 46-55.
94. Kawahara, M. and I. Asano, *BHK type narrow gap GMA welding process*, in *Narrow Gap Welding (NGW), The State-of-the-Art in Japan*, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 39-45.
95. Shono, S., O. Matsumoto, and H. Kamei, *Development and application of narrow-gap GMA welding process with corrugated wire*, in *Narrow Gap*

Welding (NGW), The State-of-the-Art in Japan, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 54-59.

96. Kanbe, Y. and K. Suda, *Narrow gap welding process with oscillating arc "Loopnap"*, in *Narrow Gap Welding (NGW), The State-of-the-Art in Japan*, F.M.e. al, Editor. 1986, Japan Welding Society: Tokyo, Japan. p. 60-64.
97. Baba, N., et al., *Development of intelligent machine for all-positional MAG welding.* Advanced Technology in Welding, Materials Processing and Evaluation, Proceedings, 5th JWS International Symposium, Tokyo, Vol.I, Paper III, 1990. **10**: p. 17-19.
98. Fujimura, H., E. Ide, and H. Inoue, *Estimation of Contact Tip--Workpiece Distance in GMA Welding.* Quarterly Journal of the Japan Welding Society. Vol., 1987. **5**(3): p. 25-30.
99. Ponomarev, V., A.V. Costa, and A. Scotti, *Evaluation of the voltage drop in electrode metal droplets under MIG/MAG welding conditions.* Jom, 2003. **11**(11): p. 25-28.
100. Yamamoto, H., et al., *Estimation of arc length and wire extension using neural network.* Welding International. Vol., 2003. **17**(5): p. 357-364.
101. Di Pietro, P., J. Norrish, and S. Trenholme, *Active through-arc control of stick-out in mechanised girth welding systems.* WTIA Technology Week. Proceedings, 50th WTIA Annual Conference and the 12th International TWI Computer Technology in Welding and Manufacturing Conference, 2002.
102. Sugitani, Y., *High speed rotary automatic arc welding technology.* Kinzoku, 1995. **65**(10): p. 23-34.
103. Matsuda, F., et al., *Application of fast weaving to CO2 arc welding. Report 1: On bead appearance and spattering loss.* Transactions of JWRI. Vol., 1979. **8**(1): p. 13-19.
104. Sugitani, Y., W. Mao, and M. Ushio. *Adaptive Control of Weld Bead Shape Utilizing Arc Sensor in One Side GMAW Process with Backing Plate*. in *Proc. of IIW Commission XII, Doc.XII-1360-94* 1994.
105. Kodama, M., H. Goda, and H. Iwabuchi, *Development of high-frequency oscillating arc. Report 2: Arc sensor for simultaneous detection of torch aiming deviation and gap width.* Welding International. Vol., 2001. **15**(12): p. 952-964.
106. Kang, Y.H. and S.J. Na, *A study on narrow gap gas metal arc welding using electromagnetic arc oscillation.* Gas Metal Arc Welding for the 21st Century, Proceedings, Conference, Orlando, FL, 2000.
107. Oshima, S., *A process and an apparatus for arc welding with high-speed weaving*, in *UK patent office*. 1978, Matsumoto Kikai Co Ltd.: Japan. p. 15.
108. Hsu, C., *Wave designer: pulsed GMAW online waveform editor and soft oscilloscope.* Ninth International Conference on Computer Technology in Welding, 1999.
109. Freescale Semiconductor Co, *Metrowerks CodeWarrior - Version 4.2.6.922* 2005, http://www.metrowerks.com.
110. Microsoft Corporation, *Visual Basic for Windows - Version 6.0*. 2001, http://msdn.microsoft.com/vbrun/.

111. Motorola, S.H.K.L., *DSP56F801/803/805/807 - 16-Bit Digital Signal Processor - User's Manual*, in *DSP56F801-7UM/D - Rev. 3.0*. 2001. p. 782.
112. Motorola, S.H.K.L., *DSP56800 - 16-Bit Digital Signal Processor - Family Manual*, in *DSP56800FM/D Rev. 2.0, 05/2002*. 2002. p. 444.
113. Vector Informatik GmbH, *CAN Driver Library - User Interface Description for Visual Basic - Version 3.4*. 2002. p. 24.
114. Bosch, *CAN Specification - Version 2.0*. 1991: Robert Bosch GmbH, Postfach 50, D-7000 Stuttgard 1. p. 72.
115. Ponomarev, V., A.V. Costa, and A. Scott, *Voltage drop in electrode molten metal droplets under MIG/MAG welding conditions.* International Journal for the Joining of Materials. Vol., 2004. **16**(2): p. 49-54.
116. Siewert, T.A., R.B. Madigan, and T.P. Quinn, *Sensors control gas metal arc welding.* Advanced Materials and Processes. Vol., 1997. **151**(4): p. 23-25.
117. Kisilevskii, F.N. and V.V. Dolinenko, *Adjustment of model of arc sensor for adaptive welding robot.* Weld Quality The Role of Computers, Proceedings, International Conference on Improved Weldment Control with Special Reference to Computer Technology, Vienna, Austria, 1988.
118. *National Instruments Corporation*. [Web] 2005 [cited 10/10/2005]; Available from: http://www.ni.com/.
119. Huang, T.C. and D. Yapp. *Synchronised data acquisition and video imaging of metal transfer in gas metal arc welding*. in *Proceeding of Commission XII, IIW, Ljubljana, IIW Doc. XII-1672-01*. 2001.
120. Wikipedia. *Common User Access - CUA*. 2005 [cited 11/05/2005]; Available from: http://en.wikipedia.org/wiki/Common_User_Access.
121. Djuric, P.M. and S.M. Kay, *Spectrum Estimation and Modeling*, in *Digital Signal Processing Handbook*. 1999, CRC Press LLC. p. 22.
122. Tallarida, R.J., *Financial Mathematics*, in *Pocket Book of Integrals and Mathematical Formulas*. 1999, CHAPMAN & HALL/CRC: Boca Raton London New York Washington, D.C. p. 35.
123. DataQ Instruments. *A Closer Look At The Advanced CODAS Moving Average Algorithm*. Sept 2005 [cited; Available from: http://www.dataq.com/support/documentation/pdf/article_pdfs/an14.pdf.
124. Bould, M., et al., *Digital image analysis: improving accuracy and reproducibility of radiographic measurement.* Clinical Biomechanics, 1999. **14**(6): p. 434.
125. Nagesh, D.S. and G.L. Datta, *Prediction of weld bead geometry and penetration in shielded metal-arc welding using artificial neural networks.* Journal of Materials Processing Technology, 2002. **123**(2): p. 303.
126. Liratzis, H., *Different bead shapes at different angles of girth welding*. 2006: PhD research work - WERC - Cranfield University.
127. Sun, J.S. and C.S. Wu, *Effects of welding heat input on microstructure and hardness in heat-affected zone of HQ130 steel.* Modelling and Simulation in Materials Science and Engineering, 2001. **9**(1): p. 25-36.
128. Tsai, N.S. and T.W. Eagar. *Changes of weld pool shape by variations in the distribution of heat source in arc welding*. in *Proceedings of Modelling*

of Casting and Welding Processes II. 1984. Engineering Foundation Conference, Henniker, New Hampshire, USA.
129. Eagar, T.W. and N.S. Tsai, *Temperature fields produced by travelling distributed heat sources.* Welding Journal, 1983. **62**(12): p. 346s-355s.

Appendix A. Algorithms of the 1st phase of experimentation

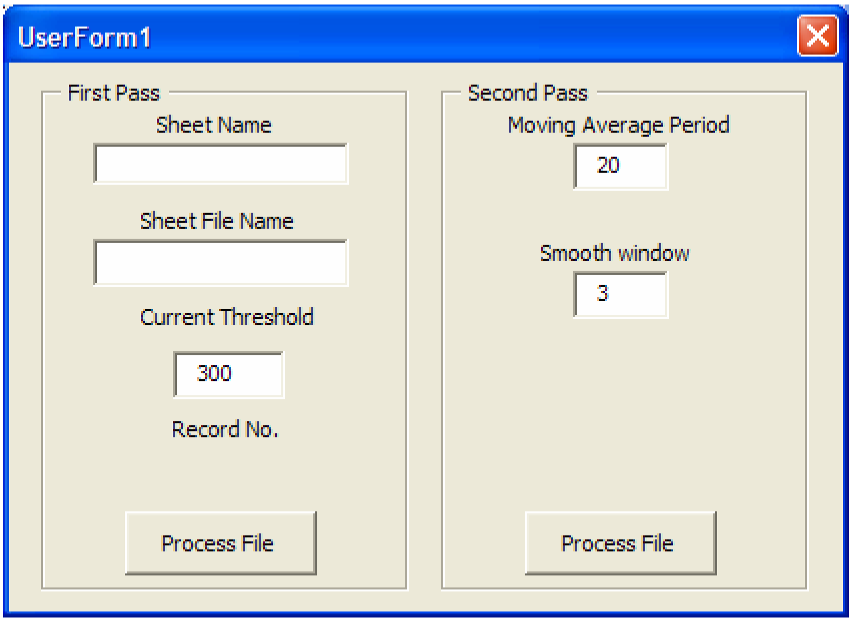


Figure A.1 – Excel macro form and source code of the 1st phase of experimentation

```
Public SheetFilename, OutputFilename As String
Public RecNo As Long

Private Sub CommandButton1_Click()

    StartLine = Chr(10) + Chr(10)
    StartFlag = False
    ArrayIndex = 0
    ArraySum = 0
    MovAverage = 0
    MinValue = 100
    MaxValue = 0
    UpDown = 0
    ZeroVoltage = False
    HighCurrent = False
    MaxCellCount = 1
    MinCellCount = 1
    OldCurrent = 0
    OldVoltage = 0
    RecNo = 0
    TimeCounter = 0
    TimerFlag = False

    Open ActiveWorkbook.Path & "\" & SheetFilename For Input As #1
    'Open ActiveWorkbook.Path & "\" & OutputFilename For Output As #2
```

```
While Not EOF(1) And ZeroVoltage = False
    Input #1, LineRead
    If Left(LineRead, 2) = StartLine Then
        StartFlag = True
        LineRead = Mid(LineRead, 2)
    End If
    If StartFlag Then
        If Left(LineRead, 1) = vbLf Then
            NewVoltage = Val(Mid(LineRead, 2))
            If NewVoltage < 1# And HighCurrent = True Then
                ZeroVoltage = True
            End If
        Else
            NewCurrent = Val(LineRead)
            If NewCurrent > Val(TextBox6.Text) Then
                HighCurrent = True
                If OldCurrent <> 0 Then
                    Idif = OldCurrent - NewCurrent
                    If Idif < 0 And UpDown = 0 Then
                        UpDown = 1
                    End If
                    If UpDown = 1 Then
                        ValArray(ArrayIndex) = NewVoltage
                        ArrayIndex = ArrayIndex + 1
                    End If
                    If Idif > 0 And UpDown = 1 Then
                        MaxValue = ValArray(0)
                        For i = 1 To ArrayIndex - 1
                            If ValArray(i) > MaxValue Then
                                MaxValue = ValArray(i)
                            End If
                        Next i
                        CellLabel = "A" + Trim(Str(MaxCellCount))
                        Range(CellLabel).Value = MaxValue
                        CellLabel = "H" + Trim(Str(MaxCellCount))
                        Range(CellLabel).Value = TimeCounter
                        MaxCellCount = MaxCellCount + 1
                        UpDown = 0
                        ArrayIndex = 0
                        TimerFlag = True
                    End If
                End If
            End If
            OldCurrent = NewCurrent
            If TimerFlag Then
                TimeCounter = TimeCounter + 0.1
            End If
        End If
```

```
            OldVoltage = NewVoltage
            DoEvents
            Label9.Caption = Str(RecNo) & " / " & MaxCellCount
            RecNo = RecNo + 1
        End If
    Wend
    Close #1
    MsgBox ("Ended")
End Sub


Private Sub CommandButton2_Click()

    ArrayIndex = 0
    ArraySum = 0
    MovAverage = 0
    MinValue = 100
    MaxValue = 0
    MinPos = 0
    MaxPos = 0
    UpDown = 0
    MaxCellCount = 1
    MinCellCount = 1
    CellCount = 1
    toggle = True

    For i = 1 To Val(TextBox7.Text)
        CellLabel = "A" + Trim(Str(i))
        CellValue = Range(CellLabel).Value
        MovAvgSum = MovAvgSum + CellValue
    Next i
    MovAverage = MovAvgSum / Val(TextBox7.Text)

    Do
        'CellLabel = "B" + Trim(Str(CellCount))
        'Range(CellLabel).Value = 0
        'CellLabel = "C" + Trim(Str(CellCount))
        'Range(CellLabel).Value = 0
        CellLabel = "A" + Trim(Str(CellCount))
        CellValue = Val(Range(CellLabel).Value)
        If CellValue <> 0 Then
            ValArray(ArrayIndex) = CellValue
            ArrayIndex = ArrayIndex + 1
            If ArrayIndex = Val(TextBox5.Text) Then
                ArrayIndex = 0
            End If
            ArraySum = 0
            If CellCount >= Val(TextBox5.Text) Then
```

```
        For i = 0 To Val(TextBox5.Text) - 1
            ArraySum = ArraySum + ValArray(i)
        Next i
        CellAvgValue = ArraySum / Val(TextBox5.Text)
    Else
        CellAvgValue = CellValue
    End If
    MovAvgSum = MovAvgSum + CellValue - MovAverage
    MovAverage = MovAvgSum / Val(TextBox7.Text)

    If CellAvgValue > MovAverage And UpDown = 0 Then
        UpDown = 1
    End If
    If CellAvgValue < MovAverage And UpDown = 0 Then
        UpDown = 2
    End If

    If CellAvgValue > MovAverage And CellAvgValue > MaxValue Then
        MaxValue = CellAvgValue
        MaxOrigValue = CellValue
        MaxPos = CellCount
    End If

    If CellAvgValue < MovAverage And CellAvgValue < MinValue Then
        MinValue = CellAvgValue
        MinOrigValue = CellValue
        MinPos = CellCount
    End If

    If CellAvgValue > MovAverage And UpDown = 2 Then
        If toggle = True Then
            CellLabelOut = "F" + Trim(Str(MinPos)) ' MinLeft Line and Point
            Range(CellLabelOut).Value = MinOrigValue
            CellLabelOut = "C" + Trim(Str(MinPos))
            Range(CellLabelOut).Value = MinOrigValue
            If MinPos > 1 Then
                BackCount = MinPos
                Do
                BackCount = BackCount - 1
                If BackCount > 0 Then
                    CellLabelOut = "C" + Trim(Str(BackCount))
                    CellNewValue = Range(CellLabelOut).Value
                End If
                Loop While (BackCount > 0) And (CellNewValue = 0)
                If BackCount > 0 Then
                    CellInc = (MinOrigValue - Val(CellNewValue)) / (MinPos -
BackCount)
                    For i = BackCount + 1 To MinPos - 1
```

```
                CellNewValue = CellNewValue + CellInc
                CellLabelOut = "C" + Trim(Str(i))
                Range(CellLabelOut).Value = CellNewValue
            Next i
        End If
    End If
    toggle = False
Else
    CellLabelOut = "G" + Trim(Str(MinPos)) ' MinRight Line and Point
    Range(CellLabelOut).Value = MinOrigValue
    CellLabelOut = "D" + Trim(Str(MinPos))
    Range(CellLabelOut).Value = MinOrigValue
    If MinPos > 1 Then
        BackCount = MinPos
        Do
        BackCount = BackCount - 1
        If BackCount > 0 Then
            CellLabelOut = "D" + Trim(Str(BackCount))
            CellNewValue = Range(CellLabelOut).Value
        End If
        Loop While (BackCount > 0) And (CellNewValue = 0)
        If BackCount > 0 Then
            CellInc = (MinOrigValue - Val(CellNewValue)) / (MinPos -
BackCount)
            For i = BackCount + 1 To MinPos - 1
                CellNewValue = CellNewValue + CellInc
                CellLabelOut = "D" + Trim(Str(i))
                Range(CellLabelOut).Value = CellNewValue
            Next i
        End If
    End If
    toggle = True
End If

'CellLabelOut = "E" + Trim(Str(MinCellCount))
'Range(CellLabelOut).Value = MinOrigValue

MinCellCount = MinCellCount + 1
MinValue = 100
UpDown = 1
End If

If CellAvgValue < MovAverage And UpDown = 1 Then
    CellLabelOut = "E" + Trim(Str(MaxPos)) ' MaxLine and Point
    Range(CellLabelOut).Value = MaxOrigValue
    CellLabelOut = "B" + Trim(Str(MaxPos))
    Range(CellLabelOut).Value = MaxOrigValue
    If MaxPos > 1 Then
```

```
                BackCount = MaxPos
                Do
                BackCount = BackCount - 1
                If BackCount > 0 Then
                   CellLabelOut = "B" + Trim(Str(BackCount))
                   CellNewValue = Range(CellLabelOut).Value
                End If
                Loop While (BackCount > 0) And (CellNewValue = 0)
                If BackCount > 0 Then
                   CellInc = (MaxOrigValue - Val(CellNewValue)) / (MaxPos -
BackCount)
                   For i = BackCount + 1 To MaxPos - 1
                      CellNewValue = CellNewValue + CellInc
                      CellLabelOut = "B" + Trim(Str(i))
                      Range(CellLabelOut).Value = CellNewValue
                   Next i
                End If
             End If
             'CellLabelOut = "D" + Trim(Str(MaxCellCount))
             'Range(CellLabelOut).Value = MaxOrigValue

             MaxCellCount = MaxCellCount + 1
             MaxValue = 0
             UpDown = 2
          End If
          'CellLabelOut = "F" + Trim(Str(CellCount))
          'Range(CellLabelOut).Value = CellAvgValue
          'CellLabelOut = "G" + Trim(Str(CellCount))
          'Range(CellLabelOut).Value = MovAverage
          CellCount = CellCount + 1
       End If
   Loop While CellValue <> 0
   MsgBox ("Ended")
End Sub
```

Appendix B. Results of 1st phase of experimentation

Table B.1 - Trial table description

Trial	Oscillation Frequency (Hz)	Oscillation Width (mm)
A1	3.33	5
A2	3.33	6
A3	3.33	4
A4	5.0	5
A5	1.67	5
A6	0	0

Table B.2 - Welding Setup

Power Supply	Lincoln Power Wave F355i
Wire Feeder	Lincoln Power Feed 10 Robotic
Gas	BOC Trimix (5% He, 12.5% CO2, Argon)
Welding Head	RMS MOW II
Signal Acquisition	Yokogawa Oscilloscope ScopeCorder DL750
Current Probe	LEM PR1030 Hall-effect
Wire	Carbofil NIMO 1 (1mm)

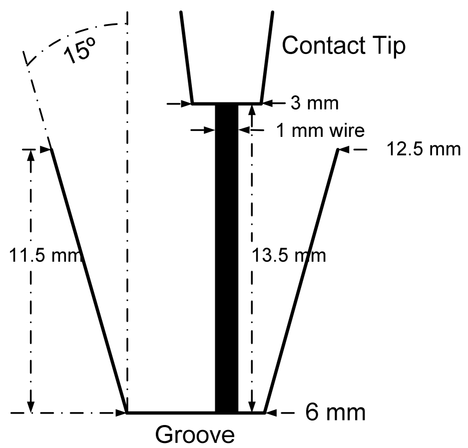


Figure B.1 – Trials A1 to A6 groove preparation

Table B.3 - Pulse parameters exported from WaveDesigner

Wave Designer 2000 - TextExport -

Wed Apr 07 11:54:51 BST 2004 WireFeed inch / min 300

wfscf	0.394
Peak_Amps	350
Peak_Duration	1.7
Background_Amps	50
Background_Duration	3.8
Adapt	0
Reference_Voltage	32
Ramp_up_rate	394
Ramp_overshoot	4
Peak_Current	350
Peak_Time	1.7
Tailout_time	2
Tailout_Speed	0.2
Step_off_Current	50
Background_Current	50
Background_Time	1.8
Frequency	181.8
Adaptive	0
Amp_Sec	1.21
Open_Circuit_Voltage	48
Strike_Current	450
Minimum_Strike_Time	2.5
Starting_Voltage	23
Starting_Current	180
Starting_Time	0
Short_Detect_Volt	5
Pinch_Current_Rise_Rate	55
Arc_Reestablish_Volt	15
Shorting_mode	0
Adaptive_Loop	0
Peak_Voltage	32
Peak_Amps_SF	20
Peak_Time_SF	0
Background_Amp_SF	10
Frequency_SF	30
Type_of_Adaptive_Control	1
Set_Voltage	32
End_Amp	550
End_Time	2.5
Vreg_start	23
Vreg_weld	32
stv	23

fpv	32
fpt	1.7
ffr	394
fft	0.8
psf	0.2
ptsf	0
pk	350
pkt	1.7
expsf	0.1
expa	50
expt	2
expspd	0.2
backsf	0.1
persf	-0.3
back	50
per	5.5
wfsa	300
wfs	762
x1	100
y1	175
z1	500
x	1
y	1
z	0
k1	0
k2	0
kl	3073
ks	5461
hs1	0
hs2	0
hs3	0
n1	5
n2	5
pktim	0.9
Weld Process	GMAW
Wire Type	Steel
Wire Size	1.0mm
Process Name	1mm
Procedure	Pulse
Gas	ArCO2

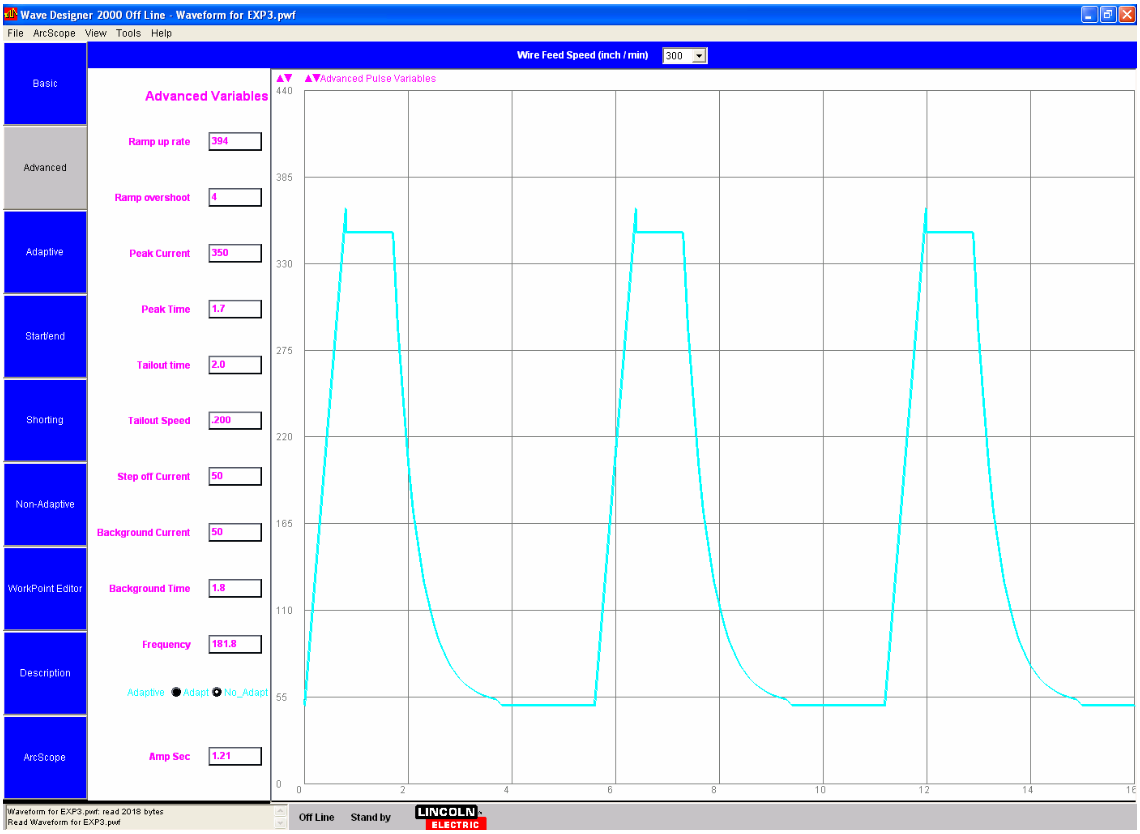


Figure B.2 - Pulse waveform shape used for trials A1 to A6 - Lincoln WaveDesigner

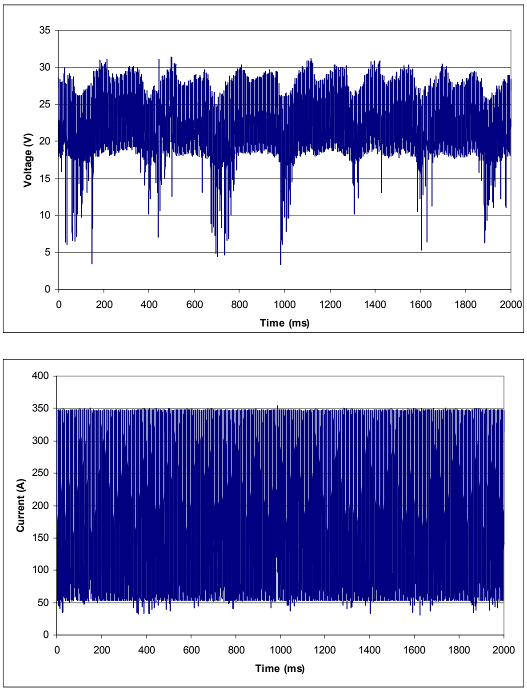
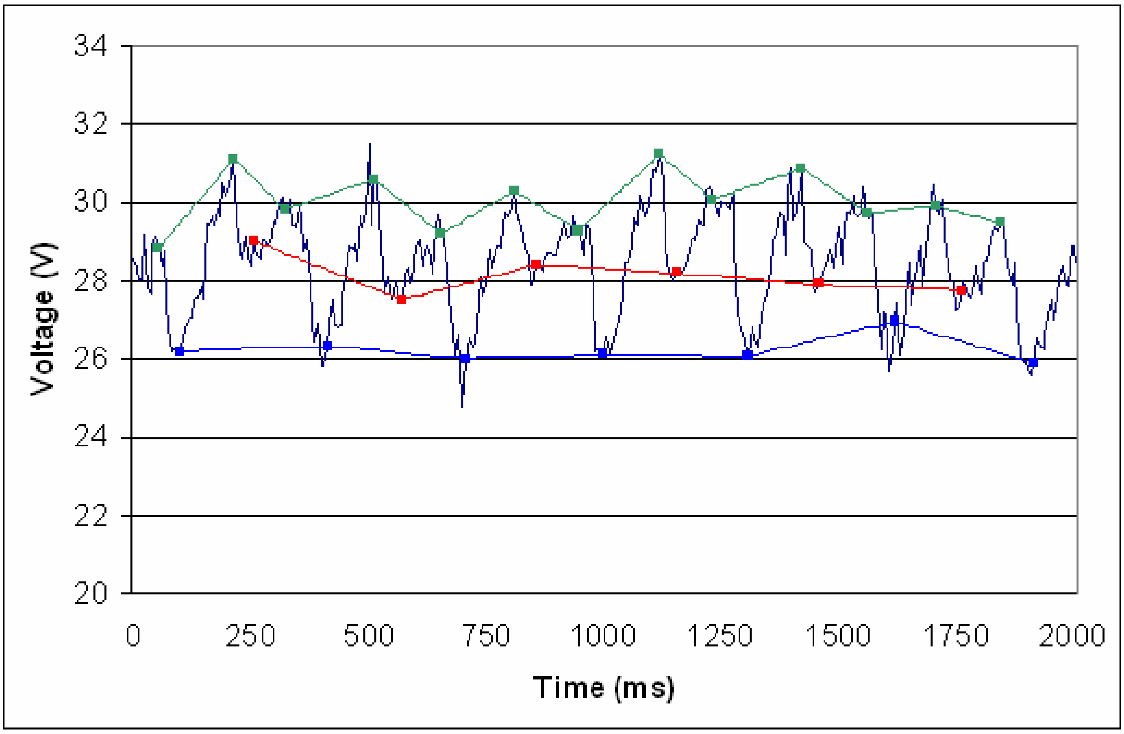


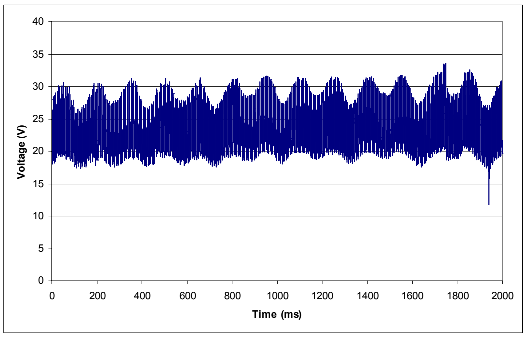
Figure B.3 - Voltage and Current waveforms from trial A1



Red and Blue lines are left and right side of contact tip oscillation (lower voltages)

Green is the contact tip oscillation when at centre (higher voltages)

Figure B.4 - Voltage result after applying the algorithms on trial A1



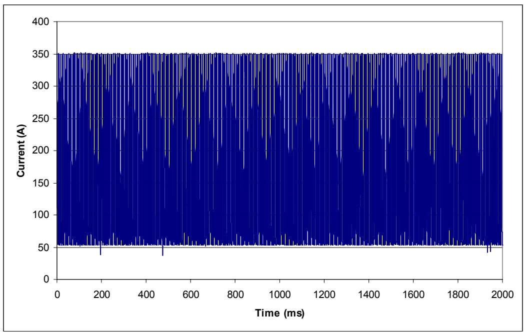
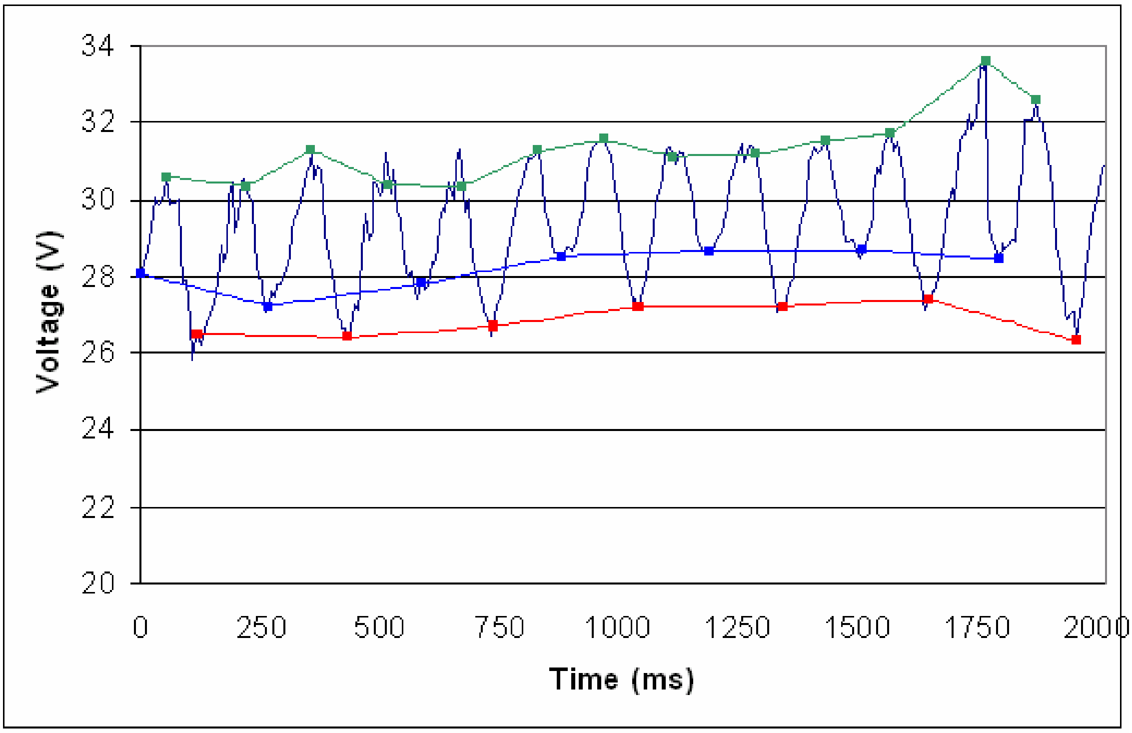


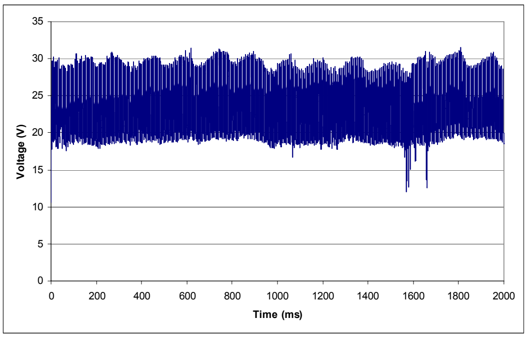
Figure B.5 - Voltage and Current waveforms from trial A2



Red and Blue lines are left and right side of contact tip oscillation (lower voltages)

Green is the contact tip oscillation when at centre (higher voltages)

Figure B.6 - Voltage result after applying the algorithms on trial A2



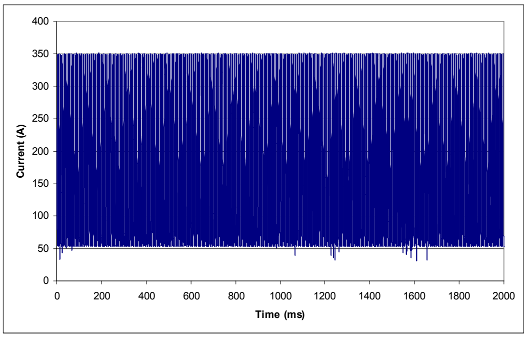
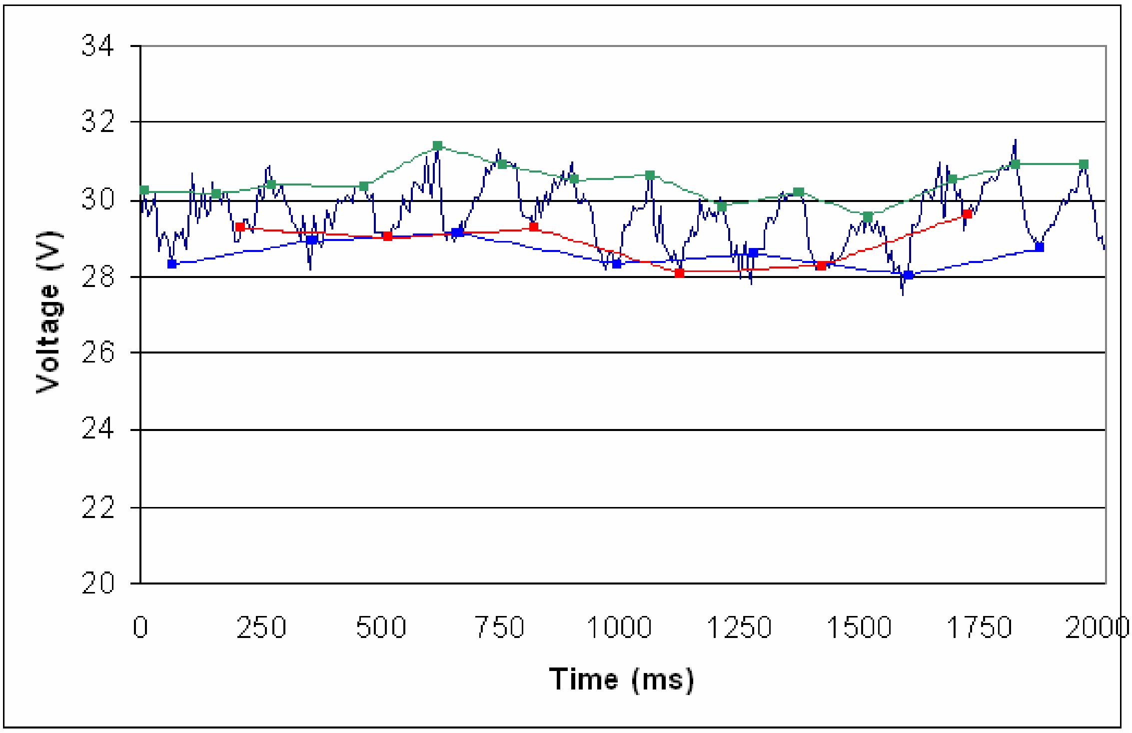


Figure B.7 - Voltage and Current waveforms from trial A3



Red and Blue lines are left and right side of contact tip oscillation (lower voltages)

Green is the contact tip oscillation when at centre (higher voltages)

Figure B.8 - Voltage result after applying the algorithms on trial A3

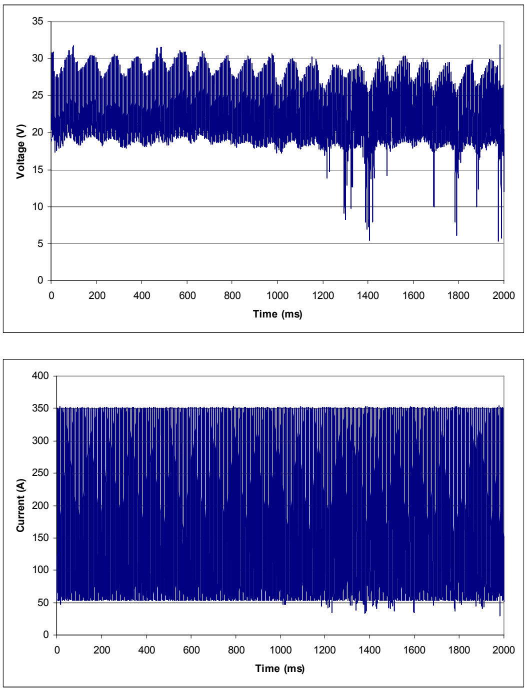
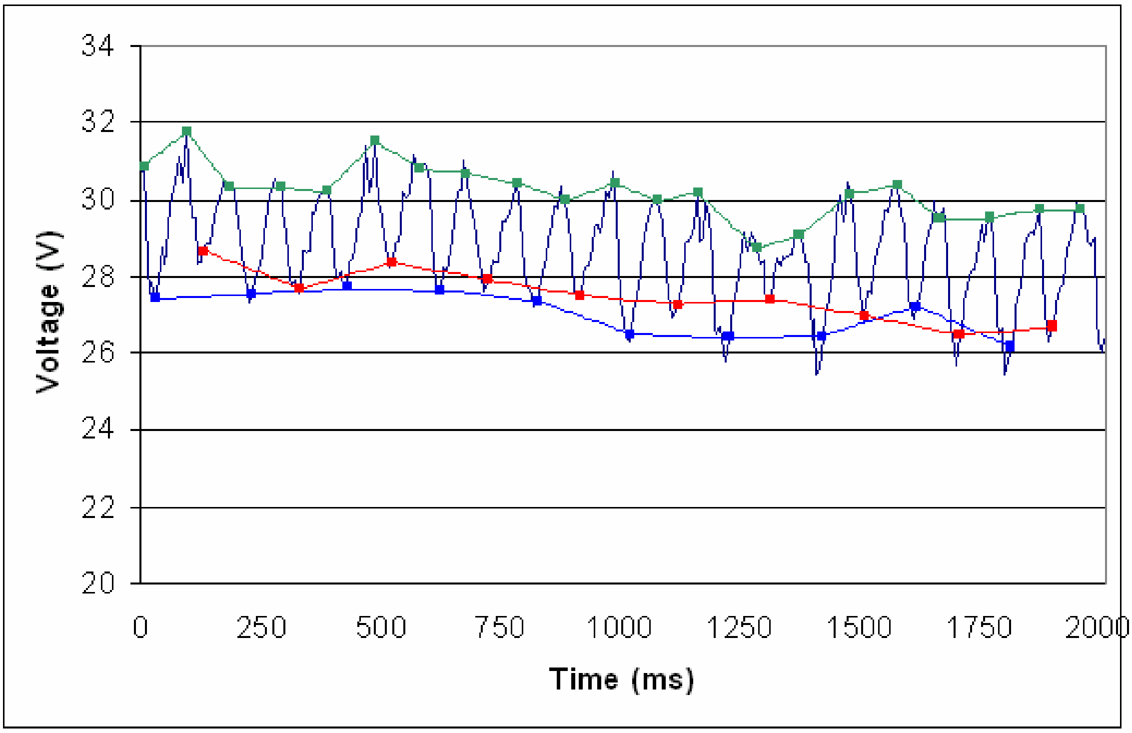


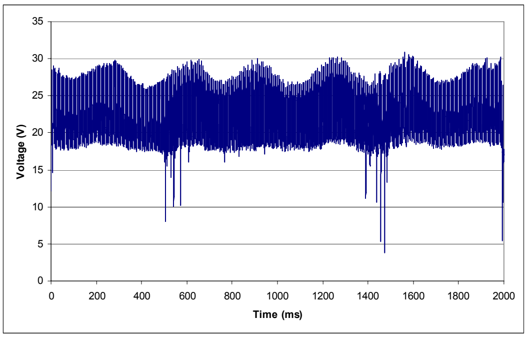
Figure B.9 - Voltage and Current waveforms from trial A4



Red and Blue lines are left and right side of contact tip oscillation (lower voltages)

Green is the contact tip oscillation when at centre (higher voltages)

Figure B.10 - Voltage result after applying the algorithms on trial A4



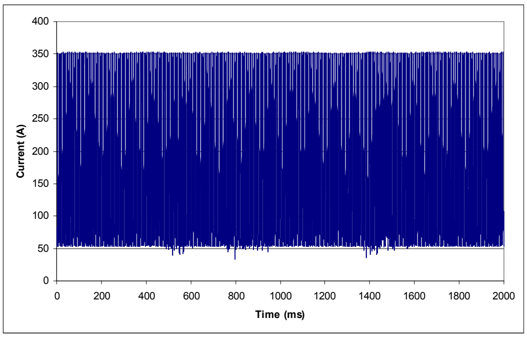
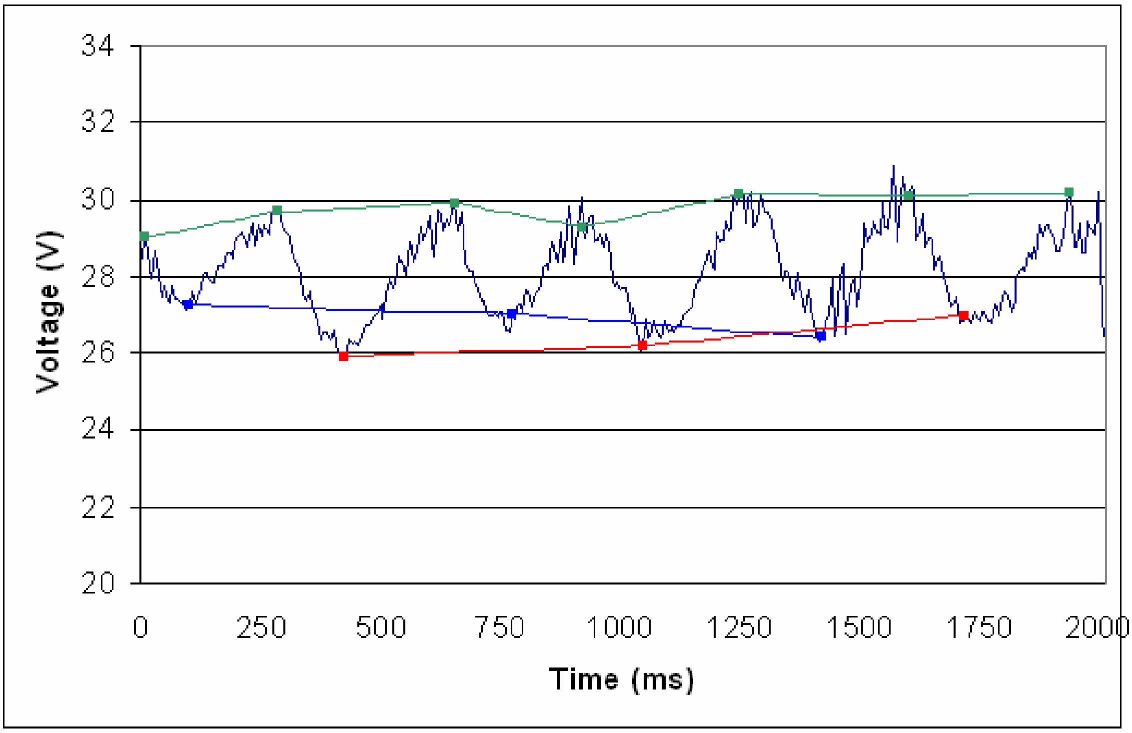


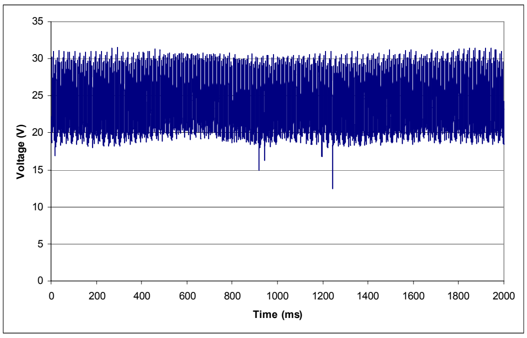
Figure B.11 - Voltage and Current waveforms from trial A5



Red and Blue lines are left and right side of contact tip oscillation (lower voltages)

Green is the contact tip oscillation when at centre (higher voltages)

Figure B.12 - Voltage result after applying the algorithms on trial A5



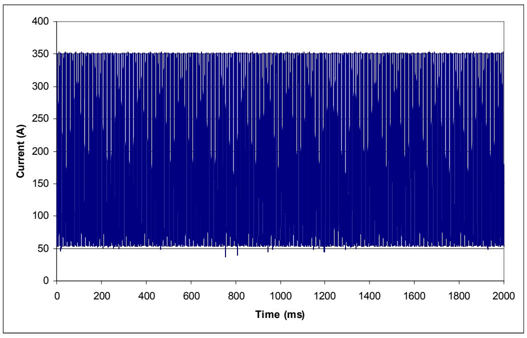
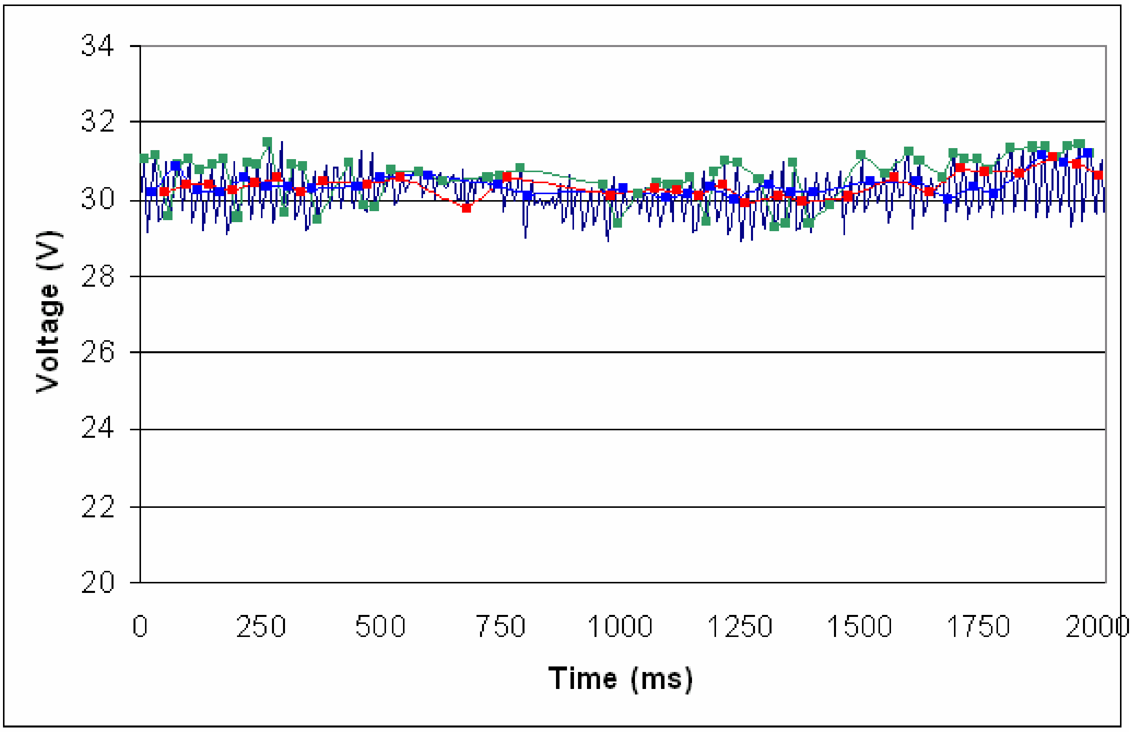


Figure B.13 - Voltage and Current waveforms from trial A6



Red and Blue lines are left and right side of contact tip oscillation (lower voltages)

Green is the contact tip oscillation when at centre (higher voltages)

Figure B.14 - Voltage result after applying the algorithms on trial A6

Appendix C. Digital Signal Processor developed control source code

```
/************************************************************************
/
/ Company: Cranfield University.
/
/ Copyright 2003 by Cranfield University.
/                   All rights reserved.
/
/ Author: Gil Lopes
/
/ Description:      This program uses the RMS Welding Systems VISENSE box for
/ sensing voltages and currents and therefore to send CAN messages through the CAN
/ network to correct torch parameters      on-line. Parts of the code are an adaption
/ from RMS code.
/
/ Version: 1.00
/
/ Released: Oct 2, 2003
/
/ File: vimain.c  (this is the main() file)
/
/************************************************************************/

// Global Code Definitions

// Include Files
#include "vimain.h"

// Version Number
#define MAJOR_REV 1x00
#define MINOR_REV 0x00


/************************************************************************/
//
// Function: Timer 0 Interrupt service routine
//
// Arguments:none
//
// Return Value: none
//
// Description: Handles LED events
//          quarter of second interrupt period that makes the LED
//          flashing accordingly to the bDevNum (Device Number)
//
/************************************************************************/
```

```
#pragma interrupt saveall

void Timer0 (void)
{
      asm (bfclr #SCR_TCF,X:TMRA0_SCR);             // Clears Time Compare Flag for the
                                                    // counter to start from the
beginning

      if (S_Process == S_Arc)
      {
                  ToggleLED;
      }
      else
      {
              if (scCounter0 == (bDevNum +1) * 2)   // compares the counter with the
                                                    // double of the device number plus
                                                    // 1 (to avoid the zero Dev Number =
                                                    // no LED blinking)
              {                                     // because each interrupt only
                                                    // toggles the LED
                      scCounter0 = -4;              // initializes the counter to a
                                                    // negative value for the pause time
              }                                     // more negative the value, bigger
                                                    // the pause (LED off)
              else
              {
                      if (scCounter0>=0) ToggleLED; // when the counter arrives
                                                    // to zero the LED starts
                                                    // blinking
                      scCounter0++;                 // increments the counter
              }
      }

#ifdef DEBUG_AVERAGE_SEND
      ADCAverageSend();
#endif

#ifdef       DEBUG_VOLTAGE_SEND
      ADCVoltageSend();
#endif

      return;
}
/*************************************************************************/
//
// Function: Timer 1 Interrupt service routine
//
// Arguments:none
```

```
//
// Return Value: none
//
// Description:  Processes CAN messages for vertical tracking when
//                            working whithout torch oscillation
//
/***********************************************************************/

#pragma interrupt saveall

void Timer1 (void)
{
#ifdef DEBUG_PROFILE
      SetGPIO(A, 6);                       // profiling pin
#endif
      asm (bfclr #SCR_TCF,X:TMRA1_SCR);    // Clears Time Compare Flag for the counter
                                           // to start from the beginning

      if (S_Process == S_Arc)              // If Process Status is in Arc state the
                                           // algorithm is processed
      {
                                           // Scale ADC Value to a Voltage reading

             ADC_V_Average = (word) ((((dword) ADC_V_Average_RAW) * ((dword)
ADC_Mult)) >> LgADC_Div);
                                           // Scale Ave Voltage Read from ADC value to
                                           // Voltage*100
                                           // (accuracy to .01V) -- casts to make sure
                                           // 32bit math is used
             ADC_V_Average += ADC_Offset;  // Straight linear scalling y = mx + b
                                           // Calculate Vertical Adjustment

             if ((ADC_V_Average > CAN_V_Average - Abnormal_Tol) && (ADC_V_Average <
CAN_V_Average + Abnormal_Tol))
                                           // defining a possible average window of
                                           // +/- Abnormal_Tol
             {
                    if (ADC_V_Average > CAN_V_Average)
                                           // if average calculated is
                                           // bigger than the average
                                           // received from the CAN
                    {
                           CAN_Vertical = (int) ((ADC_V_Average - CAN_V_Average) >>
LgPID_Divisor) - 1;                        // calculate the new value to lower the
                                           // torch
                    asm {
                           move CAN_Vertical,Y0
                                           // after the calculation, the result
                                           // is the amount of movement on the
```

```
                        // torch but
                not Y0          // it needs to be negated before
                        // sending to the bug to apply the
                        // proper movement DOWN
                move Y0,CAN_Vertical    // put the new value again into the
                                // variable
                }
        }

        if (ADC_V_Average < CAN_V_Average)
                                // if instead the average
                                // calculated is lower
        {
                CAN_Vertical = (int) ((CAN_V_Average - ADC_V_Average) >>
LgPID_Divisor);                         // calculate the new amount to be
                                // send to the bug, but since is to
                                // put the torch UP, doesn't meed to
                                // negate
        }
}

                                // Send Vertical Adjust CAN messages
if (CAN_Vertical != 0)                  // if there is a value to be sent
                                // different from zero, lets send it
{
        asm {
                move CAN_Vertical,Y0            // move the amount to the Y0
                                        // register
                move #$4c,X:CAN_TB0_IDR0        // Buffer 0 / Torch up/down
                move bCAN_id1,X0
                move X0,X:CAN_TB0_IDR1          // ID 0x260 + bDevNum (0-3)
                move #$02,X:CAN_TB0_DLR         // length 2
                move #$00,X:CAN_TB0_TBPR        // maximum priority
                move Y0,X:CAN_TB0_DSR1          // LSB loaded to its
                                        // position
                move #$0008,X0          // define X0 with the value for
                                // right shifting (8 bits) (2nd
                                // shift to send 2nd byte)
                lsrr Y0,X0,Y0           // shifts Y0 8 bits right
                move Y0,X:CAN_TB0_DSR0          // MSB and Sign loaded to
                                        // its position

                move #CANTFLG_TXE0,X:CANTFLG    //release message
                }

                CAN_Vertical = 0;       // reset the amount message again to
                                // be prepared for the next read and
                                // calculation
        }
```

```
      }

#ifdef DEBUG_PROFILE
      ClrGPIO(A, 6);                              // profiling pin
#endif
      return;
}

/************************************************************************/
//
// Function: Timer 2 Interrupt service routine
//
// Arguments:none
//
// Return Value: none
//
// Description:  Handles ADC events. Defines sampling pace
//
/************************************************************************/

#pragma interrupt saveall

void Timer2 (void)
{
      asm (bfclr #SCR_TCF,X:TMRA2_SCR);     // Clears Time Compare Flag for the
                                            // counter to start from the beginning

      asm (move #(ADCR1_STARTADC | ADCR1_EOSIE),X:ADCA_ADCR1);
// start ADC with once sequence, single ended and interrupt enabled
// (User's Manual -  p269)
      return;
}

#pragma mark ----------------


/************************************************************************/
//
// Function: CAN_Receive Interrupt service routine
//
// Arguments:none
//
// Return Value: none
//
// Description:  Handles CAN received messages
//
/************************************************************************/

#pragma interrupt saveall
```

```
void CAN_Receive (void)
{
      static int ConversionCoeffsReceived = 0;
      static unsigned short coeff[CO_EFF_SIZE];

asm {

      move #$ffff,X:CANRFLG          // acknowledge interrupt by reseting receive flag
                                     // putting the proper receiving bit on

      move X:CANIDAC,X0                      // move the hit value to X0 register
      bfclr #$fff0,X0                        // clear unimportant bits

      //*************** First FILTER HIT comparator *******************
      cmp #$0000,X0                          // compare if it's hit 0 (filter 0)
                                             // (SetVoltage Filter)
      jne canr_filter1                       // jump if not to filter 1

      move X:CAN_RB_DSR0,Y0          // move first byte (DSR0) of message to Y0
      move #$08,X0                   // define value 8 for shifting
      lsll Y0,X0,Y0                  // shift Y0 to left 8 times
      add X:CAN_RB_DSR1,Y0           // add the second byte (DSR1) of message to Y0
      move Y0,CAN_V_Average          // the CAN_V_Average now has the value from the
                                     // pendant which is the set voltage*100
      jmp canr_label_end             // finish the routine

canr_filter1:
// *************** Next FILTER HIT comparator *******************
      cmp #$0001,X0                  // compare if it's hit 1 (filter 1) (OS_P_STATUS
                                     // Filter) (Arc Start/Stop)
      jne canr_filter2               // jump if not to filter 2
      move X:CAN_RB_DSR0,X0          // move the first and only byte from the message to
                                     // X0 register
      cmp #$0002,X0                  // compare if is 0x02 (ARC START)
      jne canr_label2                // if not jump to next check
      move CAN_V_Average,Y0          // if is arc ON check if CAN_V_Average is still
                                     // Zero or if it has received already a value
      cmp #Zero,Y0                   // compare it is Zero
      jeq canr_label_end             // if it is, exit routine
      move #SCR_TCFIE,X:TMRA2_SCR    // if not, then it means that Arc is ON and it's
                                     // time to enable timer2 compare interrupt for
                                     // starting the ADC
      move #SCR_TCFIE,X:TMRA1_SCR    // if not, then it means that Arc is ON and it's
                                     // time to enable timer1 compare interrupt for
                                     // starting the ADC
      move #Zero, ADC_Timer          // Reset ADC Timer
      move #Zero, ADC_V_In           // Reset Voltage In
      move #Zero, ADC_V_Out          // Reset Voltage Out
```

```
}

      ADC_V_Sum = (word) (((dword) CAN_V_Average << LgADC_Div) / (word) ADC_Mult);
                                // Convert set voltage from CAN
      ADC_V_Sum -= ADC_Offset;      // to voltage from ADC
      ADC_V_Average_RAW = ADC_V_Sum;// and set initial RAW average value
      ADC_V_Sum <<= LgBufferSize;   // and respective SUM variable

asm {
      jmp canr_label_end            // jump to the end
canr_label2:
      cmp #$0003,X0                 // check if value is 3 (ARC STOP)
      jne canr_label_end            // if not, jump to the end of the routine

      move #Zero,X:TMRA2_SCR        // if it is, disable timer2 compare interrupt to
                                    // stop everything
      move #Zero,X:TMRA1_SCR        // if it is, disable timer1 compare interrupt to
                                    // stop everything
      move #$0002,bTPos             // Defaults torch positioning to 2 (N/A)
      move #S_Stop, S_Process       // Sets Process to its stop
      move #Zero,ADC_Last_Avg       // clear Last Average for the next run
      bfclr #LED,X:GPIO_E_DR        // Turn OFF LED
      jmp canr_label_end            // jump to the end

canr_filter2:
// *************** Next FILTER HIT comparator *******************
      cmp #$0002,X0                 // compare if it's hit 2 (filter 2) (CF_ThruArcTrac
                                    // Filter) (Configure calibration values)
      jne canr_filter3              // jump if not to filter 3

      move X:CAN_RB_DLR,X0          // compare the length of the message
      cmp #$0003,X0                 // if its three, continue
      jne canr_label3               // otherwise only set the flag so that the current
                                    // co-effs are returned
      move X:CAN_RB_DSR0,X0         // move the first byte from the message to X0
                                    // register
      move X0, ADC_Mult             // move X0 ADC_Mult
      move X:CAN_RB_DSR1,X0         // move the second byte from the message to X0
                                    // register
      move X0, ADC_Div              // move X0 ADC_Div
      move X:CAN_RB_DSR2,X0         // move the third byte from the message to X0
                                    // register
      move X0, ADC_Offset           // move X0 ADC_Offset
      jmp canr_label_end            // jump to the end
canr_label3:
      move #$0001,X0                // set flag to say that the co-effs have been
                                    // received
      move X0, ConversionCoeffsReceived // move X0 ADC_Mult
      jmp canr_label_end
```

```
canr_filter3:
// *************** Next FILTER HIT comparator ******************
      cmp #$0003,X0                 // compare if it's hit 3 (filter 3)
                                    // (ID0_CAN_Tch_Pos Filter) (torch position)
      jne canr_label_end            // jump if not to end of routine (no more filters)
      move X:CAN_RB_DSR1,X0         // move second byte from message to X0 register
      move X0,bTPos                 // define torch position
      move #Zero,X:TMRA1_SCR        // Disable Timer1 - H/V tracking will be done by
                                    // TorchPos() routine
      move S_Process, Y0            // Check if Process is in Arc state
      cmp #S_Arc, Y0                // compare if it's Arc state
      jne canr_label_end            // If not jump end
}
      TorchPos();                   // else call routine
canr_label_end:
// *************** Finish Filter Hits comparators ******************

      // If new Conversion Coeffs have been received, save them to flash.
      if (ConversionCoeffsReceived != 0)
      {
             coeff[0] = ADC_Mult;
             coeff[1] = ADC_Div;
             coeff[2] = ADC_Offset;
             coeff[3] = STORAGE_FLAG;
             EraseFlash(coeff_add, CO_EFF_SIZE);
             WriteFlash(coeff, coeff_add, CO_EFF_SIZE);
             LgADC_Div = log2(ADC_Div);
             ConversionCoeffsReceived = 0;
             ADCCoeffSend();
      }

      return;
}

/*************************************************************************/
//
// Function: CAN_Transmit Interrupt service routine
//
// Arguments:none
//
// Return Value: none
//
// Description: Handles CAN transmited messages
//
/*************************************************************************/

#pragma interrupt saveall
```

```
void CAN_Transmit (void)
{
      asm (move #Zero,X:CANTCR);             // reset interrupt state
      return;
}


#pragma mark ----------------

/************************************************************************/
//
// Function: ADC_Complete Interrupt service routine
//
// Arguments:none
//
// Return Value: none
//
// Description:  Handles all the calculations needed after sampling is completed
//
/************************************************************************/

#pragma interrupt saveall

void ADC_Complete (void)
{

#ifdef DEBUG_PROFILE
      SetGPIO(A, 7);                         // profiling pin
#endif

      asm (move #$0800,X:ADCA_ADSTAT);      // reset interrupt state
                                             // (User's Manual - p277)

asm {
      move X:ADCA_ADRSLT2,Y0                 // load Y0 register with value from AN1
      bfclr #$8000,Y0                        // clear sign bit - assuming always
                                             // positive voltages
      move #$0003,X0                         // define X0 with the value for right
                                             // shifting (3 bits) (1st shift to put
                                             // voltage in place)
      lsrr Y0,X0,Y0                          // shifts Y0 3 bits right
      move Y0,ADC_V_Read                     // move result to ADC_V_Read
                                             // (ADC voltage read)
      move X:ADCA_ADRSLT5,Y0                 // load Y0 register with value from AN5
      bfclr #$8000,Y0                        // clear sign bit - assuming always
                                             // positive voltages
      move #$0003,X0                         // define X0 with the value for right
                                             // shifting (3 bits) (1st shift to put
                                             // current in place)
```

```
    lsrr Y0,X0,Y0                         // shifts Y0 3 bits right
    move Y0,ADC_I_Read                    // move result to ADC_I_Read
                                          // (ADC current read)
    }


    if ((ADC_I_Read > Initial_Trigger) && (S_Process != S_Arc))
    {
            S_Process = S_Arc;
            ADC_I_Trigger = Initial_Trigger;
            ADC_V_Peak = 0;
    }

    if (S_Process == S_Arc)
    {
            if (ADC_Timer > Timer_Trigger)
            {
                    if (ADC_I_Read > ADC_I_Trigger)
                    {
                            if (ADC_I_Read > ADC_I_Max)
                            {
                                    ADC_I_Max = ADC_I_Read;
                                    ADC_V_Peak = ADC_V_Read;
                                    ADC_I_Trigger = (ADC_I_Max >> 1) + (ADC_I_Max >>
2);
                            }
                    }
                    else
                    {
                            ADC_I_Max = ADC_I_Trigger;
                    }
                    ADC_V_Sum += (dword) ADC_V_Read;
                    // To the sum variable the new read value is added
                    ADC_V_Sum -= (dword) ADC_V_Average_RAW;
                    // and subtracted by the old raw average
                    ADC_V_Average_RAW = (word) ((dword) ADC_V_Sum >> LgBufferSize);
                    // then shifted to obtain the new raw average value
            }
            else
            {
                    ADC_Timer++;
            }
    }



#ifdef DEBUG_PROFILE
    ClrGPIO(A, 7);                                // profiling pin
#endif
```

```
    return;
}

/***************************************************************************/
//
// Function: ADCAverageSend function
//
// Arguments: none
//
// Return Value: none
//
// Description: Sends average voltage calculated through CAN
//
/***************************************************************************/

#pragma interrupt called

void ADCAverageSend (void)
{
asm {
    move #$14,X:CAN_TB0_IDR0            // Buffer 0 / Average Voltage
    move #$80,X:CAN_TB0_IDR1            // ID 0xa4
    move #$03,X:CAN_TB0_DLR             // length 3
    move #$00,X:CAN_TB0_TBPR            // maximum priority
    move ADC_V_Average,Y0               // load Y0 register with value from AN1
    move Y0,X:CAN_TB0_DSR2              // LSB loaded to its position
    move #$0008,X0                      // define X0 with the value for right
                                        // shifting (8 bits) (2nd shift to send 2nd
                                        // byte)
    lsrr Y0,X0,Y0                       // shifts Y0 8 bits right
    move Y0,X:CAN_TB0_DSR1              // MSB and Sign loaded to its position
    move #$a0,X:CAN_TB0_DSR0            // load the channel reference AV1 = 0xa1
    move #CANTFLG_TXE0,X:CANTFLG        // release message
    }

    return;
}


/***************************************************************************/
//
// Function: ADCVoltageSend function
//
// Arguments: none
//
// Return Value: none
//
// Description: Sends values from AN1, AN2 and AN3 through CAN
//              buffers 0, 1 and 2 respectively. 1st byte on message
```

```
//            identifies channel name (a1, a2 and a3).
//
/*****************************************************************************/

#pragma interrupt called

void ADCVoltageSend (void)
{

asm {
      move #$14,X:CAN_TB0_IDR0            // Buffer 0 / Value from AN1
      move #$20,X:CAN_TB0_IDR1            // ID 0xa1
      move #$03,X:CAN_TB0_DLR             // length 3
      move #$00,X:CAN_TB0_TBPR            // maximum priority
      move X:ADCA_ADRSLT0,Y0              // load Y0 register with value from AN0 for
                                          // current
      move #$0003,X0                      // define X0 with the value for right
shifting
                                          // (3 bits) (1st shift to put voltage in
place)
      lsrr Y0,X0,Y0                       // shifts Y0 3 bits right
      move Y0,X:CAN_TB0_DSR2              // LSB loaded to its position
      move #$0008,X0                      // define X0 with the value for right
shifting
                                          // (8 bits) (2nd shift to send 2nd byte)
      lsrr Y0,X0,Y0                       // shifts Y0 8 bits right
      move Y0,X:CAN_TB0_DSR1              // MSB and Sign loaded to its position
      move #$a1,X:CAN_TB0_DSR0            // load the channel reference AN1 = 0xa1
      move #$14,X:CAN_TB1_IDR0            // Buffer 1 / Value from AN2
      move #$40,X:CAN_TB1_IDR1            // ID 0xa2
      move #$03,X:CAN_TB1_DLR             // length 3
      move #$00,X:CAN_TB1_TBPR            // maximum priority
      move X:ADCA_ADRSLT4,Y0              // load Y0 register with value from AN4 for
                                          // current
      move #$0003,X0                      // define X0 with the value for right
shifting
                                          // (3 bits) (1st shift to put voltage in
place)
      lsrr Y0,X0,Y0                       // shifts Y0 3 bits right
      move Y0,X:CAN_TB1_DSR2              // LSB loaded to its position
      move #$0008,X0                      // define X0 with the value for right
shifting
                                          // (8 bits) (2nd shift to send 2nd byte)
      lsrr Y0,X0,Y0                       // shifts Y0 8 bits right
      move Y0,X:CAN_TB1_DSR1              // MSB and Sign loaded to its position
      move #$a2,X:CAN_TB1_DSR0            // load the channel reference AN1 = 0xa2
      move #$14,X:CAN_TB2_IDR0            // Buffer 2 / Value from AN3
      move #$60,X:CAN_TB2_IDR1            // ID 0xa3
      move #$03,X:CAN_TB2_DLR             // length 3
```

```
      move #$00,X:CAN_TB2_TBPR             // maximum priority
      move X:ADCA_ADRSLT5,Y0               // load Y0 register with value from AN5 for
                                           // current
      move #$0003,X0                       // define X0 with the value for right
shifting
                                           // (3 bits) (1st shift to put voltage in
place)
      lsrr Y0,X0,Y0                        // shifts Y0 3 bits right
      move Y0,X:CAN_TB2_DSR2               // LSB loaded to its position
      move #$0008,X0                       // define X0 with the value for right
shifting
                                           // (8 bits) (2nd shift to send 2nd byte)
      lsrr Y0,X0,Y0                        // shifts Y0 8 bits right
      move Y0,X:CAN_TB2_DSR1               // MSB and Sign loaded to its position
      move #$a3,X:CAN_TB2_DSR0             // load the channel reference AN1 = 0xa3

      move #(CANTFLG_TXE0 | CANTFLG_TXE1 | CANTFLG_TXE2), X:CANTFLG
                                           //release all three messages
      }

      return;
}


//****************************************************************************/
//
// Function: ADCCoeffSend function
//
// Arguments: none
//
// Return Value: none
//
// Description: Sends values of the ADC conversion coefficients out through CAN
//
/****************************************************************************/

#pragma interrupt called

void ADCCoeffSend (void)
{
asm {
      move #$D0,X:CAN_TB0_IDR0             // Use buffer 0
      move bCAN_id1,X0                     // Device Number, ID1 = id based on devnum
      move X0,X:CAN_TB0_IDR1               // Send on ID 0x680 + device ID
      move #$03,X:CAN_TB0_DLR              // length 3
      move #$00,X:CAN_TB0_TBPR             // maximum priority
      move ADC_Mult,Y0                     // load Y0 register with value from
ADC_Multi
      move Y0,X:CAN_TB0_DSR0               // load into byte 0
```

```
        move ADC_Div,Y0                   // load Y0 register with value from ADC_Div
        move Y0,X:CAN_TB0_DSR1            // load into byte 1
        move ADC_Offset,Y0                // load Y0 register with value from
ADC_Offset
        move Y0,X:CAN_TB0_DSR2            // load into byte 2
        move #CANTFLG_TXE0,X:CANTFLG      // release message
        }
        return;
}

//****************************************************************************/
//
// Function: TorchPos function
//
// Arguments: none
//
// Return Value: none
//
// Description: Calculates and sends CAN messages to correct vertical
//              and horizontal tracking when torch oscillation is present
//
/****************************************************************************/

#pragma interrupt called

void TorchPos (void)
{
#ifdef DEBUG_PROFILE
        SetGPIO(A, 6);                    // profiling pin
#endif


        if (ADC_V_Peak != 0)
        {
                                          // Scale ADC Value to a Voltage reading

                ADC_V_Average = (word) ((((dword) ADC_V_Average_RAW) * ((dword)
ADC_Mult)) >> LgADC_Div);                 // Scale Ave Voltage Read from ADC value to
                                          // Voltage*100
                                          // (accuracy to .01V) -- casts to make sure
32bit
                                          // math is used
                ADC_V_Average += ADC_Offset;
                                          // Straight linear scalling y = mx + b

                if (ADC_Last_Avg == 0) ADC_Last_Avg = ADC_V_Average;
                                          // Sets for the first time the previous
                                          // average variable
                if (bTPos == T_Mid)       // If the sent position is middle then the
```

```
                                        // vertical tracking is done
            {
                                        // Calculate Vertical Adjustment
                    if ((ADC_V_Average > (CAN_V_Average - Abnormal_Tol)) &&
(ADC_V_Average < (CAN_V_Average + Abnormal_Tol)))
                                        // defining a possible average window of
+/-
                                        // Abnormal_Tol
                    {
                            if (ADC_V_Average > CAN_V_Average)
                                        // if average calculated is bigger than the
                                        // average received from the CAN
                            {
                                    CAN_Vertical = (int) ((ADC_V_Average -
CAN_V_Average) >> LgPID_Divisor) - 1; // calculate the new value to lower the torch
                                    asm {
                                            move CAN_Vertical,Y0
                                    // after the calculation, the result is the
                                    // amount of movement on the torch but
                                            not Y0
                                    // it needs to be negated before sending to
                                    // the bug to apply the proper movement DOWN
                                            move Y0,CAN_Vertical
                                    // put the new value again into the variable
                                    }
                            }

                            if (ADC_V_Average < CAN_V_Average)
                                    // if instead the average calculated is lower
                            {
                                    CAN_Vertical = (int) ((CAN_V_Average -
ADC_V_Average) >> LgPID_Divisor);   // calculate the new amount to be send to the
                                    // bug,but since is to put the torch UP,
                                    // doesn't meed to negate
                            }
                    }

                                    // Send Vertical Adjust CAN messages
                    if (CAN_Vertical != 0)
                                    // if there is a value to be sent different
                                    // from zero, lets send it
                    {
                            asm {
                                    move CAN_Vertical,Y0
                                    // move the amount to the Y0
                                    // register
                                    move #$4c,X:CAN_TB0_IDR0
                                    // Buffer 0 / Torch up/down
```

```
                move bCAN_id1,X0                // move address
                move X0,X:CAN_TB0_IDR1 // ID 0x260 + bDevNum (0-3)
                move #$02,X:CAN_TB0_DLR         // length 2
                move #$00,X:CAN_TB0_TBPR        // maximum priority
                move Y0,X:CAN_TB0_DSR1          // LSB loaded to its position
                move #$0008,X0                  // define X0 with the value for
                                                // right shifting (8 bits)
                                                // (2nd shift to send 2nd byte)
                lsrr Y0,X0,Y0                   // shifts Y0 8 bits right
                move Y0,X:CAN_TB0_DSR0          // MSB and Sign loaded to its
                                                // position
        #ifdef DEBUG_OSCAVG_SEND
                move #$14,X:CAN_TB1_IDR0        // Buffer 1 / Average Voltage
                move #$80,X:CAN_TB1_IDR1        // ID 0xa4
                move #$03,X:CAN_TB1_DLR         // length 3
                move #$00,X:CAN_TB1_TBPR        // maximum priority
                move ADC_V_Average,Y0           // load Y0 register with value from
                                                // Avg
                move Y0,X:CAN_TB1_DSR2          // LSB loaded to its position
                move #$0008,X0                  // define X0 with the value for
                                                // right shifting (8 bits)
                                                // (2nd shift to send 2nd byte)
                lsrr Y0,X0,Y0                   // shifts Y0 8 bits right
                move Y0,X:CAN_TB1_DSR1          // MSB and Sign loaded to its
                                                // position
                move #$a0,X:CAN_TB1_DSR0        // load the channel reference
                                                // AV1 = 0xa1

                move #(CANTFLG_TXE0 | CANTFLG_TXE1),X:CANTFLG
                                                //release message
        #else
                move #CANTFLG_TXE0,X:CANTFLG    //release message
        #endif

        }

            CAN_Vertical = 0;
// reset the amount message again to be prepared for the next read and calculation
        }
        else
        {
#ifdef DEBUG_OSCAVG_SEND
            asm{
                move #$14,X:CAN_TB0_IDR0        // Buffer 1 / Average Voltage
                move #$80,X:CAN_TB0_IDR1        // ID 0xa4
                move #$03,X:CAN_TB0_DLR         // length 3
                move #$00,X:CAN_TB0_TBPR        // maximum priority
                move ADC_V_Average,Y0           // load Y0 register with value
                                                // from Avg
```

```
                move Y0,X:CAN_TB0_DSR2        // LSB loaded to its position
                move #$0008,X0                // define X0 with the value for
                                              // right shifting (8 bits)
                                              // (2nd shift to send 2nd byte)
                lsrr Y0,X0,Y0                 // shifts Y0 8 bits right
                move Y0,X:CAN_TB0_DSR1        // MSB and Sign loaded to its
                                              // position
                move #$a0,X:CAN_TB0_DSR0      // load the channel reference
                                              // AV1 = 0xa1
                move #CANTFLG_TXE0,X:CANTFLG  //release message
        }
#endif

}


        }
        else
        {
                // Calculate Horizontal Adjustment
                if (bTPos == T_Out)
                {
                        ADC_V_Out = ADC_V_Peak;
                }
                else
                {
                        ADC_V_In = ADC_V_Peak;
                }

                if ((ADC_V_In != 0) && (ADC_V_Out != 0))
                {
                        if (ADC_V_Out < ADC_V_In)
                        {
                                CAN_Horizontal = CAN_H_Step_Out;
                        }
                        else
                        {
                                CAN_Horizontal = CAN_H_Step_In;
                        }

                }
                asm {
                        move CAN_Horizontal,Y0 // after the calculation, the
                                              // result is the amount of
                                              // movement on the torch but
                        not Y0                // it needs to be negated
                                              // before sending to the bug to
                                              // apply the proper movement
                                              // Out
                        move Y0,CAN_Horizontal // put the new value again into
```

```
                                              // the variable
                    move #ID0_SP_HorzAd,X:CAN_TB0_IDR0 // Buffer 0 / Torch In/Out
                     move bCAN_id1,X0             // load X0 with device ID
                    move X0,X:CAN_TB0_IDR1        // and put it in position
                    move #$02,X:CAN_TB0_DLR       // length 2
                    move #$00,X:CAN_TB0_TBPR      // maximum priority
                    move CAN_Horizontal,Y0        // load Y0 register with value
                                                  // from AN1
                    move Y0,X:CAN_TB0_DSR1        // LSB loaded to its position
                    move #$0008,X0                // define X0 with the value for
                                                  //right shifting (8 bits)
                                                  // (2nd shift to send 2nd byte)
                    lsrr Y0,X0,Y0                 // shifts Y0 8 bits right
                    move Y0,X:CAN_TB0_DSR0        // MSB and Sign loaded to its
                                                  // position
                    move #CANTFLG_TXE0,X:CANTFLG  //release message
                           }
             }
     }

#ifdef DEBUG_PROFILE
     ClrGPIO(A, 6);                               // profiling pin
#endif
     return;

}


#pragma mark ----------------

/*****************************************************************************/
//
// Function: Main function
//
// Arguments: none
//
// Return Value: none
//
// Description: Initialize hardware and executes main loop
//
/*****************************************************************************/
void main(void)
{
// Variables definition

// Variables Initialization
     scCounter0 = 0;
     bDevNum = 0;
     bCANSpeed = 0;
```

```
	CAN_V_Average = 0;
	bTPos = 2;					// default torch position (N/A)
	S_Process = S_Stop;

// Other Initialization
	init();

#ifdef DEBUG_PROFILE
// Initialize GPIO pins 6 and 7 for profiling
	SetupGPOutput('A', 6);		// portA 6 used for debug
	SetupGPOutput('A', 7);		// portA 7 used for debug
#endif

/// Main Loop
	while (1)
	{



	}						// Main Loop

}						// Main

#pragma mark ----------------





/*****************************************************************************/
//
// Function: Init function
//
// Arguments: none
//
// Return Value: none
//
// Description: Initialize whole hardware
//				[ Interrupts Setup	]
//				[ Timers Setup		]
//				[ GPIO Setup		]
//				[ CAN Setup			]
//				[ ADC Setup			]
//
/*****************************************************************************/
void init(void)
{
	static unsigned short coeff[CO_EFF_SIZE];

asm {
```

```
// Interrupts Setup ================================================================
      bfset #$0100,sr                      // prepare Status Register
      bfclr #$0200,sr                      // for enabling interrupts
      bfset #$fe12,X:IPR                   // enable both IRQ A & B, and all other
                                           // interrupts (User's Manual - p138)
      bfset #$0100,X:GPR10                 // Timer A Ch 0 interrupt priority - 1
                                           // (User's Manual - p144)
      bfset #$4000,X:GPR10                 // Timer A Ch 1 interrupt priority - 4
                                           // (User's Manual - p144)
      bfset #$0007,X:GPR11                 // Timer A Ch 2 interrupt priority - 7
                                           // (User's Manual - p144)
      bfset #$7700,X:GPR3                  // CAN Receive and Transmit interrupt
                                           // priorities - 7 (User's Manual - p144)
      bfset #$7000,X:GPR13                 // ADC Complete interrupt priority - 7
                                           // (User's Manual - p144)
// End of Interrupt Setup ==========================================================

// Timers Setup ====================================================================
// IPBus / 128 and CountDown=0x7a12 is equivalent to 1/10 of a second
// Timer A Ch 0 - Priority 1 (MIN) - responsible for the LED events
      move #Zero,X:TMRA0_CTRL              // stop timer 0 (User's Manual - p428)
      move #CountDown0,X:TMRA0_CMP1        // timer 0 CMP1 loaded with CountDown value
                                           // (User's Manual - p425)
      move #Zero,X:TMRA0_CMP2              // timer 0 CMP2 loaded with Zero (#$0000)
                                           // (User's Manual - p426)
      move #CountDown0,X:TMRA0_LOAD // timer 0 LOAD loaded with CountDown0 value
                                           // (User's Manual - p427)
      move #CountDown0,X:TMRA0_CNTR        // timer 0 CNTR loaded with CountDown0
value
                                           // (User's Manual - p428)
      move #$3e30,X:TMRA0_CTRL      // timer0 primary source on IPBus / 128, Len
                                           // 1,  Dir down (User's Manual - p419)
#ifdef TIMER0_ON
      move #SCR_TCFIE,X:TMRA0_SCR          // enable timer0 compare interrupt
                                           //(User's Manual - p422)
#endif

// Timer A Ch 1 - Priority 4  - responsible for CAN events
      move #Zero,X:TMRA1_CTRL              // stop timer 1 (User's Manual - p428)
      move #CountDown1,X:TMRA1_CMP1        // timer 1 CMP1 loaded with CountDown value
                                           // (User's Manual - p425)
      move #Zero,X:TMRA1_CMP2              // timer 1 CMP2 loaded with Zero (#$0000)
                                           // (User's Manual - p426)
      move #CountDown1,X:TMRA1_LOAD // timer 1 LOAD loaded with CountDown1 value
                                           // (User's Manual - p427)
      move #CountDown1,X:TMRA1_CNTR // timer 1 CNTR loaded with CountDown1 value
                                           // (User's Manual - p428)
      move #$3e30,X:TMRA1_CTRL      // timer1 primary source on IPBus / 128, Len
                                           // 1,  Dir down (User's Manual - p419)
```

```
#ifdef TIMER1_ON
      move #SCR_TCFIE,X:TMRA1_SCR           // enable timer1 compare interrupt
                                            // (User's Manual - p422)
#endif

// Timer A Ch 2 - Priority 7 (MAX) - responsible for the ADC events
      move #Zero,X:TMRA2_CTRL       // stop timer 0 (User's Manual - p428)
      move #CountDown2,X:TMRA2_CMP1 // timer 2 CMP1 loaded with CountDown value
                                    // (User's Manual - p425)
      move #Zero,X:TMRA2_CMP2       // timer 2 CMP2 loaded with Zero (#$0000)
                                    // (User's Manual - p426)
      move #CountDown2,X:TMRA2_LOAD // timer 2 LOAD loaded with CountDown2 value
                                    // (User's Manual - p427)
      move #CountDown2,X:TMRA2_CNTR // timer 2 CNTR loaded with CountDown2 value
                                    // (User's Manual - p428)
      move #$3e30,X:TMRA2_CTRL      // timer2 primary source on IPBus / 128, Len
                                    // 1,  Dir down (User's Manual - p419)
#ifdef TIMER2_ON
      move #SCR_TCFIE,X:TMRA2_SCR   // enable timer2 compare interrupt
                                    // (User's Manual - p422)
#endif

// End of Timers Setup ===============================================================

// GPIO Setup ==================================================================
// GPIO (LED) Port E ; Pin 6 (User's Manual - from p197)
      bfset #LED,X:GPIO_E_PUR       // ensure pull up is enabled
                                    // (User's Manual - p197)
      bfset #LED,X:GPIO_E_DDR       // set pin as output (User's Manual - p197)
      bfclr #LED,X:GPIO_E_PER       // clear peripheral enable so GPIO pin can be
                                    // used as IO (User's Manual - p198)
      bfclr #LED,X:GPIO_E_IENR      // disable interupts (User's Manual - p199)
      bfclr #LED,X:GPIO_E_DR        // turn off LED (User's Manual - p197)

// GPIO (Rotary Switch and Debug) Port A ; Pins 1,2,3,4 Inputs - Pins 7,8 Outputs
      bfset #GPIO_A_Pins_Mask,X:GPIO_A_PUR  // ensure pull ups are enabled
                                    //(User's Manual - p197)
      move  #GPIO_A_Pins,X:GPIO_A_DDR// define which are inputs and outputs
                                    //(User's Manual - p197)
      bfclr #GPIO_A_Pins_Mask,X:GPIO_A_PER  // clear peripheral enable so GPIO pins can
be
                                    // used as IOs (User's Manual - p198)
      bfclr #GPIO_A_Pins_Mask,X:GPIO_A_IENR // disable interupts (User's Manual - p199)
      move X:GPIO_A_DR,X0           // read the input value to the X0 reg
                                    // (User's Manual - p197)
      not X0                        // invert X0 bits because input is inverted
      move X0,Y0                    // copy X0 reg to Y0 reg
      bfclr #$fff5,Y0               // clear all bits in Y0 reg except the bits
                                    //that are in position (2nd and 4th)
```

```
        bftstl #$0004,X0             // test 3rd bit in X0 reg
        bcs label1                   // if it's zero jmp label1
        bfset #$0001,Y0              // if it's not zero, set the 1st bit in Y0
label1:
        bftstl #$0001,X0             // test 1st bit in X0 reg
        bcs label2                   // if it's zero jmp label2
        bfset #$0004,Y0              // if it's not zero set the same 3rd bit in Y0
label2:
        dec Y0                       // decrement Y0 by one to adapt the
                                     // values (0-3 instead 1-4)
        move Y0,X0                   // copy Y0 to X0 for using Y0 with the bDevNum
                                     // and X0 to bCANSpeed
        bfclr #$fffc,Y0              // clear all bits except 1st and 2nd (those
                                     // who defines the Device Number from 0-3)
        move Y0,bDevNum              // copy the resultant value to bDevNum
        bfclr #$fff3,X0              // clear now all bits except 3rd and 4th ones
                                     // (those who defines the CAN Speed)
        lsr X0                       // shifts the result to the right just for
                                     // having a CAN speed defined from 0-3
        lsr X0                       // the second needed shift
        move X0,bCANSpeed            // move the resultant value to the proper
                                     // variable (bCANSpeed)

// End of GPIO Setup ==========================================================
        }

        bCAN_id1 = bDevNum << 5;     // setup a byte two for the can id registers
                                     // in the DSP based on the devnum to make life
                                     // easier


        asm {
// CAN Setup  ==================================================================
// CAN Parameters:  jump width 2, 3 samples, time seg1 6 TQ, time segment2 12 TQ,
// prescaler 4,
        bfset #CANCTL0_SFTRES,X:CANCTL0      // set the soft reset mode bit
                                             // (User's Manual - p225)
        move #(CANCTL1_CANE | CANCTL1_CLKSRC),X:CANCTL1// enable CAN module and define
                                             // IPBus as source clock
                                             // (User's Manual - p227)
        move #$001f,Y0                       // prepare Y0 register to the BRP
                                             // calculation
        move bCANSpeed,X0                    // move to X0 the CAN Speed from the
                                             // rotary switch
        inc X0                               // increment X0 by 1 because Speed=3
                                             // is the lowest BRP
        bfclr #$000c,X0                      // incrementing X0; in case of speed
                                             // 3, bit 2 will have a 1 so its
                                             // cleared
```

```
      lsll Y0,X0,Y0                       // make a left shift to Y0, X0 times
                                          // to have the proper value in the
                                          // most significant nibble
      move #$0004,X0                      // now lets define new value for
                                          // right shifting
      lsrr Y0,X0,Y0                       // do the right shifting X0 times.
                                          // BRP is now defined in Y0
                                          // register.
      bfset #CANBTR0_SJW1,Y0              // its only need to add the SJW1
                                          // value; it means jump width 2 and
                                          // prescaler equals to Y0
      move Y0,X:CANBTR0                   // and move it to the proper CAN
                                          // register (User's Manual - p229)
      bfset #(CANBTR1_SAMP | CANBTR1_TSEG22 | CANBTR1_TSEG20 | CANBTR1_TSEG13 |
CANBTR1_TSEG12),X:CANBTR1
                                          // Sampling bit + bit 0 and 2 from
                                          // TSEG2 + bit 2 and 3 from TSEG1
                                          // (User's Manual - p230)

      // CAN filters

      move #CANIDAC_IDAM0,X:CANIDAC       // Four 16-bit Acceptance Filters
                                          // (User's Manual - p238)

      move #ID0_SetVoltageMsg,X:CANIDAR0  // SetVoltage Filter
      move #$00,X:CANIDMR0                // Message ID0 = ID0_SetVoltageMsg
      move bCAN_id1,X0                    // Torch Number ID1 = id based on
                                          // devnum
      move X0,X:CANIDAR1                  // Torch Number ID1 = id based on
                                          // devnum
      move #$07,X:CANIDMR1                // Torch Number ID1 = id based on
                                          // devnum
      move #ID0_OS_P_STATUS,X:CANIDAR2    // OS_PxStatus index Filter
      move #$00,X:CANIDMR2                // Message ID0 = ID0_OS_P_STATUS
      move bCAN_id1,X0                    // Torch Number ID1 = id based on
                                          // devnum
      move X0,X:CANIDAR3                  // Torch Number ID1 = id based on
                                          // devnum
      move #$f7,X:CANIDMR3                // Torch Number ID1 = id based on
                                          // devnum
      // filter for config
      move #ID0_CF_ThruArcTrac,X:CANIDAR4 // CF_ThruArcTrac index Filter
      move #$00,X:CANIDMR5                // Message ID0 = ID0_CF_ThruArcTrac
      move bCAN_id1,X0                    // Torch Number ID1 = id based on
                                          // devnum
      move X0,X:CANIDAR5                  // Torch Number ID1 = id based on
                                          // devnum
      move #$f7,X:CANIDMR5                // Torch Number ID1 = id based on
                                          // devnum
```

```
    // filter for torch position
    move #ID0_CAN_Tch_Pos,X:CANIDAR6       // 0x03Ax from the BUG indicating
                                           // toch posistion
    move #$00,X:CANIDMR6                   // Message ID0 = ID0_CAN_Tch_Pos
    move bCAN_id1,X0                       // Torch Number ID1 = id based on
                                           // devnum
    move X0,X:CANIDAR7                     // Torch Number ID1 = id based on
                                           // devnum
    move #$f7,X:CANIDMR7                   // Torch Number ID1 = id based on
                                           // devnum
    // End of CAN filters definition

    bfclr #(CANCTL0_SFTRES | CANCTL0_SLPRQ),X:CANCTL0    // clears the soft reset
mode
                                           // bit and sleep mode bit
                                           // (User's Manual - p225)
    move #$ffff,X:CANRFLG                  // clears CAN Receiver Flag
                                           // Register (User's Manual - p232)
    move #Zero,X:CANTFLG                   // clears CAN Transmitter Flag
                                           // Register (User's Manual - p236)
    bfset #CANRIER_RXFIE,X:CANRIER         // sets the Receiver Full Interrupt
                                           // Enable (User's Manual - p235)
    bfset #Zero,X:CANTCR                   // sets the Transmitter Empty
                                           // Interrupt Enable for buffer 0
                                           // (not used for now)
                                           // (User's Manual - p237)
// End of CAN Setup ==================================================================

// ADC (Analog-to-Digital Converter) Setup ========================================
    move #ADCR1_STOPADC,X:ADCA_ADCR1       // stop ADC to setup registers
                                           // (User's Manual - p269)
    move #$0009,X:ADCA_ADCR2               // clock divisor selection
                                           // (User's Manual - p273)
    move #Zero,X:ADCA_ADZCC                // disables zero crossing control
                                           // (User's Manual - p273)
    move #$3210,X:ADCA_ADLST1              // order in which channels are
                                           // sampled (User's Manual - p274)
    move #$0054,X:ADCA_ADLST2              // continuation of channel order
                                           // (User's Manual - p274)
    move #$00c0,X:ADCA_ADSDIS              // which channels are off (6th and
                                           // 7th) (User's Manual - p276)
    move #Zero,X:ADCA_ADCR1                // clear whole control 1 register,
                                           // inclusive the stop bit
    // Adjusting the Offset
        move #Zero,X:ADCA_ADOFS0
        move #Zero,X:ADCA_ADOFS1
        move #Zero,X:ADCA_ADOFS2
        move #Zero,X:ADCA_ADOFS3
        move #Zero,X:ADCA_ADOFS4
```

```
            move #Zero,X:ADCA_ADOFS5
            move #Zero,X:ADCA_ADOFS6
      // End of Offset adjustement

// uncomment next line for debugging purposes (ADC start)
//    asm (move #(ADCR1_STARTADC | ADCR1_EOSIE),X:ADCA_ADCR1);// start ADC with once
sequence, single ended and interrupt enabled (User's Manual - p269)

// End of ADC Setup ===============================================================
      }

      // Set ADC Conversion co-efficients from flash
      ReadFlash(coeff, coeff_add, CO_EFF_SIZE);
      if (coeff[3] != 0x000F)               // if no coeffvicients yet
      {                                     // load the default values
            coeff[0] = (unsigned short) Def_ADC_Mult;
            coeff[1] = (unsigned short) Def_ADC_Div;
            coeff[2] = (unsigned short) Def_ADC_Offset;
            coeff[3] = STORAGE_FLAG;
            EraseFlash(coeff_add, CO_EFF_SIZE);
            WriteFlash(coeff, coeff_add, CO_EFF_SIZE);
      }
      ADC_Mult = (byte) coeff[0];
      ADC_Div = (byte) coeff[1];
      LgADC_Div = (byte) log2(ADC_Div);
      ADC_Offset = (byte) coeff[2];
      ADCCoeffSend();

      return;

}     // Init


//**************************************************************************
//
// Company: Cranfield University.
//
// Copyright 2003 by Cranfield University.
//                   All rights reserved.
//
// Author: Gil Lopes
//
// Description: header file for vimain.c
//
// File: vimain.h
//
//**************************************************************************/

// Global Code Definitions
```

```
#ifndef _VIMAIN_H
#define _VIMAIN_H

#define DEBUG_VOLTAGE_SEND          // Activates sending voltages through the CAN
#undef DEBUG_VOLTAGE_SEND

#define DEBUG_AVERAGE_SEND          // Activates sending average voltages through the
                                    // CAN
#undef DEBUG_AVERAGE_SEND

#define DEBUG_OSCAVG_SEND           // Activates sending average voltages through the
                                    // CAN when oscillating
#undef DEBUG_OSCAVG_SEND

#define DEBUG_PROFILE               // Activates toggle on GPIO pin 6 to profile the
                                    // CAN interupts
#undef DEBUG_PROFILE                // and pin 7 to profile the ADC interupts

#define TIMER0_ON                   // Activate or deactivates TIMER0 events (ID LED)
//#undef TIMER0_ON

#define TIMER1_ON                   // Activate or deactivates TIMER1 events (CAN)
#undef TIMER1_ON                    // must always be set to OFF (undefined), only ON
                                    // when Arc starts or for debugging purposes

#define TIMER2_ON                   // Activate or deactivates TIMER2 events (ADC)
#undef TIMER2_ON                    // must always be set to OFF (undefined), only ON
                                    // when Arc starts or for debugging purposes

// Include Files
#include "56803.h"
#include "ext_reg.h"
#include "gpio.h"                   // include routines to easily manipulate GPIO
#include "flashlib.h"               // include flash routines
#include <math.h>                   // math routines (mainly for log2)

//***********************************************************************/
// Program Defines
/***********************************************************************/

// LED definitions
#define LED          0x0020         // Port E - Pin 6 (Pins from 1 to 8)
#define TurnOnLED    asm (bfset #LED,X:GPIO_E_DR)
#define TurnOffLED   asm (bfclr #LED,X:GPIO_E_DR)
#define ToggleLED    asm (bfchg #LED,X:GPIO_E_DR)

// GPIO Port A definitions
#define GPIO_A_Pins                 0x00f0  // Define Pins 1-4 Inputs (Rotary Switch)
```

```
                                                  // and 5-8 Outputs (7,8 Debug)
#define GPIO_A_Pins_Mask    0x00cf          // Which pins to change (1-4,6,7)

// Counter definitions (31250 (0x7a12) count ticks = 1/10 sec with IPBus/128 defined)
#define CountDown0   0xf424                 // 62500 count ticks = 1/5 sec with
                                            // IPBus/128 defined
#define CountDown1   0x7a12                 // (0x7a12) 31250 count ticks = 1/10 sec
                                            // with IPBus/128 defined
#define CountDown2   0x001f                 // 10080.6 samples/sec (closest number to
                                            // 10Ksamples/sec)
#define Zero         0x0000                 // sometimes very useful

// ADC definitions
#define BufferSize          1024            // For average calculation
#define LgBufferSize        10              // Log to base 2 of BufferSize
#define Def_ADC_Mult 24                     // defines the default Multiplier to
                                            // scale ADC values to voltage
#define Def_ADC_Div         16              // defines the default Divisor to scale
                                            // ADC values to voltage
#define Def_LgADC_Div       4               // Log to base 2 of default Divisor
#define Def_ADC_Offset      -30             // defines the default Offset to scale
                                            // ADC values to voltage
#define coeff_add           0x0100          // where in flash the Voltage conversion
                                            // co-effs are stored
#define CO_EFF_SIZE  4                      // number of conversion co-effs (Mult,
                                            // Div, Offset, and Storage Flag)
#define STORAGE_FLAG        0x000F          // storage flag to mark that co-effs have
                                            // been placed in flash
#define Initial_Trigger     0x00FF          // when current goes over this value an
                                            // arc was triggered
#define Timer_Trigger       0x4E20          // 2 seconds of samplings for triggering


// Control Algorithm definitions
#define PID_Divisor         32              // divisor for height calculations (lower
                                            // the values higher the torch movements)
#define LgPID_Divisor       3               // Log to base 2 of PID_Divisor
#define LgPID_H_Divisor     4               // Log to base 2 of PID Horizontal
                                            // Divisor
#define Abnormal_Tol 1000                   // Window tolerance for abnormal arc
                                            // behaviour (+10V, -10V from defined
                                            // average)

//G Status of process
#define S_Stop                      0x00    // Process is stopped
#define S_Start                     0x01    // Process received a start from pendant
#define S_Ignition          0x02            // Process detected voltage for ignition
                                            // (above average value)
#define S_Arc               0x03            // Process detected arc voltage (below
```

```
                                                // average after ignition)
#define Err_Window          0x400               // For 10KHz sampling and pulse freq of
                                                // 50Hz=2 pulse width in error (hi or lo)

// CAN definitions
#define ID0_SetVoltageMsg   0x6c                // from the pendant, defines the average
                                                // voltage in the message body
#define ID0_OS_P_STATUS             0x68        // from the pendant, defines start/stop
                                                // of arc
#define ID0_CF_ThruArcTrac  0x50                // from an adminstrator, define new
                                                // conversion co-effs.
#define ID0_CAN_Tch_Pos             0x74        // from the BUG msg=0x03Ax indicating one
                                                // of the three torch positions
#define ID0_SP_HorzAd               0x4a        // Torch In/Out message (SP_HorzAd
                                                // ID=0x25x)
#define CAN_Msg_Ready_ON    0x01                // use this define for debugging purposes
                                                // (1 sends a message, 0 doesn't send the
                                                // message)
#define CAN_H_Step_In               0x0005
#define CAN_H_Step_Out              0xFFFA

// Torch definitions
#define T_Out                       0x01        // Torch is in outter position
#define T_In                        0xff        // Torch is in inner position
#define T_Mid                   0x00      // Torch is in the middle

// Global definitions
#define byte unsigned char                      // although the processor only work with
                                                // 16bit minimum variables,
#define word unsigned int                       // the byte definition could be useful
                                                // to better understand the program
#define dwordunsigned long
//***********************************************************************/
// Constants

// Global Variable Declarations
static int i,j;                                 // universal temporary counters

static signed char scCounter0;                  // for being used in Timer0 counter (LED)
static byte bDevNum;                            // device number (0-3)
static byte bCAN_id1;                           // 2nd byte value for the CAN id
                                                // registers in DSP -- its the devnum
                                                // left shifted 5
static byte bCANSpeed;                          // CAN speed (0=500k, 1=250k, 2=125k,
                                                // 3=1M)
static byte bTPos;                              // torch position from bug
                                                // (out=1;in=255;mid=0;default=2)
static word ADC_V_Read;                         // ADC instant voltage read
static word ADC_I_Read;                         // ADC instant current read
```

```
static word ADC_Timer;              // Waits defined time to avoid initial
                                    // pulse disturbance
static word ADC_V_Average;          // ADC resultant calculated average
                                    // (After scaling)
static word ADC_V_Average_RAW;      // ADC resultant calculated average
                                    // before scaling
static word ADC_Last_Avg;           // previous position average

static word ADC_V_Peak;
static word ADC_V_In;
static word ADC_V_Out;
static word ADC_I_Max;
static word ADC_I_Trigger;

static dword ADC_V_Sum;             // Sum variable for average calculation

static word CAN_V_Average;          // Average voltage defined from Pendant
                                    // through a CAN message
static word CAN_Vertical;           // carries the new height value to send
                                    // to the bug
static word CAN_Horizontal;         // Horizontal value to be sent to torch
                                    // horizontal adjustment

static word S_Process;              // Process Status
static word Err_Hi;                 // Error of constantly high voltage read
static word Err_Lo;                 // Error of constantly low voltage read

// ADC Conversion Constants
static byte ADC_Mult;               // Multiplier to scale ADC values to
                                    // voltage
static byte ADC_Div;                // Divisor to scale ADC values to voltage
static byte LgADC_Div;              // Log to base 2 of Divisor
static byte ADC_Offset;             // Offset to scale ADC values to voltage
// Function Prototypes
void Timer0(void);                  // LED events
void Timer1(void);                  // CAN events
void Timer2(void);                  // ADC events
void CAN_Receive (void);            // CAN message receive events
void CAN_Transmit (void);           // CAN message transmit events
void ADC_Complete (void);           // ADC completed sampling events
void ADCAverageSend (void);         // ADC_CAN send averages through CAN
void ADCVoltageSend (void);         // ADC_CAN send voltages through CAN
void ADCCoeffSend (void);           // ADC_CAN send voltages through CAN
void TorchPos (void);               // Torch positioning routine
void init(void);                    // Init routine to initializes whole
                                    // hardware

#endif
```

Appendix D. Controller Area Network monitoring program source code

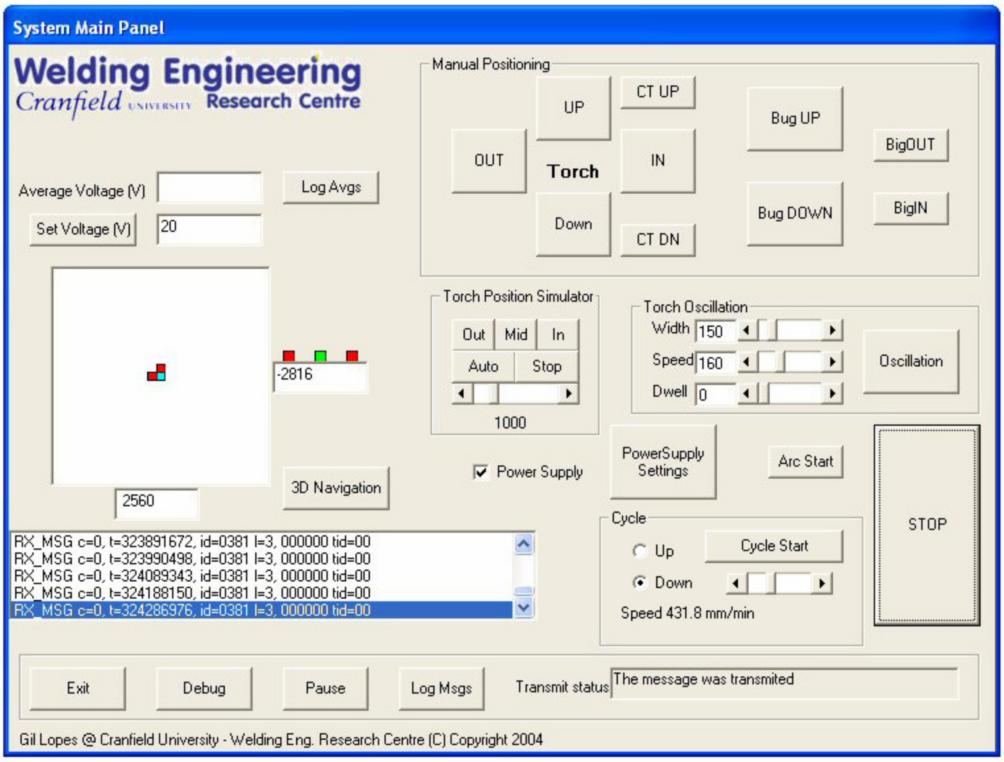
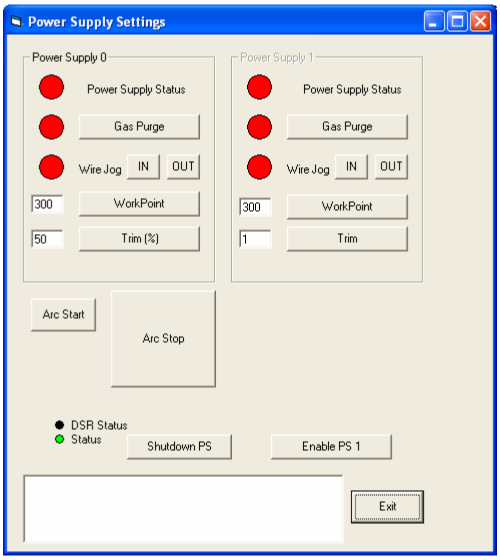


Figure D.1 – CAN Monitoring program - Main window



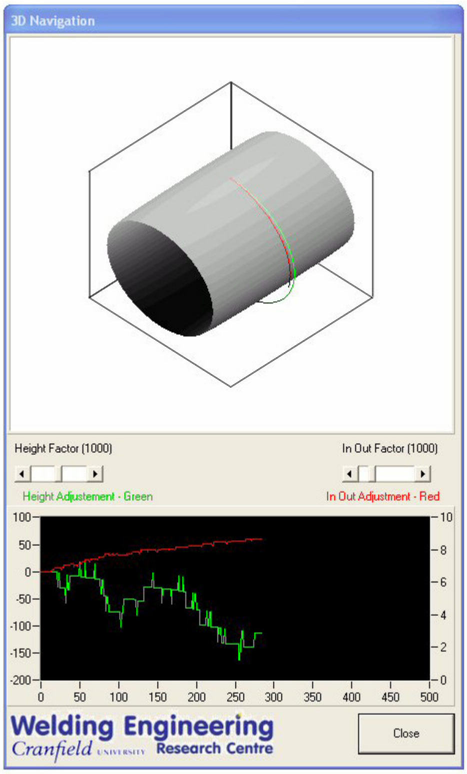


Figure D.2 - Power Supply control window (left) and torch offset error graphs (right)

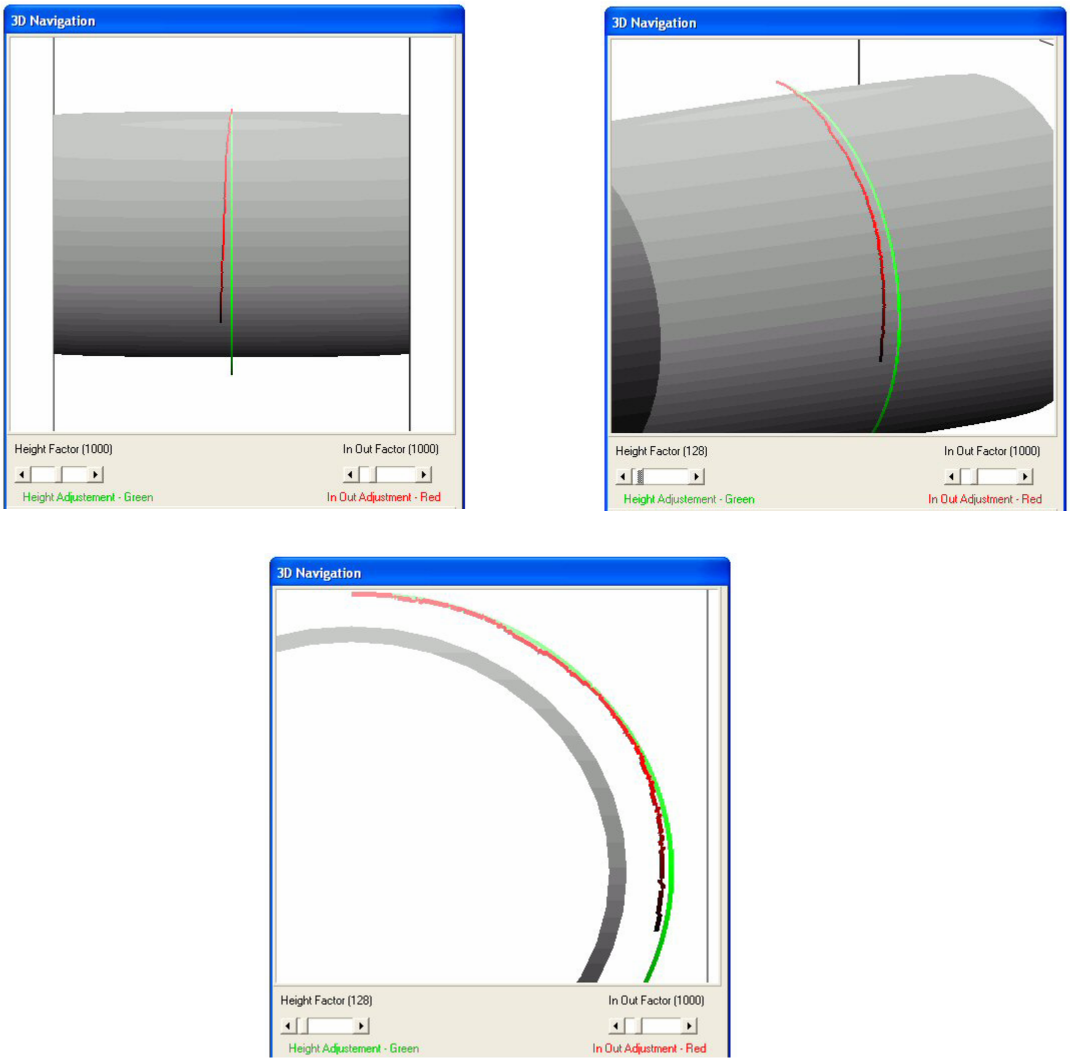


Figure D.3 - Different perspectives of the torch offset error 3D graph window

Figure D.4 – CAN monitoring program Visual Basic source code

```
Module RMS.BAS Code

Public Const Torus_Points = 40
Public Const Weld_Points = 500
Public Const Pi = 3.1415926535
Public Const Height_Comp = 1000
Public Const In_Out_Comp = 1000
Public Const Torus_Size = 30
Public Const Seam_Size = 35

Public tmessage_array(8) As String
Public tmessage_ID As String
Public tmessage_DLC As Integer
Public tmessage_extend As Boolean

Public message_array(8) As Integer
Public message_size As Integer

Public torch_pos, torch_dir As Integer

Public LogAvgs As Boolean
Public LogMsgs As Boolean

Public cycle_start As Boolean

Public Seam_x1(Weld_Points)
Public Seam_y1(Weld_Points)
Public Seam_z1(Weld_Points)
Public Seam_t1(Weld_Points)

Public Weld_Position As Integer
Public Torch_Height_Sum As Integer
Public Torch_In_Out_Sum As Integer

Public Torch_Height(Weld_Points) As Integer
Public Torch_In_Out(Weld_Points) As Integer

Public Sub Parse_Message(evstr As String)
Dim i As Integer
Dim strTemp1 As String * 20

    message_size = Val(Mid(evstr, InStr(evstr, "l=") + 2, 1))
    strTemp1 = Mid(evstr, InStr(evstr, "l=") + 5, message_size * 2)

    For i = 0 To message_size - 1
       message_array(i) = Val("&H" + Mid(strTemp1, (i * 2) + 1, 2))
```

```
    Next i

End Sub

Public Sub Process_Panel(evstr As String) ' Process all incoming messages to use just
the needed ones
Dim i, l As Integer
Dim j, k As Double
Dim strTemp1 As String
Dim evstr_log As String * 100

    If LogMsgs Then
        Open App.Path + "\MsgsLog.txt" For Append As #1
            evstr_log = Trim(evstr)
            Print #1, Date$ + vbTab + Time$ + vbTab + evstr_log
        Close #1
    End If

   If InStr(evstr, "id=00A4") <> 0 Then ' Voltage Average
        Parse_Message (evstr)
        j = Val(Trim(Str(message_array(1) * 256 + message_array(2)))) / 100
        FrontEnd.Text1.Text = Str(j)
        If LogAvgs Then
            Open App.Path + "\AvgsLog.txt" For Append As #1
                Print #1, Date$ + vbTab + Time$ + vbTab + Str(j)
            Close #1
        End If
   End If

   If InStr(evstr, "id=0261") <> 0 Then ' Torch Vertical Adjustment from the VI-Sense
        Parse_Message (evstr)
        j = Val(Trim(Str(message_array(1))))
        j = j * 256
        j = (j + message_array(2))
        If (j And &H8000) Then
            j = j - 65536
        End If

        Torch_Height_Sum = Torch_Height_Sum + (j / 100)

        If j > 0 Then
            FrontEnd.Shape1.Top = 1200 - (j / 25)
            FrontEnd.Shape1.Height = 100 + (j / 25)
        Else
            FrontEnd.Shape1.Top = 1200 ' + (j / 25)
            FrontEnd.Shape1.Height = 100 - (j / 25)
        End If
        FrontEnd.Text3.Text = Str(j)
   End If
```

```
    If InStr(evstr, "id=0251") <> 0 Then ' Torch Horizontal Adjustment from the VI-Sense
         Parse_Message (evstr)
         j = Val(Trim(Str(message_array(1))))
         j = j * 256
         j = (j + message_array(2))
         If (j And &H8000) Then
             j = j - 65536
         End If

         Torch_In_Out_Sum = Torch_In_Out_Sum + (-j / 1000)

         If j > 0 Then
             FrontEnd.Shape6.Left = 1200 '- (j / 25)
             FrontEnd.Shape6.Width = 100 + (j / 25)
         Else
             FrontEnd.Shape6.Left = 1200 + (j / 25)
             FrontEnd.Shape6.Width = 100 - (j / 25)
         End If
         FrontEnd.Text4.Text = Str(j)
    End If

    If InStr(evstr, "id=03A1") <> 0 Then ' Torch position from the BUG
         Parse_Message (evstr)
         j = Val(Trim(Str(message_array(1))))
         FrontEnd.Shape3.FillColor = &HFF&
         FrontEnd.Shape4.FillColor = &HFF&
         FrontEnd.Shape5.FillColor = &HFF&
         Select Case (j)
         Case 1: FrontEnd.Shape3.FillColor = &HFF00&
         Case 0: FrontEnd.Shape4.FillColor = &HFF00&
         Case 255: FrontEnd.Shape5.FillColor = &HFF00&
         Case Else
         End Select
    End If



End Sub


Public Sub Transmit_Message()
Dim TmpMsg As vbMsg
Dim j As Integer
Dim subStr As String
Dim strLen As Integer
Dim transmitID As String

transmitID = Hex$(Val("&h" & tmessage_ID) And &H7FFFFFFF)
```

```
If (Main.extended.Value) Then
  transmitID = Hex$(Val("&h" & tmessage_ID) Or &H80000000)
End If

ev.tag = v_TRANSMIT_MSG

j = 0
strLen = Len(transmitID)
For ii = strLen - 1 To 1 Step -2
  subStr = "&h" & Mid$(transmitID, ii, 2)
  TmpMsg.idBytes(j) = Val(subStr)
  j = j + 1
Next ii

If (strLen Mod 2) <> 0 Then
  subStr = "&h" & Mid$(transmitID, 1, 1)
  TmpMsg.idBytes(j) = Val(subStr)
End If

For aa = 0 To tmessage_DLC
  TmpMsg.data(aa) = Val("&h" & tmessage_array(aa))
Next
TmpMsg.dlc = tmessage_DLC 'Val("&h" & Main.dlc.Text)
If TmpMsg.dlc > 8 Then
  TmpMsg.dlc = 8
End If
TmpMsg.flags = 0

ev = Build_vbEvent_tagData_vbMsg(ev, TmpMsg)

FrontEnd.Label3.Caption = "Transmit a message"

vErr = vbTransmit(gPortHandle, chanMask, ev)
If vErr Then Fehler

' for checking if a message was transmited
transmited = True
transmitCounter = 10 ' wait 100ms for answer
End Sub
```

Module PS.BAS code

```
Public Declare Function PS_Functions Lib "PS.dll" Alias "PS_Funcs" _
    (ByVal func_name As Integer, ByVal fParam0 As Integer, ByVal fParam1 As Integer) As
Integer

Public stat As Integer
Public GasPurge(1) As Boolean
Public WireJog(1) As Boolean
Public PS_Active(1) As Boolean
Public PS1_enabled As Boolean
Public PS_used As Integer
Public Status_LED As Boolean
Public Arc_Start As Boolean
Public PS_Power_State As Boolean

' PS_Functions Description
' Function Number/Name              fParam0            fParam1              Return
'
========================================================================================
' 1 / Test DLL                     -------             ------
' 2 / Startup Power Supply         -------             ------
' 3 / PS Status Update           Power Supply number   ------
' 4 / Shutdown Power Supply      Power Supply number   ------
' 5 / Set Weld Mode              Power Supply number   Mode Number
' 6 / Arc Start/Stop             Power Supply number  1=Start;0=Stop
' 7 / Gas Purge                  Power Supply number  1=Gas  ;0=No Gas
' 8 / Wire Jog                   Power Supply number  1=Jog  ;0=No Jog
' 9 / Set Work Point             Power Supply number  Work Point Value
'10 / Trim Point                 Power Supply number  Trim Point Value
'11 / Read Actual WorkPoint      Power Supply number   ------               long
'
========================================================================================
' Power Supply number starts in 0 (zero)

Public Sub init_PS()
For i = 0 To 1
    PS_Active(i) = False
    GasPurge(i) = False
    WireJog(i) = False
Next i
PS1_enabled = False
PS_used = 1
PS_Power_State = False

stat = PS_Functions(2, 0, 0) 'Startup PS
If stat <= 0 Then
    PS_Active(0) = False
Else
```

```
    PS_Active(0) = True
    PS.PS0_StatusLED.FillColor = &HFF00&
End If

stat = PS_Functions(9, 0, 300)
stat = PS_Functions(10, 0, 50)

End Sub

Public Sub Timers(state As Boolean)
    PS.Timer2.Enabled = state
    FrontEnd.Timer1.Enabled = state
    FrontEnd.Timer2.Enabled = state
    Main.Timer1.Enabled = state
End Sub
```

Form FRONTEND.FRM code

```
Private Sub Command1_Click() ' Set Voltage Request
If Text2.Text <> "" Then
    tmessage_ID = "361"
    tmessage_DLC = 2
    tmessage_extend = False
    tmessage_array(0) = Trim(Hex$(Int((Val(Text2.Text) * 100) / 256)))
    tmessage_array(1) = Trim(Hex$(Int((Val(Text2.Text) * 100) Mod 256)))
    Call Transmit_Message
End If
End Sub

Private Sub Command10_Click() ' Torch Right (IN)
tmessage_ID = "251"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "0A"
Call Transmit_Message
End Sub

Private Sub Command11_Click() ' Oscillation
Call oscillation
End Sub

Private Sub Command12_Click() ' Simulates Torch arriving Outter region
tmessage_ID = "3a1"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "0"
tmessage_array(1) = "1"
Call Transmit_Message
End Sub

Private Sub Command13_Click() ' Simulates Torch arriving Middle region
tmessage_ID = "3a1"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "0"
tmessage_array(1) = "0"
Call Transmit_Message
End Sub

Private Sub Command14_Click() ' Simulates Torch arriving Inner region
tmessage_ID = "3a1"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "0"
```

```
tmessage_array(1) = "FF"
Call Transmit_Message
End Sub

Private Sub Command15_Click()
torch_pos = 0
torch_dir = 1
FrontEnd.Timer2.Enabled = True
End Sub

Private Sub Command16_Click()
FrontEnd.Timer2.Enabled = False
End Sub

Private Sub Command17_Click()
If Not LogAvgs Then
    LogAvgs = True
    FrontEnd.Command17.Caption = "Stop Log"
Else
    LogAvgs = False
    FrontEnd.Command17.Caption = "Log Avgs"
End If
End Sub

Private Sub Command18_Click()
If Not LogMsgs Then
    LogMsgs = True
    FrontEnd.Command18.Caption = "Stop Log"
Else
    LogMsgs = False
    FrontEnd.Command18.Caption = "Log Msgs"
End If
End Sub

Private Sub Command19_Click() ' Torch UP for CT (Contact Tip) changing
tmessage_ID = "261"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "01"
tmessage_array(1) = "EA"
Call Transmit_Message
End Sub

Private Sub Command20_Click() ' Torch DOWN for CT (Contact Tip) changing
tmessage_ID = "261"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "FE"
tmessage_array(1) = "15"
```

```
Call Transmit_Message
End Sub

Private Sub Command21_Click()
tmessage_ID = "251"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "32"
Call Transmit_Message
End Sub

Private Sub Command22_Click()
tmessage_ID = "251"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "FF"
tmessage_array(1) = "CD"
Call Transmit_Message
End Sub

Private Sub Command23_Click()
stat = PS_Functions(1, 0) 'Test PS DLL
PS.Show
End Sub

Private Sub Command24_Click()
Nav3D.Show
End Sub

Private Sub Command9_Click() ' Torch Left (OUT)
tmessage_ID = "251"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "FF"
tmessage_array(1) = "F5"
Call Transmit_Message
End Sub

Private Sub Command7_Click() ' Torch UP
tmessage_ID = "261"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "0A"
Call Transmit_Message
End Sub
```

```
Private Sub Command8_Click() ' Torch DOWN
tmessage_ID = "261"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "FF"
tmessage_array(1) = "F5"
Call Transmit_Message
End Sub

Private Sub Command2_Click() ' Arc Start
Weld_Position = 0
Torch_Height_Sum = 0
Torch_In_Out_Sum = 0
Nav3D.Output2D.ClearData

tmessage_ID = "341"
tmessage_DLC = 1
tmessage_extend = False
tmessage_array(0) = "2"
Call Transmit_Message

If PS_Value.Value = 1 Then
    Arc_Start = True
    'Call Timers(False)
    If PS_Active(0) Then
        stat = PS_Functions(6, 0, 1)
    End If
    If PS_used = 2 And PS_Active(1) Then
        stat = PS_Functions(6, 1, 1)
    End If
    PS.OutList.AddItem "PS_Status(contactor ON)= " & stat
    PS_Value.Enabled = False
    Call Timers(True)
    'Timer2.Enabled = False
End If
Nav3D.Timer1.Enabled = True
End Sub

Private Sub Command5_Click() ' BUG UP
tmessage_ID = "201"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "00"
Call Transmit_Message
tmessage_ID = "201"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "01"
```

```
tmessage_array(1) = "F8"
Call Transmit_Message
End Sub

Private Sub Command6_Click() ' BUG DOWN
tmessage_ID = "201"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "00"
Call Transmit_Message
tmessage_ID = "201"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "FE"
tmessage_array(1) = "07"
Call Transmit_Message
End Sub

Private Sub Command3_Click() ' Cycle Start
Dim i As Integer

cycle_start = True

tmessage_ID = "201"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "00"
Call Transmit_Message

If (FrontEnd.Option1) Then
    tmessage_ID = "201"
    tmessage_DLC = 2
    tmessage_extend = False
    tmessage_array(0) = Trim(Hex$(Int(HScroll1.Value / 256)))  '00
    tmessage_array(1) = Trim(Hex$(Int(HScroll1.Value Mod 256))) 'FA
    Call Transmit_Message
Else
    i = Not (HScroll1.Value)
    tmessage_ID = "201"
    tmessage_DLC = 2
    tmessage_extend = False
    tmessage_array(0) = Trim(Right(Hex$(Int(i / 256)), 2)) 'FF
    tmessage_array(1) = Trim(Right(Hex$(Int(i Mod 256)), 2)) '05
    Call Transmit_Message
End If
    Label7.Caption = "Speed " & HScroll1.Value
End Sub
```

```
Private Sub Command4_Click()
' Process and send STOP message (Arc, motion, etc)

cycle_start = False

If Arc_Start Then
    'Call Timers(False)
    If PS_Value.Value = 1 Then
        If PS_Active(0) Then
            stat = PS_Functions(6, 0, 0)
        End If
        If PS_used = 2 And PS_Active(1) Then
            stat = PS_Functions(6, 1, 0)
        End If
    End If
    PS.OutList.AddItem "PS_Status(contactor OFF)= " & stat
    PS_Value.Enabled = True
    Arc_Start = False
    'Call Timers(True)
    Timer3.Enabled = True

    tmessage_ID = "341" 'Arc Stop
    tmessage_DLC = 1
    tmessage_extend = False
    tmessage_array(0) = "3"
    Call Transmit_Message

Else
    tmessage_ID = "341" 'Arc Stop
    tmessage_DLC = 1
    tmessage_extend = False
    tmessage_array(0) = "3"
    Call Transmit_Message

    tmessage_ID = "201" 'Stop BUG
    tmessage_DLC = 2
    tmessage_extend = False
    tmessage_array(0) = "00"
    tmessage_array(1) = "00"
    Call Transmit_Message

    tmessage_ID = "231" ' Stop oscillation
    tmessage_DLC = 6
    tmessage_extend = False
    tmessage_array(0) = "0"
    tmessage_array(1) = "0"
    tmessage_array(2) = "0"
    tmessage_array(3) = "0"
    tmessage_array(4) = "0"
```

```
    tmessage_array(5) = "0"
    Call Transmit_Message

End If

Nav3D.Timer1.Enabled = False

End Sub

Private Sub Debug_Click()

    Main.OnOffline.Caption = "Open&Driver"
    Main.OnOffline_Click
    Main.Show

End Sub

Private Sub Exit_Click()
vErr = vbClosePort(gPortHandle)
gPortHandle = INVALID_PORTHANDLE
vErr = vbCloseDriver
If Main.moreInfo.Value Then
  Main.Output.AddItem ">>> Close Port"
  Main.Output.AddItem ">>> Close Driver"
End If

Call Timers(False)

Unload Main
Unload FrontEnd
Unload PS

stat = PS_Functions(4, 1, 0) 'Shutdown PS1 and PS2

End Sub

Private Sub Form_Activate()
FE_Activated = True
MN_Activated = False
Main.Timer1.Enabled = False
FrontEnd.Timer1.Enabled = True
End Sub

Private Sub Form_Load()

Declarations

h = CreateEvent(vbNullString, False, False, vbNullString)
```

```
vErr = InitDriver
If vErr Then Fehler

Call init_PS

End Sub

Private Sub HScroll1_Change()
Dim i As Integer

If cycle_start Then
    If (FrontEnd.Option1) Then
        tmessage_ID = "201"
        tmessage_DLC = 2
        tmessage_extend = False
        tmessage_array(0) = Trim(Hex$(Int(HScroll1.Value / 256)))  '00
        tmessage_array(1) = Trim(Hex$(Int(HScroll1.Value Mod 256))) 'FA
        Call Transmit_Message
    Else
        i = Not (HScroll1.Value)
        tmessage_ID = "201"
        tmessage_DLC = 2
        tmessage_extend = False
        tmessage_array(0) = Trim(Right(Hex$(Int(i / 256)), 2)) 'FF
        tmessage_array(1) = Trim(Right(Hex$(Int(i Mod 256)), 2)) '05
        Call Transmit_Message
    End If
End If
Label7.Caption = "Speed " & HScroll1.Value * 2.54 & "mm/min"
End Sub

Private Sub HScroll2_Change() ' Speed Change
Call oscillation
End Sub

Private Sub HScroll3_Change() ' Width Change
Call oscillation
End Sub

Private Sub HScroll4_Change() ' Dwell Change
Call oscillation
End Sub

Private Sub HScroll5_Change()
FrontEnd.Label9.Caption = FrontEnd.HScroll5.Value
FrontEnd.Timer2.Interval = FrontEnd.HScroll5.Value
End Sub
```

```
Private Sub Pause_Click()
    If FE_Activated = False Then
        FE_Activated = True
        Pause.Caption = "Pause"
    Else
        FE_Activated = False
        Pause.Caption = "Continue..."
    End If
End Sub

Private Sub Text5_Change()
If Text5.Text <> "" Then
    HScroll2.Value = Text5.Text
End If
End Sub

Private Sub Text6_Change()
If Text6.Text <> "" Then
    HScroll3.Value = Text6.Text
End If
End Sub

Private Sub Text7_Change()
If Text7.Text <> "" Then
    HScroll4.Value = Text7.Text
End If
End Sub

Private Sub Timer1_Timer()

If Not FE_Activated Then
   Exit Sub
End If

Dim evstr As String * 255
Dim Tmp As Long
Dim timestamp As Double

vErr = vbReceive1(gPortHandle, pEvent)

' check if a message was transmited
If (transmited) Then
  transmitCounter = transmitCounter - 1
  If transmitCounter = 0 Then
    FrontEnd.Label3.Caption = "Couldn't transmit the message! "
    transmited = False
  End If
End If
```

```
If vErr = VSUCCESS Then
   vErr = vbGetEventString(pEvent, evstr)
   FrontEnd.Output.AddItem evstr

    Call Process_Panel(evstr) 'Refresh FrontPanel values with new received ones

   ' check if a message was transmited
   If (transmited) Then
     If (pEvent.tagData(4) And MSGFLAG_TX) Then
       FrontEnd.Label3.Caption = "The message was transmited "
       transmited = False
     End If
   End If

   messageCount = messageCount + 1
   If pEvent.timestamp Then
      timestamp = pEvent.timestamp
      If (timestamp < 0) Then
        timestamp = timestamp + 4294967296#    'fix signed reperesentation of VB as
unsigned
      End If
      If lastTime > timestamp Then
        FrontEnd.Output.AddItem "!!! Time decreasing !!!  DeltaT = -" & lastTime -
timestamp
      End If
      lastTime = timestamp
   End If
   If Get_vbEvent_tagData_vbMsg(pEvent).flags And MSGFLAG_OVERRUN Then
      overrunCount = overrunCount + 1
   End If
ElseIf vErr <> VERR_QUEUE_IS_EMPTY Then
   Fehler
Else
   Exit Sub
End If

lc = FrontEnd.Output.ListCount
FrontEnd.Output.ListIndex = lc - 1
If lc > 30000 Then FrontEnd.Output.Clear

End Sub

Sub oscillation()
tmessage_ID = "231"
tmessage_DLC = 6
tmessage_extend = False
tmessage_array(0) = Trim(Hex$(Int(HScroll2.Value / 256)))
tmessage_array(1) = Trim(Hex$(Int(HScroll2.Value Mod 256)))
tmessage_array(2) = Trim(Hex$(Int(HScroll3.Value / 256)))
```

```
tmessage_array(3) = Trim(Hex$(Int(HScroll3.Value Mod 256)))
tmessage_array(4) = Trim(Hex$(Int(HScroll4.Value / 256)))
tmessage_array(5) = Trim(Hex$(Int(HScroll4.Value Mod 256)))
Text5.Text = HScroll2.Value
Text6.Text = HScroll3.Value
Text7.Text = HScroll4.Value
Call Transmit_Message
End Sub

Private Sub Timer2_Timer()

If torch_pos = 0 Then
    torch_pos = torch_pos + torch_dir
    tmessage_array(1) = "0"

ElseIf torch_pos = 1 Then
    torch_dir = -1
    torch_pos = torch_pos + torch_dir
    tmessage_array(1) = "1"

ElseIf torch_pos = -1 Then
    torch_dir = 1
    torch_pos = torch_pos + torch_dir
    tmessage_array(1) = "FF"
End If

tmessage_ID = "3a1"
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "0"
Call Transmit_Message

End Sub

Private Sub Timer3_Timer()
tmessage_ID = "341" 'Arc Stop
tmessage_DLC = 1
tmessage_extend = False
tmessage_array(0) = "3"
Call Transmit_Message

tmessage_ID = "201" 'Stop BUG
tmessage_DLC = 2
tmessage_extend = False
tmessage_array(0) = "00"
tmessage_array(1) = "00"
Call Transmit_Message

tmessage_ID = "231" ' Stop oscillation
```

```
tmessage_DLC = 6
tmessage_extend = False
tmessage_array(0) = "0"
tmessage_array(1) = "0"
tmessage_array(2) = "0"
tmessage_array(3) = "0"
tmessage_array(4) = "0"
tmessage_array(5) = "0"
Call Transmit_Message
Timer3.Enabled = False
End Sub
```

Form NAV3D.FRM

```
Private Sub Command1_Click()
Nav3D.Hide
End Sub

Private Sub Form_Load()
Dim tx(Torus_Points, Torus_Points)
Dim Ty(Torus_Points, Torus_Points)
Dim Tz(Torus_Points, Torus_Points)
Dim Tt(Torus_Points)

Dim Ix(Weld_Points)
Dim Iy(Weld_Points)
Dim Iz(Weld_Points)

    For i = 0 To Torus_Points
        Tt(i) = (i - (Torus_Points / 2)) / (Torus_Points / 2) * Pi
    Next i

    For i = 0 To Torus_Points
        For j = 0 To Torus_Points
            tx(i, j) = Sin(Tt(j))
            Ty(i, j) = (Cos(Tt(j)) + Torus_Size) * Cos(Tt(i))
            Tz(i, j) = (Cos(Tt(j)) + Torus_Size) * Sin(Tt(i))
        Next j
    Next i

    ' Plot torus data
    Output3D.Plots(1).Plot3DParametricSurface tx, Ty, Tz

    For i = 0 To Weld_Points
        Seam_t1(i) = (Pi / 2) - (i / (Weld_Points / Pi))
    Next i

    For i = 0 To Weld_Points
        Ix(i) = 0
        Iy(i) = (Cos(Seam_t1(i)) + Seam_Size) * Cos(Seam_t1(i))
        Iz(i) = (Cos(Seam_t1(i)) + Seam_Size) * Sin(Seam_t1(i))
    Next i
    Output3D.Plots(2).Plot3DCurve Ix, Iy, Iz

    Nav3D.Height_Label.Caption = "Height Factor (" & Height_Comp & ")"
    Nav3D.In_Out_Label.Caption = "In Out Factor (" & In_Out_Comp & ")"

    Nav3D.Height_Scroll.Value = Height_Comp
    Nav3D.In_Out_Scroll.Value = In_Out_Comp

    Nav3D.Output2D.Plots(1).XAxis.Minimum = 0
```

```
    Nav3D.Output2D.Plots(1).XAxis.Maximum = Weld_Points

End Sub


Private Sub Height_Scroll_Change()
    Call Factor_Change
End Sub

Private Sub In_Out_Scroll_Change()
    Call Factor_Change
End Sub

Sub Factor_Change()
Dim new_index As Integer

    Nav3D.Height_Label.Caption = "Height Factor (" & Nav3D.Height_Scroll.Value & ")"
    Nav3D.In_Out_Label.Caption = "In Out Factor (" & Nav3D.In_Out_Scroll.Value & ")"

    For i = 0 To Weld_Points
        If i < Weld_Position Then
            Seam_x1(i) = Torch_In_Out(i) / Nav3D.In_Out_Scroll.Value
            Seam_y1(i) = (Cos(Seam_t1(i)) + Seam_Size + (Torch_Height(i) /
Nav3D.Height_Scroll.Value)) * Cos(Seam_t1(i))
            Seam_z1(i) = (Cos(Seam_t1(i)) + Seam_Size + (Torch_Height(i) /
Nav3D.Height_Scroll.Value)) * Sin(Seam_t1(i))
            new_index = i
        Else
            Seam_x1(i) = Seam_x1(new_index)
            Seam_y1(i) = Seam_y1(new_index)
            Seam_z1(i) = Seam_z1(new_index)
        End If
    Next i
    Output3D.Plots(3).Plot3DCurve Seam_x1, Seam_y1, Seam_z1

End Sub

Private Sub Timer1_Timer()

    If Weld_Position <= Weld_Points Then
        Torch_Height(Weld_Position) = Torch_Height_Sum
        Torch_In_Out(Weld_Position) = Torch_In_Out_Sum

        Seam_x1(Weld_Position) = Torch_In_Out_Sum / Nav3D.In_Out_Scroll.Value
        Seam_y1(Weld_Position) = (Cos(Seam_t1(Weld_Position)) + Seam_Size +
(Torch_Height_Sum / Nav3D.Height_Scroll.Value)) * Cos(Seam_t1(Weld_Position))
        Seam_z1(Weld_Position) = (Cos(Seam_t1(Weld_Position)) + Seam_Size +
(Torch_Height_Sum / Nav3D.Height_Scroll.Value)) * Sin(Seam_t1(Weld_Position))
```

```
        For i = Weld_Position + 1 To Weld_Points
            Seam_x1(i) = Seam_x1(Weld_Position)
            Seam_y1(i) = Seam_y1(Weld_Position)
            Seam_z1(i) = Seam_z1(Weld_Position)
        Next i

        Output3D.Plots(3).Plot3DCurve Seam_x1, Seam_y1, Seam_z1
        Weld_Position = Weld_Position + 1
    End If
    Nav3D.Output2D.Plots(1).ChartY Torch_Height_Sum
    Nav3D.Output2D.Plots(2).ChartY Torch_In_Out_Sum

End Sub
```

Form PS.FRM code

```
Private Sub Command1_Click()
    PS.Hide
End Sub

Private Sub PS_Power_Click()
If PS_Power_State = False Then
    PS_Power_State = True
    PS_Power.Caption = "Power ON PS"
    stat = PS_Functions(4, 1, 0)
Else
    PS_Power_State = False
    PS_Power.Caption = "Shutdown PS"
    stat = PS_Functions(2, 0, 0)
End If
End Sub

Private Sub PS0_Trim_Click()
i = CDbl(PS0_Trim_Val.Text)
stat = PS_Functions(10, 0, i) 'Trim
End Sub

Private Sub PS0_Workpoint_Click()
i = CDbl(PS0_Workpoint_Val.Text)
stat = PS_Functions(9, 0, i) 'WorkPoint
End Sub

Private Sub PS1_button_Click()
If PS1_enabled = False Then
    Frame2.Enabled = True
    PS1_button.Caption = "Disable PS 1"
    PS1_enabled = True
    PS_used = 2
Else
    PS1_enabled = False
    Frame2.Enabled = False
    PS1_button.Caption = "Enable PS 1"
    PS_used = 1
End If
End Sub

Private Sub PS0_GasPurge_Click()
    If GasPurge(0) = False Then
        stat = PS_Functions(7, 0, 1)
        GasPurge(0) = True
        PS0_GasPurgeLED.FillColor = &HFF00&
    Else
        stat = PS_Functions(7, 0, 0)
```

```
        GasPurge(0) = False
        PS0_GasPurgeLED.FillColor = &HFF&
    End If
End Sub

Private Sub PS0_WireJog_IN_Click()
    If WireJog(0) = False Then
        stat = PS_Functions(8, 0, -1)
        WireJog(0) = True
        PS0_WireJogLED.FillColor = &HFF00&
    Else
        stat = PS_Functions(8, 0, 0)
        WireJog(0) = False
        PS0_WireJogLED.FillColor = &HFF&
    End If
End Sub

Private Sub PS0_WireJog_OUT_Click()
    If WireJog(0) = False Then
        stat = PS_Functions(8, 0, 1)
        WireJog(0) = True
        PS0_WireJogLED.FillColor = &HFF00&
    Else
        stat = PS_Functions(8, 0, 0)
        WireJog(0) = False
        PS0_WireJogLED.FillColor = &HFF&
    End If
End Sub

Private Sub PS1_Trim_Click()
stat = PS_Functions(10, 1, Val(PS0_Trim_Val.Text))  'Trim
End Sub

Private Sub PS1_WireJog_IN_Click()
    If WireJog(1) = False Then
        stat = PS_Functions(8, 1, -1)
        WireJog(1) = True
        PS1_WireJogLED.FillColor = &HFF00&
    Else
        stat = PS_Functions(8, 1, 0)
        WireJog(1) = False
        PS1_WireJogLED.FillColor = &HFF&
    End If
End Sub

Private Sub PS1_WireJog_OUT_Click()
    If WireJog(1) = False Then
        stat = PS_Functions(8, 1, 1)
        WireJog(1) = True
```

```
        PS1_WireJogLED.FillColor = &HFF00&
    Else
        stat = PS_Functions(8, 1, 0)
        WireJog(1) = False
        PS1_WireJogLED.FillColor = &HFF&
    End If
End Sub

Private Sub PS1_GasPurge_Click()
    If GasPurge(1) = False Then
        stat = PS_Functions(7, 1, 1)
        GasPurge(1) = True
        PS1_GasPurgeLED.FillColor = &HFF00&
    Else
        stat = PS_Functions(7, 1, 0)
        GasPurge(1) = False
        PS1_GasPurgeLED.FillColor = &HFF&
    End If
End Sub

Private Sub PS1_Workpoint_Click()
stat = PS_Functions(9, 1, Val(PS0_Workpoint_Val.Text)) 'WorkPoint
End Sub

Private Sub Timer2_Timer()
If Status_LED Then
    StatusLED.FillColor = &HFF00&
    Status_LED = False
Else
    StatusLED.FillColor = &H0&
    Status_LED = True
End If

psi = PS.OutList.ListCount
PS.OutList.ListIndex = psi - 1
If psi > 30000 Then PS.OutList.Clear

stat = PS_Functions(3, 0, 0)
OutList.AddItem "PS_Status(Status reading PS0)= " & stat
If stat = 1 Then
    PS0_StatusLED.FillColor = &HFF00&
    PS_Active(0) = True
End If
If stat = 0 Then
    PS0_StatusLED.FillColor = &HFF&
    PS_Active(0) = False
End If
If PS_used = 2 Then
    stat = PS_Functions(3, 1, 0)
```

```
    OutList.AddItem "PS_Status(Status reading PS1)= " & stat
    If stat = 1 Then
        PS1_StatusLED.FillColor = &HFF00&
        PS_Active(1) = True
    End If
    If stat = 0 Then
        PS1_StatusLED.FillColor = &HFF&
        PS_Active(1) = False
    End If
End If
End Sub
```

Appendix E. Results of 2nd phase of experimentation

Table E.1 – Trials B1 to B4 setup and bead measurements

Run Number	Travel Speed (m/min)	Osc freq (Hz)	Osc Width (mm)	Initial CTWD (mm)	Voltage Set (V)	Number of Wires	Bead H. Dif A (mm)	Bead L. A (mm)	Bead H. Dif B (mm)	Bead L. B (mm)
1	0.76	8.33	2.5	13.5	21	2	15.2	190	0	0
2	0.76	8.33	2.5	13.5	19	2	11.2	120	0	0
3	0.76	8.33	2.5	13.5	21	2	8.4	120	11.2	70
4	0.76	8.33	6	13.5	21	2	5.4	120	8.2	70
							2.8	120	5.8	70

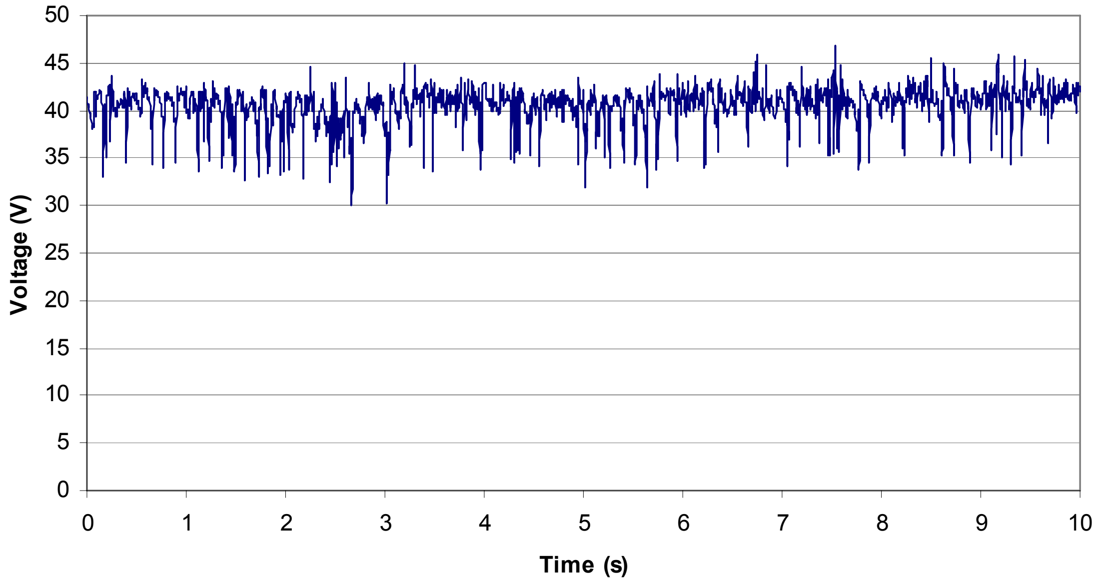


Figure E.1 – Peak voltages from trial B1

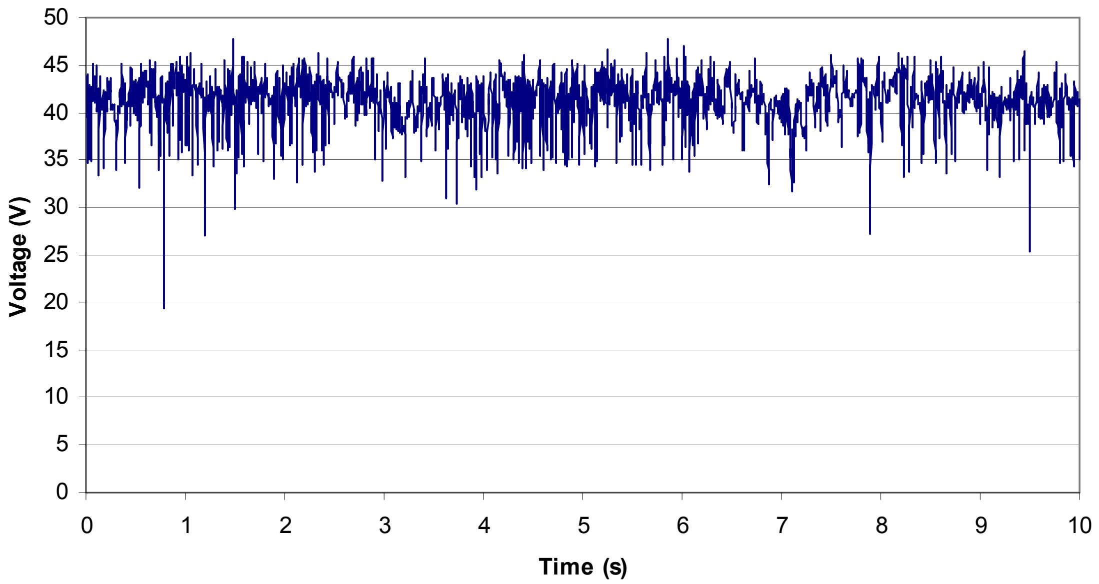


Figure E.2 – Peak voltages from trial B3

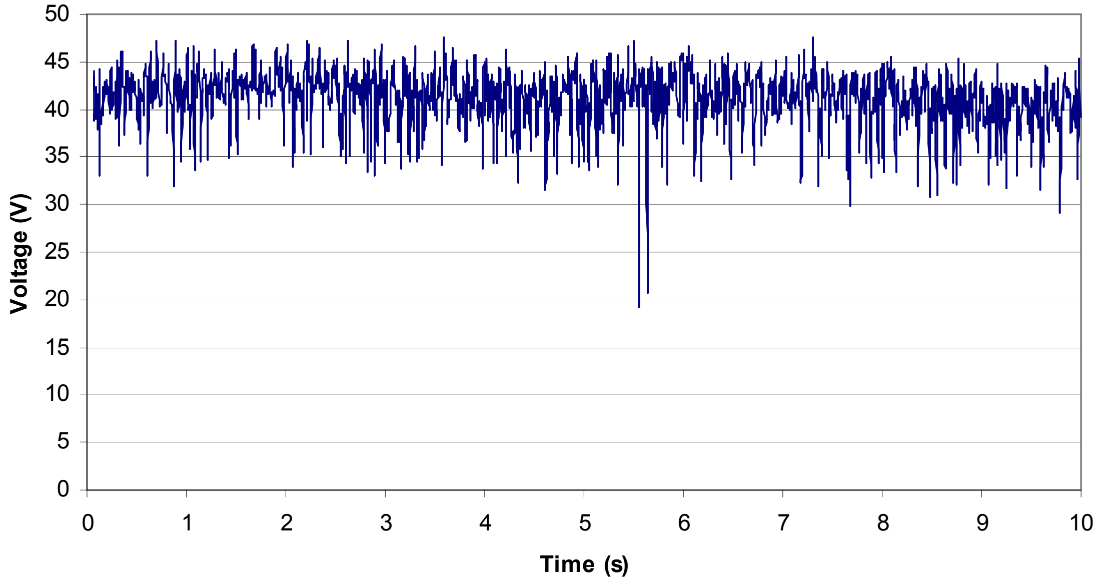


Figure E.3 – Peak voltages from trial B4

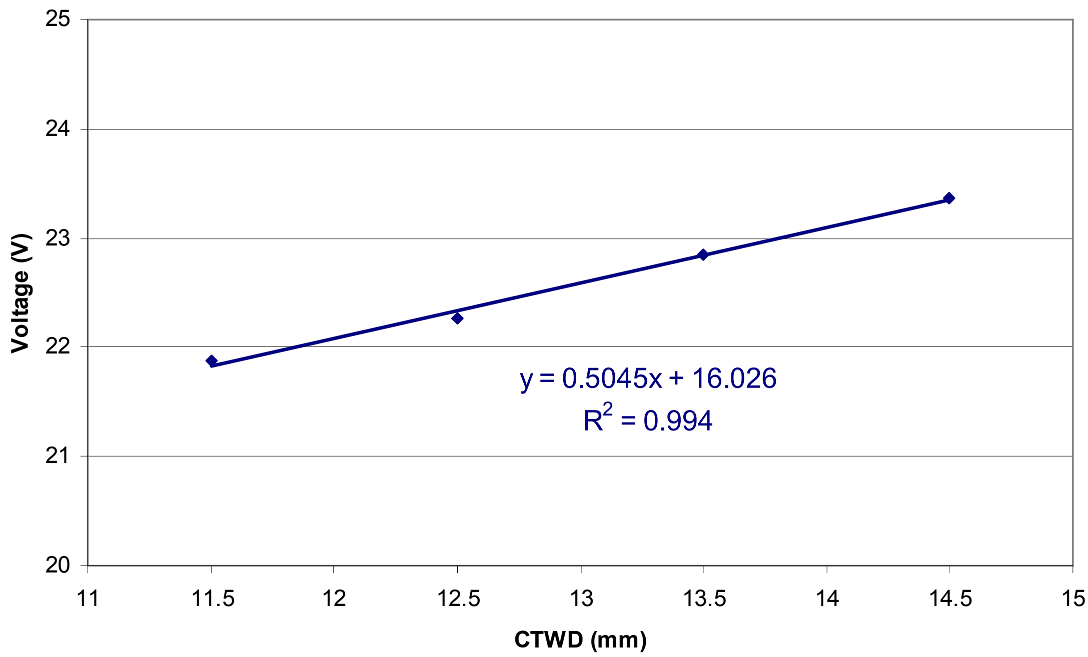


Figure E.4 – Voltage measurements at different CTWD values for 2nd phase of experimentation

Appendix F. TRIO BASIC motion controller source code

Figure F.1 - TRIO BASIC motion controller source code for 3rd phase of experimentation

```
'==========================================================
'Name:     Startup.bas
'Function: Startup sequence for axes setup
'------------------------------------------------------------------------
'Date:     08/04
'Author:   Gil Lopes
'------------------------------------------------------------------------
'          Copyright (c) 2004 Cranfield University
'          Welding Engineering Research Centre, Cranfield University
'          Cranfield, Bedford, MK43 0AL
'          Tel: +44-1234-750111 Fax: +44-1234-754109
'==========================================================
SETCOM(9600, 8, 1, 0, 1)
WDOG=ON

' set axis units
BASE(0) ' TRAVEL axis
UNITS=402.4 ' UNITS IN mm; speed = mm/sec
P_GAIN=1
I_GAIN=0
D_GAIN=50
OV_GAIN=0
VFF_GAIN=0
SPEED=5 ' mm/sec
ACCEL=10000
DECEL=10000
FE_LIMIT=200000
DEFPOS(0)
SERVO=ON

BASE(1) ' Oscillator axis
UNITS=2000 '1 FULL REV (MOVE(1) WILL BE ONE COMPLETE TURN)
P_GAIN=1
I_GAIN=0
D_GAIN=50
SPEED=5
ACCEL=100
DECEL=100
FE_LIMIT=1000
DEFPOS(0)
SERVO=ON
```

```
BASE(3) ' Width axis
UNITS=100000 '1 FULL REV (MOVE(1) WILL BE ONE COMPLETE TURN)
P_GAIN=1
I_GAIN=0
D_GAIN=50
SPEED=5
ACCEL=100
DECEL=100
FE_LIMIT=100
DEFPOS(0)
SERVO=ON


RUN "MAIN"

STOP
'==============================================================
' End of file
'==============================================================

'==============================================================
'Name:      Main.bas
'Function:  Main program
'------------------------------------------------------------------------
'Date:      08/04
'Author:    Gil Lopes
'------------------------------------------------------------------------
'           Copyright (c) 2004 Cranfield University
'           Welding Engineering Research Centre, Cranfield University
'           Cranfield, Bedford, MK43 0AL
'           Tel: +44-1234-750111 Fax: +44-1234-754109
'==============================================================

init:

' MAIN vars for global initialisations
width_min = 5.4 'Calibrated minimum Width in mm
width_max = 9.8 'Calibrated maximum Width in mm
step_min = 0 'Minimum step for platform down position
step_max = -17 'Maximum step for platform up position
enc_count_step = 100000 'Encoder counts per step
units_step = ((enc_count_step*step_max)-(enc_count_step*step_min))
units_mm = (width_min-width_max)
units_calc = units_step/units_mm
units_default = 100000 'Encoder counts per revolution

speed_min = 0 'Minimum achieved acceptable speed
```

```
speed_max = 25 'Maximum achieved acceptable speed
speed_start = 5 'Speed at boot

travel_speed = 0.3 'Travel Speed in m/min
travel_speed_max = 1.5 ' Maximum Travel Speed
travel_speed_min = 0.1 ' Minimum Travel Speed

'VR initialization
VR(1)=0 'Display options
VR(2)=speed_start 'Displayed Speed value
VR(3)=width_max 'Displayed Width value
VR(4)=0 'Keyboard strokes
VR(5)=0 'Process events and status
VR(6)=0 'KEYBOARD status
VR(7)=0 'PROCS status
VR(8)=0 'DISPLAY status
VR(9)=0 'INI running

VR(10)=width_min
VR(11)=width_max
VR(12)=units_calc 'UNITS for BASE(3)
VR(13)=units_default
VR(14)=speed_min
VR(15)=speed_max

VR(16)=0 ' Table length
VR(17)=0 ' Number of welds positions per plate
VR(18)=0 ' Actual weld position
VR(19)=travel_speed ' Travel Speed in m/min
VR(20)=travel_speed_max
VR(21)=travel_speed_min

' Main start
PRINT #4, "   CRANFIELD UNI.  "
PRINT #4, "         *         "
PRINT #4, "    WELDING ENG.   "
PRINT #4, "  RESEARCH CENTRE  "
WA(4000)

RUN "CLRSCR"
WA(1000)
RUN "INI"
WAIT UNTIL VR(9)=1
RUN "DISPLAY"

'Main Loop
begin:
key_in=0
```

```
GET #4, key_in
IF key_in<>0 THEN VR(4)=key_in
IF key_in=69 THEN GOTO init
IF VR(6)=0 THEN
   VR(6)=1
   RUN "KEYBOARD"
ENDIF
IF VR(7)=0 THEN
   VR(7)=1
   RUN "PROCS"
ENDIF
IF VR(8)=0 THEN
   VR(8)=1
   RUN "DISPLAY"
ENDIF
WA(1000)
GOTO begin

'==============================================================
'Name:      clrscr.bas
'Function:  Cleans up the screen
'------------------------------------------------------------------------
'Date:      08/04
'Author:    Gil Lopes
'------------------------------------------------------------------------
'           Copyright (c) 2004 Cranfield University
'           Welding Engineering Research Centre, Cranfield University
'           Cranfield, Bedford, MK43 0AL
'           Tel: +44-1234-750111 Fax: +44-1234-754109
'==============================================================


FOR i=1 TO 4
   PRINT#10, "                    "
NEXT i

STOP
```

```
'==========================================================
'Name:      Ini.bas
'Function:  handles the initialisation routine
'------------------------------------------------------------------------
'Date:      08/04
'Author:    Gil Lopes
'------------------------------------------------------------------------
'           Copyright (c) 2004 Cranfield University
'           Welding Engineering Research Centre, Cranfield University
'           Cranfield, Bedford, MK43 0AL
'           Tel: +44-1234-750111 Fax: +44-1234-754109
'==========================================================

PRINT #4, " Move platform  >  "
PRINT #4, " down to base   >  "
PRINT #4, " with the keys  >  "
PRINT #4, " and press Enter>  "

BASE(3)
SPEED=1
UNITS=VR(13)
SERVO=ON

begin:
key_in=0
GET #4, key_in

IF key_in=73 THEN
   SPEED=5
   UNITS=VR(12)
   DEFPOS(VR(11))
   WAIT UNTIL OFFPOS=0
   GOTO step1
ENDIF

IF key_in=81 THEN MOVE(-2)
IF key_in=82 THEN MOVE(-.5)
IF key_in=83 THEN MOVE(.5)
IF key_in=84 THEN MOVE(2)

GOTO begin

step1:
PRINT #4, " Move table to  >  "
PRINT #4, " start position >  "
PRINT #4, "                >  "
PRINT #4, " and press Enter>  "
```

```
BASE(0)
SPEED=10
SERVO=ON

travelstart:
key_in=0
GET #4, key_in

IF key_in=73 THEN
   DEFPOS(0)
   GOTO step2
ENDIF

IF key_in=81 THEN MOVE(-10)
IF key_in=82 THEN MOVE(-1)
IF key_in=83 THEN MOVE(1)
IF key_in=84 THEN MOVE(10)

GOTO travelstart

step2:
PRINT #4, " Move table to  >  "
PRINT #4, " end position   >  "
PRINT #4, "                >  "
PRINT #4, " and press Enter>  "

BASE(0)
SPEED=10
SERVO=ON

travelend:
key_in=0
GET #4, key_in

IF key_in=73 THEN
   VR(16)=DPOS
   SPEED=100
   MOVEABS(0)
   VR(18)=1 ' Actual weld position
   GOTO step3
ENDIF

IF key_in=81 THEN MOVE(-10)
IF key_in=82 THEN MOVE(-1)
IF key_in=83 THEN MOVE(1)
IF key_in=84 THEN MOVE(10)

GOTO travelend
```

```
step3:
PRINT #4, " Number of welds>  "
PRINT #4, " per plate      >  "
PRINT #4, "       5        >  "
PRINT #4, " and press Enter>  "

nwelds=5

numberofwelds:
key_in=0
GET #4, key_in

IF key_in=73 THEN
   VR(17)=nwelds
   VR(9)=1 ' INI ends
   STOP
ENDIF

IF key_in=82 THEN nwelds=nwelds+1
IF key_in=83 THEN nwelds=nwelds-1

IF nwelds<1 THEN nwelds=1
IF nwelds>9 THEN nwelds=9

IF key_in=82 OR key_in=83 THEN
   PRINT #4, " Number of welds>  "
   PRINT #4, " per plate      >  "
   IF nwelds=1 THEN PRINT #4, "       1        >  "
   IF nwelds=2 THEN PRINT #4, "       2        >  "
   IF nwelds=3 THEN PRINT #4, "       3        >  "
   IF nwelds=4 THEN PRINT #4, "       4        >  "
   IF nwelds=5 THEN PRINT #4, "       5        >  "
   IF nwelds=6 THEN PRINT #4, "       6        >  "
   IF nwelds=7 THEN PRINT #4, "       7        >  "
   IF nwelds=8 THEN PRINT #4, "       8        >  "
   IF nwelds=9 THEN PRINT #4, "       9        >  "
   PRINT #4, " and press Enter>  "
ENDIF

GOTO numberofwelds

'==============================================================
'Name:      Display.bas
'Function:  Handles display output
'----------------------------------------------------------------------
'Date:      08/04
'Author:    Gil Lopes
```

```
'-------------------------------------------------------------------------
'Notes:    VR(1) = Display options
'          VR(2) = Speed value
'          VR(3) = Width value
'-------------------------------------------------------------------------
'          Copyright (c) 2004 Cranfield University
'          Welding Engineering Research Centre, Cranfield University
'          Cranfield, Bedford, MK43 0AL
'          Tel: +44-1234-750111 Fax: +44-1234-754109
'=========================================================================

IF VR(1)=0 THEN
   PRINT #4, "Speed(Hz) Width(mm)"
   PRINT #4, " ";VR(2), VR(3)
   IF VR(18)=1 THEN PRINT #4, "P1";" TS(m/min)=";VR(19)
   IF VR(18)=2 THEN PRINT #4, "P2";" TS(m/min)=";VR(19)
   IF VR(18)=3 THEN PRINT #4, "P3";" TS(m/min)=";VR(19)
   IF VR(18)=4 THEN PRINT #4, "P4";" TS(m/min)=";VR(19)
   IF VR(18)=5 THEN PRINT #4, "P5";" TS(m/min)=";VR(19)
   IF VR(18)=6 THEN PRINT #4, "P6";" TS(m/min)=";VR(19)
   IF VR(18)=7 THEN PRINT #4, "P7";" TS(m/min)=";VR(19)
   IF VR(18)=8 THEN PRINT #4, "P8";" TS(m/min)=";VR(19)
   IF VR(18)=9 THEN PRINT #4, "P9";" TS(m/min)=";VR(19)
   PRINT #4, "      IDLE      "
ENDIF

IF VR(1)=1 THEN
   PRINT #4, "Speed(Hz) Width(mm)"
   PRINT #4, " ";VR(2), VR(3)
   IF VR(18)=1 THEN PRINT #4, "P1";" TS(m/min)=";VR(19)
   IF VR(18)=2 THEN PRINT #4, "P2";" TS(m/min)=";VR(19)
   IF VR(18)=3 THEN PRINT #4, "P3";" TS(m/min)=";VR(19)
   IF VR(18)=4 THEN PRINT #4, "P4";" TS(m/min)=";VR(19)
   IF VR(18)=5 THEN PRINT #4, "P5";" TS(m/min)=";VR(19)
   IF VR(18)=6 THEN PRINT #4, "P6";" TS(m/min)=";VR(19)
   IF VR(18)=7 THEN PRINT #4, "P7";" TS(m/min)=";VR(19)
   IF VR(18)=8 THEN PRINT #4, "P8";" TS(m/min)=";VR(19)
   IF VR(18)=9 THEN PRINT #4, "P9";" TS(m/min)=";VR(19)
   PRINT #4, "     RUNNING     "
ENDIF

IF VR(1)=2 THEN
   PRINT #4, "Speed(Hz) Width(mm)"
   PRINT #4, VR(2), VR(3)
   PRINT #4, "                "
   PRINT #4, "      IDLE      "
ENDIF
```

```
VR(8)=0

STOP

'=================================================================
'Name:      Procs.bas
'Function:  Handles Process Events and Status
'-------------------------------------------------------------------------
'Date:      08/04
'Author:    Gil Lopes
'-------------------------------------------------------------------------
'          Copyright (c) 2004 Cranfield University
'          Welding Engineering Research Centre, Cranfield University
'          Cranfield, Bedford, MK43 0AL
'          Tel: +44-1234-750111 Fax: +44-1234-754109
'=================================================================

IF VR(5)=3 THEN
   BASE(0)
   SPEED=10
   MOVEABS((VR(16)/VR(17)) * (VR(18)-1))
   WAIT IDLE
   VR(5)=0
ENDIF

IF VR(5)=2 THEN
   BASE(3)
   MOVEABS(VR(3))
   'WAIT UNTIL OFFPOS=0
ENDIF


BASE(1)
SPEED=VR(2)

IF VR(5)=1 THEN
   IF MSPEED=0 THEN MOVE(10000)
   BASE(0)
   SPEED=(VR(19)*1000)/60
   MOVEABS((VR(16)/VR(17))*VR(18))
   VR(1)=1
ENDIF

IF VR(5)=0 THEN
   SPEED=0
   RAPIDSTOP
   RAPIDSTOP
   RAPIDSTOP
```

```
ENDIF

VR(7)=0

STOP

'=================================================================
'Name:     keyboard.bas
'Function:  Handles Keystroke Processes
'-----------------------------------------------------------------------
'Date:     08/04
'Author:    Gil Lopes
'-----------------------------------------------------------------------
'         Copyright (c) 2004 Cranfield University
'         Welding Engineering Research Centre, Cranfield University
'         Cranfield, Bedford, MK43 0AL
'         Tel: +44-1234-750111 Fax: +44-1234-754109
'=================================================================

IF VR(4)<>0 THEN
   IF VR(4)=80 THEN VR(2)=VR(2)+10 'Increment Speed by 10
   IF VR(4)=79 THEN VR(2)=VR(2)+1 'Increment Speed by 1
   IF VR(4)=78 THEN VR(2)=VR(2)-1 'Decrement Speed by 1
   IF VR(4)=77 THEN VR(2)=VR(2)-10 'Decrement Speed by 10
   IF VR(4)=81 THEN
      VR(3)=VR(3)+1 'Increment Width by 1
      VR(5)=2
   ENDIF
   IF VR(4)=82 THEN
      VR(3)=VR(3)+0.1 'Increment Width by 0.1
      VR(5)=2
   ENDIF
   IF VR(4)=83 THEN
      VR(3)=VR(3)-0.1 'Decrement Width by 0.1
      VR(5)=2
   ENDIF
   IF VR(4)=84 THEN
      VR(3)=VR(3)-1 'Decrement Width by 1
      VR(5)=2
   ENDIF

   IF (VR(4)=34) AND (VR(18)>1) THEN VR(18)=VR(18)-1
   IF (VR(4)=36) AND (VR(18)<VR(17)) THEN VR(18)=VR(18)+1
   IF VR(4)=35 THEN VR(5)=3
   IF (VR(4)=33) AND (VR(19)<VR(20)) THEN VR(19)=VR(19)+0.05
   IF (VR(4)=37) AND (VR(19)>VR(21)) THEN VR(19)=VR(19)-0.05

   IF VR(4)=46 THEN VR(5)=1 'Start the process
```

```
   IF VR(4)=49 THEN VR(5)=1 'Start the process
   IF VR(4)=44 THEN VR(5)=0 'Stop the process
   IF VR(4)=51 THEN VR(5)=0 'Stop the process
   VR(1)=VR(5)

   IF VR(2)>VR(15) THEN VR(2)=VR(15) 'Speed_Max
   IF VR(2)<VR(14) THEN VR(2)=VR(14) 'Speed_Min
   IF VR(3)>VR(11) THEN VR(3)=VR(11) 'Width_Max
   IF VR(3)<VR(10) THEN VR(3)=VR(10) 'Width_Min
ENDIF

VR(4)=0
VR(6)=0

STOP
```

Appendix G. WeldData analysis software

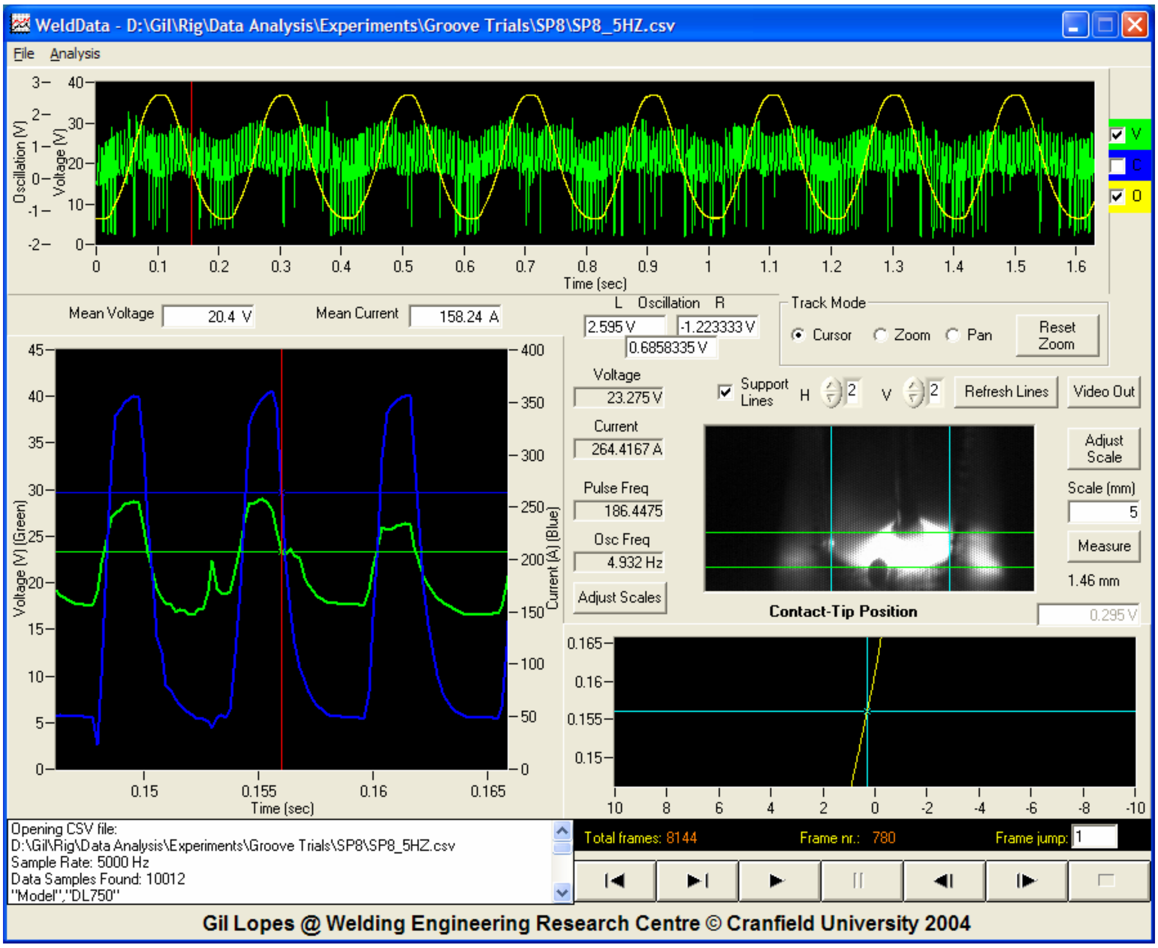


Figure G.1 – WeldData main screen and source code

"MainWindow.frm"

```
Dim VideoAdjustScale As Integer
Dim VideoMeasure As Integer
Dim VideoOutWindow As Integer
Dim VideoMeasureRatio As Single

Private Sub AdjustScaleBtn_Click()
Dim AdjustScalePicture, TestFile As String

If VideoAdjustScale = 1 Or VideoAdjustScale = 2 Then
   VideoAdjustScale = 0
   VideoScaleStatus.Caption = ""
Else
   TestFile = ""
   AdjustScalePicture = Left(DataFileName, Len(DataFileName) - 3) + "jpg"
   TestFile = Dir(AdjustScalePicture)
   If TestFile <> "" Then
      VideoOut.Picture = LoadPicture(AdjustScalePicture)
```

```
        VideoScaleStatus.Caption = "Point 1"
        VideoAdjustScale = 1
    Else
        MsgBox ("No Scaling Picture File ...")
    End If
End If

End Sub

Private Sub CCheck_Click()
With VIO
If CCheck Then
    .Axes("CAxis").Visible = True
    .Plots.Item("C").Visible = True
Else
    .Axes("CAxis").Visible = False
    .Plots.Item("C").Visible = False
End If
End With
End Sub



Private Sub Command1_Click()
'Call OptionsUpdate
VIO.TrackMode = cwGTrackDragAnnotation
Call Fill_Graphs
End Sub

Private Sub Command2_Click()
Adjust_Scales.Show
End Sub

Private Sub MeasureBtn_Click()
If VideoMeasure = 1 Or VideoMeasure = 2 Then
    VideoMeasure = 0
    VideoScaleStatus.Caption = ""
Else
    VideoMeasure = 1
    VideoScaleStatus.Caption = "Point 1"
End If
End Sub

Private Sub MenuAvgCur_Click()
    Avg_Cur.Show
End Sub

Private Sub MenuCrossPlots_Click()
```

```
Analysis.Show
End Sub

Private Sub VidBaseSuppLines_H_ValueChanged(Value As Variant,
PreviousValue As Variant, ByVal OutOfRange As Boolean)
If VidBaseSuppLines_H.Value < 0 Then
   VidBaseSuppLines_H.Value = 0
   Exit Sub
End If
If VidBaseSuppLines_H.Value > 9 Then
   VidBaseSuppLines_H.Value = 9
   Exit Sub
End If
If Value < PreviousValue Then
   If Value >= 0 Then Unload VidBaseSuppLine_H(VidBaseSuppLines_H_Idx)
   VidBaseSuppLines_H_Idx = VidBaseSuppLines_H_Idx - 1
End If
If Value > PreviousValue Then
   VidBaseSuppLines_H_Idx = VidBaseSuppLines_H_Idx + 1
   If Value > 0 Then
      Load VidBaseSuppLine_H(VidBaseSuppLines_H_Idx)
      VidBaseSuppLines_H_Pos(VidBaseSuppLines_H_Idx) =
VidBaseSuppLine_H(VidBaseSuppLines_H_Idx).Y1
      VidBaseSuppLine_H(VidBaseSuppLines_H_Idx).Visible = True
   End If
End If
End Sub

Private Sub VidBaseSuppLines_V_ValueChanged(Value As Variant,
PreviousValue As Variant, ByVal OutOfRange As Boolean)
If VidBaseSuppLines_V.Value < 0 Then
   VidBaseSuppLines_V.Value = 0
   Exit Sub
End If
If VidBaseSuppLines_V.Value > 9 Then
   VidBaseSuppLines_V.Value = 9
   Exit Sub
End If
If Value < PreviousValue Then
   If Value >= 0 Then Unload VidBaseSuppLine_V(VidBaseSuppLines_V_Idx)
   VidBaseSuppLines_V_Idx = VidBaseSuppLines_V_Idx - 1
End If
If Value > PreviousValue Then
   VidBaseSuppLines_V_Idx = VidBaseSuppLines_V_Idx + 1
   If Value > 0 Then
      Load VidBaseSuppLine_V(VidBaseSuppLines_V_Idx)
      VidBaseSuppLines_V_Pos(VidBaseSuppLines_V_Idx) =
VidBaseSuppLine_V(VidBaseSuppLines_V_Idx).X1
```

```
        VidBaseSuppLine_V(VidBaseSuppLines_V_Idx).Visible = True
    End If
End If
End Sub

Private Sub VideoOut_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
VidBaseSuppLines_H_Selected = 0
VidBaseSuppLines_H_Selected = 0
If VideoAdjustScale = 0 And VideoMeasure = 0 Then
    For i = 1 To VidBaseSuppLines_H_Idx
        If VidBaseSuppLines_H_Pos(i) = Y Then
            VidBaseSuppLines_H_Selected = i
            Exit Sub
        End If
    Next i
    For i = 1 To VidBaseSuppLines_V_Idx
        If VidBaseSuppLines_V_Pos(i) = X Then
            VidBaseSuppLines_V_Selected = i
            Exit Sub
        End If
    Next i
End If
End Sub

Private Sub VideoOut_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
If VidBaseSuppLines_H_Selected > 0 Then
    VidBaseSuppLine_H(VidBaseSuppLines_H_Selected).Y1 = Y
    VidBaseSuppLine_H(VidBaseSuppLines_H_Selected).Y2 = Y
End If
If VidBaseSuppLines_V_Selected > 0 Then
    VidBaseSuppLine_V(VidBaseSuppLines_V_Selected).X1 = X
    VidBaseSuppLine_V(VidBaseSuppLines_V_Selected).X2 = X
End If
End Sub

Private Sub VideoOut_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
If VideoAdjustScale = 1 Then
    VideoAdjustScalePoints(1, 1) = X
    VideoAdjustScalePoints(1, 2) = Y
    VideoScaleStatus.Caption = "Point 2"
    VideoAdjustScale = 2
Else
    If VideoAdjustScale = 2 Then
        VideoAdjustScalePoints(2, 1) = X
        VideoAdjustScalePoints(2, 2) = Y
```

```
        VideoAdjustScaleRatio = Sqr(Abs(VideoAdjustScalePoints(2, 1) -
VideoAdjustScalePoints(1, 1)) ^ 2 + Abs(VideoAdjustScalePoints(2, 2) -
VideoAdjustScalePoints(1, 2)) ^ 2)
        VideoScaleStatus.Caption = ""
        VideoAdjustScale = 0
    End If
End If
If VideoMeasure = 1 Then
    VideoMeasurePoints(1, 1) = X
    VideoMeasurePoints(1, 2) = Y
    VideoScaleStatus.Caption = "Point 2"
    VideoMeasure = 2
Else
    If VideoMeasure = 2 Then
        VideoMeasurePoints(2, 1) = X
        VideoMeasurePoints(2, 2) = Y
        VideoMeasureRatio = Sqr(Abs(VideoMeasurePoints(2, 1) -
VideoMeasurePoints(1, 1)) ^ 2 + Abs(VideoMeasurePoints(2, 2) -
VideoMeasurePoints(1, 2)) ^ 2)
        VideoMeasureRatio = VideoMeasureRatio * (Int(VideoScaleValue.Text) /
VideoAdjustScaleRatio)
        VideoScaleStatus.Caption = Round(VideoMeasureRatio, 2) & " mm"
        VideoMeasure = 0
    End If
End If
If VidBaseSuppLines_H_Selected > 0 Then
    VidBaseSuppLines_H_Pos(VidBaseSuppLines_H_Selected) = Y
    VidBaseSuppLine_H(VidBaseSuppLines_H_Selected).Y1 = Y
    VidBaseSuppLine_H(VidBaseSuppLines_H_Selected).Y2 = Y
    VidBaseSuppLines_H_Selected = 0
End If
If VidBaseSuppLines_V_Selected > 0 Then
    VidBaseSuppLines_V_Pos(VidBaseSuppLines_V_Selected) = X
    VidBaseSuppLine_V(VidBaseSuppLines_V_Selected).X1 = X
    VidBaseSuppLine_V(VidBaseSuppLines_V_Selected).X2 = X
    VidBaseSuppLines_V_Selected = 0
End If

End Sub

Private Sub VideoOutBtn_Click()
If VideoOutWindow = 1 Then
    With MMControl1
        .Notify = False
        .Wait = True
        .Shareable = False
        .hWndDisplay = VideoOut.hWnd
        .EjectVisible = False
```

```
    End With
    VideoOutWindow = 0
Else
    With MMControl1
        .Notify = False
        .Wait = True
        .Shareable = False
        .hWndDisplay = 0
        .EjectVisible = False
    End With
    VideoOutWindow = 1
End If

End Sub

Private Sub Form_Load()
Dim TestFile As String

DataPath = GetSetting(AppName:=AppName, section:="Startup",
Key:="DataPath")
DataFileName = GetSetting(AppName:=AppName, section:="Startup",
Key:="DataFileName")
VideoFileName = GetSetting(AppName:=AppName, section:="Startup",
Key:="VideoFileName")

' Place some settings in the registry.
'SaveSetting "MyApp", "Startup", "Top", 75
'Debug.Print GetSetting(AppName:="MyApp", section:="Startup", Key:="Left",
Default:="25")
'DeleteSetting "MyApp", "Startup"
VideoOutWindow = 0
VideoAdjustScale = 0
VideoMeasure = 0

With MMControl1
    .Notify = False
    .Wait = True
    .Shareable = False
    .hWndDisplay = VideoOut.hWnd
    .EjectVisible = False
End With

VCheck.Value = 1
CCheck.Value = 0
OCheck.Value = 0

TestFile = ""
If VideoFileName <> "" Then TestFile = Dir(VideoFileName)
```

```
If TestFile = "" Then VideoFileName = App.Path & "\NoVideoLong.avi"
Call Load_Video

TestFile = ""
If DataFileName <> "" Then TestFile = Dir(DataFileName)
If TestFile <> "" Then Call Load_Data

VI.ChartLength = Samples_P_Graph
OSC.ChartLength = Samples_P_Graph

'Option1.Value = True 'Voltage Selected

End Sub

Private Sub Form_Terminate()
End
End Sub

Private Sub Form_Unload(Cancel As Integer)

MMControl1.Command = "Close"

SaveSetting AppName, "Startup", "DataPath", DataPath
SaveSetting AppName, "Startup", "DataFileName", DataFileName
SaveSetting AppName, "Startup", "VideoFileName", VideoFileName

End Sub

Private Sub FrameJump_Change()
MMControl1.Frames = Int(FrameJump.Text)
End Sub

Private Sub MenuAvgPk_Click()
   Avg_vs_Pk.Show
End Sub

Private Sub MenuFileClose_Click(Index As Integer)
MMControl1.Command = "Close"
DataFileName = ""
VideoFileName = ""
MainWindow.Caption = AppName
List1.Clear
End Sub

Private Sub MenuFileExit_Click(Index As Integer)
Unload Me
End
End Sub
```

```
Private Sub MenuFileOpen_Click(Index As Integer)

MMControl1.Command = "Close"

With CommonDialog1
   .DefaultExt = ".csv"
   .Filter = "Oscilloscope converted data (*.csv)|*.csv"
   .ShowOpen
   DataFileName = .FileName
   DataPath = CurDir
End With

If DataFileName <> "" Then
   VideoFileName = Left(DataFileName, Len(DataFileName) - 3) + "avi"
   If VideoFileName <> "" Then Call Load_Video
   Call Load_Data
End If

End Sub

Private Sub MenuPkBk_Click()
   Pk_vs_Bk.Show
End Sub

Private Sub MMControl1_Done(NotifyCode As Integer)
'MsgBox "Done " & NotifyCode
End Sub

Private Sub MMControl1_StatusUpdate()
FrameNr.Caption = MMControl1.Position
DataPtr = MMControl1.Position
If DataPtr <> PrevDataPtr Then Call Update_Graphs
PrevDataPtr = DataPtr
End Sub

Private Sub Load_Video()
Dim StrLine As String
Dim sReturn As String * 512
Dim lPos As Long
Dim lStart As Long
Dim lWidth, lHeight As Integer

StrLine = "Open " & Chr(34) & VideoFileName & Chr(34) & " type avivideo Alias
video1"
i = mciSendString(StrLine, 0&, 0, 0)
i = mciSendString("Where video1 destination", ByVal sReturn, Len(sReturn) - 1,
0)
```

```
i = mciSendString("VideoSize video1 200%", ByVal sReturn, Len(sReturn) - 1,
0)

i = mciSendString("Play video1", 0&, 0, 0)


lStart = InStr(1, sReturn, " ")
lPos = InStr(lStart + 1, sReturn, " ")
lStart = InStr(lPos + 1, sReturn, " ")
If lStart <> 0 Then
   lWidth = Int(Mid(sReturn, lPos, lStart - lPos))
   lHeight = Int(Mid(sReturn, lStart + 1))
Else
   lWidth = 256
   lHeight = 128
End If
i = mciSendString("Close video1", 0&, 0, 0)


VideoOut.Left = Picture1.Left + ((Picture1.Width - lWidth) / 2)
VideoOut.Top = Picture1.Top + ((Picture1.Height - lHeight) / 2)
VideoOut.Width = lWidth
VideoOut.Height = lHeight


With MMControl1
   .FileName = VideoFileName
   .Command = "Open"
   VideoFrames = .Length
End With

DataPtr = MMControl1.Position
PrevDataPtr = DataPtr

End Sub

Private Sub Load_Data()
MainWindow.Caption = AppName & " - " & DataFileName

Call Check_DataFile

If DataFrames > VideoFrames Then
   TotalFrames.Caption = VideoFrames
   MaxFrames = VideoFrames
Else
   TotalFrames.Caption = DataFrames
   MaxFrames = DataFrames
End If
```

```
Call Show_Header
Call Fill_Graphs

End Sub

Private Sub Check_DataFile()
Dim flag As Boolean
Dim strInput, strOut As String
Dim buf_len, cycles, reminder As Integer
Dim lfChar As Integer

buf_len = 1000
strOut = ""

'Verify file compliance with CR+LF - if not change it
Open DataFileName For Input As #1
strInput = Input(buf_len, #1)
lfChar = InStr(strInput, Chr(10))
If lfChar <> 0 Then
    If Mid(strInput, lfChar - 1, 1) <> Chr(13) Then
        cycles = Int(LOF(1) / buf_len)
        reminder = LOF(1) Mod buf_len
        Close #1
        Open DataFileName For Input As #1
        Open DataFileName & ".tmp" For Output As #2
        For i = 1 To cycles
            strInput = strOut + Input(buf_len, #1)
            lfChar = 1
            While lfChar > 0
                lfChar = InStr(strInput, Chr(10))
                If lfChar <> 0 Then
                    strOut = Mid(strInput, 1, lfChar - 1)
                    Print #2, strOut
                    strInput = Mid(strInput, lfChar + 1)
                End If
            Wend
            strOut = strInput
        Next i
        strInput = strOut + Input(reminder, #1)
        lfChar = 1
        While lfChar > 0
            lfChar = InStr(strInput, Chr(10))
            If lfChar <> 0 Then
                strOut = Mid(strInput, 1, lfChar - 1)
                Print #2, strOut
                strInput = Mid(strInput, lfChar + 1)
            End If
```

```
        strOut = strInput
      Wend
      Print #2, strOut
      Close #1, #2
      Kill DataFileName
      Name DataFileName & ".tmp" As DataFileName
   End If
End If
Close #1
'Verify nr of frames
DataFrames = 0
DataSampleRate = 0
Open DataFileName For Input As #1
   While Not EOF(1)
      Line Input #1, strInput
      If DataSampleRate = 0 Then
         i = InStr(strInput, "HResolution")
         If i <> 0 Then
            i = InStr(strInput, ",")
            DataSampleRate = CDbl(Mid(strInput, i + 1))
         End If
      End If
      DataFrames = DataFrames + 1
   Wend
Close #1
DataFrames = DataFrames - Header

End Sub

Private Sub Show_Header()
Dim ReadLine As String

With List1
   .Clear
   .AddItem "Opening CSV file: "
   .AddItem DataFileName
   .AddItem "Sample Rate: " & Int(1 / DataSampleRate) & " Hz"
   .AddItem "Data Samples Found: " & DataFrames

   Open DataFileName For Input As #1
   For i = 1 To 11
      Line Input #1, ReadLine
      .AddItem ReadLine
   Next i
   Close #1
End With

End Sub
```

```
Private Sub Fill_Graphs()
Dim ReadLine As String
Dim V1, V2, V3 As String

DataPtr = 0
For i = 1 To MaxDataSampled
   V(i) = 0
   C(i) = 0
   O(i) = 0
Next i
For i = 1 To MaxGraphData
   Osc_Pks(i) = 0
   Osc_PksNeg(i) = 0
   Osc_PksCentre(i) = 0
Next i
Osc_PksIdx = 0
Osc_PksIdx = 0
Osc_PksNegIdx = 0
Osc_PksCentreIdx = 0
Osc_Frequency = 0
Pulse_Frequency = 0

Open DataFileName For Input As #1

For i = 1 To 11
   Line Input #1, ReadLine
Next i

i = 0
While Not EOF(1)
   Line Input #1, ReadLine
   If i < MaxDataSampled And ReadLine <> "" Then
      j = InStr(ReadLine, ",")
      V1 = Mid$(ReadLine, 1, j - 1)
      ReadLine = Mid$(ReadLine, j + 1)
      j = InStr(ReadLine, ",")
      V2 = Mid$(ReadLine, 1, j - 1)
      V3 = Mid$(ReadLine, j + 1)
      j = InStr(V3, ",")
      If j <> 0 Then V3 = Mid(V3, 1, j - 1)
      V(i) = CDbl(V1)
      C(i) = CDbl(V3)
      O(i) = CDbl(V2)
      i = i + 1
   End If
Wend
Close #1
```

```
'Call OptionsUpdate
Call Update_Scales
Call Osc_Freq_Detect
Call Pulse_Freq_Detect
Call AVG_Calculation
Call Update_Graphs

End Sub

Private Sub Pulse_Freq_Detect()
Dim PulseAvg As Single
Dim PulsePk As Single
Dim PulsePkIdx, PulsePrevIdx, PulseCycles As Integer
Dim PulseLower As Boolean

PulsePk = 0
PulsePkIdx = 0
PulsePrevIdx = 0
PulseCycles = 0
PulseLower = True

For i = 0 To DataFrames
   PulseAvg = PulseAvg + C(i)
Next i
PulseAvg = PulseAvg / DataFrames

For i = 0 To DataFrames
   If C(i) > PulseAvg Then
      PulseLower = False
      If PulsePk < C(i) Then
         PulsePk = C(i)
         PulsePkIdx = i
      End If
   Else
      If PulseLower = False Then
         If PulsePrevIdx <> 0 Then
            If PulseCycles = 0 Then
               'Pulse_Frequency = 1 / ((PulsePkIdx - PulsePrevIdx) *
DataSampleRate)
               Pulse_Frequency = PulsePkIdx - PulsePrevIdx
            Else
               'Pulse_Frequency = (Pulse_Frequency + (1 / ((PulsePkIdx -
PulsePrevIdx) * DataSampleRate))) / 2
               Pulse_Frequency = Pulse_Frequency + (PulsePkIdx -
PulsePrevIdx)
            End If
         End If
```

```
            PulsePrevIdx = PulsePkIdx
            PulsePk = 0
            PulseCycles = PulseCycles + 1
            PulseLower = True
        End If
    End If
Next i
Pulse_Frequency = 1 / ((Pulse_Frequency / PulseCycles) * DataSampleRate)

End Sub

Private Sub Osc_Freq_Detect()
Dim OscAvg As Single
Dim OscPk, OscPkNeg, OscCentre As Single
Dim OscPkIdx, OscPkNegIdx, OscCentreIdx, OscPrevIdx As Integer
Dim OscCycles As Integer
Dim OscLower As Boolean
Dim OscMid As Boolean

OscAvg = 0
OscMin = 1000000
OscMax = -1000000
OscPk = -1000000
OscPkNeg = 1000000
OscCentre = 1000000
OscPkIdx = 0
OscPkNegIdx = 0
OscPrevIdx = 0
Osc_PksIdx = 0
Osc_PksNegIdx = 0
Osc_PksCentreIdx = 0
OscCycles = -1
Osc_Frequency = 0
OscLower = True
OscMid = False

For i = 0 To DataFrames
    OscAvg = OscAvg + O(i)
Next i
OscAvg = OscAvg / DataFrames

If O(0) > OscAvg Then
    OscMid = True
Else
    OscMid = False
End If

For i = 0 To DataFrames
```

```
    If O(i) > OscAvg And OscMid = False Then
        Osc_PksCentre(Osc_PksCentreIdx) = i
        Osc_PksCentreIdx = Osc_PksCentreIdx + 1
        OscMid = True
    Else
        If O(i) < OscAvg And OscMid = True Then
        Osc_PksCentre(Osc_PksCentreIdx) = i
        Osc_PksCentreIdx = Osc_PksCentreIdx + 1
        OscMid = False
        End If
    End If

    If O(i) > OscAvg Then
        If OscLower = True And OscPkNegIdx <> 0 Then
            Osc_PksNeg(Osc_PksNegIdx) = OscPkNegIdx
            Osc_PksNegIdx = Osc_PksNegIdx + 1
            OscPkNeg = 1000000
        End If
        OscLower = False
        If OscPk < O(i) Then
            OscPk = O(i)
            OscPkIdx = i
        End If
    Else
        If OscLower = False Then
            Osc_Pks(Osc_PksIdx) = OscPkIdx
            Osc_PksIdx = Osc_PksIdx + 1
            If OscPrevIdx <> 0 Then
                If OscCycles > 0 Then
                    Osc_Frequency = (Osc_Frequency + 1 / ((OscPkIdx - OscPrevIdx)
* DataSampleRate)) / 2
                Else
                    Osc_Frequency = 1 / ((OscPkIdx - OscPrevIdx) * DataSampleRate)
                End If
            End If
            OscPrevIdx = OscPkIdx
            OscPk = -1000000
            OscCycles = OscCyles + 1
            OscLower = True
        End If
        If OscPkNeg > O(i) Then
            OscPkNeg = O(i)
            OscPkNegIdx = i
        End If
    End If
    If O(i) > OscMax Then OscMax = O(i)
    If O(i) < OscMin Then OscMin = O(i)
Next i
```

```
End Sub

Public Sub Update_Graphs()
Dim HalfMaxGraphData As Integer
Dim T1, T2 As Single

HalfMaxGraphData = Int(Samples_P_Graph / 2) - 1
Nominal_V.Caption = V(DataPtr) & " V"
Nominal_I.Caption = C(DataPtr) & " A"
Oscillation_Position = O(DataPtr) & " V"
OscFrequency.Caption = Round(Osc_Frequency, 3) & " Hz"
PulseFrequency.Caption = Pulse_Frequency
Oscillation_Min.Text = OscMin & " V"
Oscillation_Max.Text = OscMax & " V"
Oscillation_Mid.Text = ((Abs(OscMin) + Abs(OscMax)) / 2) + OscMin & " V"

With VIO
   .Cursors(1).XPosition = DataSampleRate * DataPtr
End With

k = DataSampleRate * (DataPtr - HalfMaxGraphData)
For i = 0 To Samples_P_Graph - 1
   j = (DataPtr - HalfMaxGraphData) + i
   If j < 0 Or j > MaxFrames Then
      VIGraphData(i, 0) = 0
      VIGraphData(i, 1) = 0
      OSCGraphData(i, 0) = 0
   Else
      VIGraphData(i, 0) = V(j)
      VIGraphData(i, 1) = C(j)
      OSCGraphData(i, 0) = O(j)
   End If
   OSCGraphData(i, 1) = k
   k = k + DataSampleRate
Next i

T1 = OSCGraphData(i - 1, 0)
T2 = OSCGraphData(i - 1, 1)
For i = Samples_P_Graph To MaxGraphData
   OSCGraphData(i, 0) = T1
   OSCGraphData(i, 1) = T2
Next i

'VIO.TrackMode = cwGTrackZoomRectXY
'VIO.TrackMode = cwGTrackPanPlotAreaY

With VIO
```

```
    .Cursors(1).XPosition = DataSampleRate * DataPtr
    .Axes(1).Minimum = 0
    .Axes(1).Maximum = DataSampleRate * (MaxFrames - 1)
    .Axes(2).Caption = "Voltage (V)"
    .Axes(3).Caption = "Current (A)"
    .Axes(4).Caption = "Oscillation (V)"
    .Plots.Item(1).PlotY V, 0, DataSampleRate
    .Plots.Item(2).PlotY C, 0, DataSampleRate
    .Plots.Item(3).PlotY O, 0, DataSampleRate
    If VCheck Then
        .Axes("VAxis").Visible = True
        .Plots.Item("V").Visible = True
    End If
    If CCheck Then
        .Axes("CAxis").Visible = True
        .Plots.Item("C").Visible = True
    End If
    If OCheck Then
        .Axes("OAxis").Visible = True
        .Plots.Item("O").Visible = True
    End If

End With

With VI
    .ClearData
    .Axes(1).Minimum = ((DataPtr - HalfMaxGraphData) * DataSampleRate)
    .Axes(1).Maximum = ((DataPtr + HalfMaxGraphData) * DataSampleRate)
    .Cursors(1).PointIndex = HalfMaxGraphData
    .Cursors(2).PointIndex = HalfMaxGraphData
    .Cursors(3).XPosition = (DataSampleRate * DataPtr)
    .PlotY VIGraphData, ((DataPtr - HalfMaxGraphData) * DataSampleRate),
DataSampleRate, bPlotPerRow = True
End With

With OSC
    .ClearData
    .Axes(2).Minimum = ((DataPtr - HalfMaxGraphData) * DataSampleRate)
    .Axes(2).Maximum = ((DataPtr + HalfMaxGraphData) * DataSampleRate)
    .Cursors(1).PointIndex = HalfMaxGraphData
    .PlotXY OSCGraphData, bPlotPerRow = True
End With

End Sub

Private Sub OCheck_Click()
With VIO
If OCheck Then
```

```
   .Axes("OAxis").Visible = True
   .Plots.Item("O").Visible = True
Else
   .Axes("OAxis").Visible = False
   .Plots.Item("O").Visible = False
End If
End With
End Sub

Private Sub Option4_Click()
If Option4.Value = True Then VIO.TrackMode = cwGTrackDragCursor
End Sub

Private Sub Option5_Click()
If Option5.Value = True Then VIO.TrackMode = cwGTrackZoomRectXY
End Sub

Private Sub Option6_Click()
If Option6.Value = True Then VIO.TrackMode = cwGTrackPanPlotAreaXY
End Sub

Private Sub VCheck_Click()
With VIO
If VCheck Then
   .Axes("VAxis").Visible = True
   .Plots.Item("V").Visible = True
Else
   .Axes("VAxis").Visible = False
   .Plots.Item("V").Visible = False
End If
End With
End Sub

Private Sub VidSuppLinesChk_Click()
If VidSuppLinesChk Then
   VidBaseSuppLines_H.Enabled = True
   VidBaseSuppLines_V.Enabled = True
   For i = 1 To VidBaseSuppLines_H_Idx
      VidBaseSuppLine_H(i).Visible = True
   Next i
   For i = 1 To VidBaseSuppLines_V_Idx
      VidBaseSuppLine_V(i).Visible = True
   Next i
Else
   VidBaseSuppLines_H.Enabled = False
   VidBaseSuppLines_V.Enabled = False
   For i = 1 To VidBaseSuppLines_H_Idx
      VidBaseSuppLine_H(i).Visible = False
```

```
    Next i
    For i = 1 To VidBaseSuppLines_V_Idx
        VidBaseSuppLine_V(i).Visible = False
    Next i
End If
End Sub

Private Sub VidSuppRefreshLines_Click()
    For i = 1 To VidBaseSuppLines_H_Idx
        VidBaseSuppLine_H(i).Visible = False
        VidBaseSuppLine_H(i).Visible = True
    Next i
    For i = 1 To VidBaseSuppLines_V_Idx
        VidBaseSuppLine_V(i).Visible = False
        VidBaseSuppLine_V(i).Visible = True
    Next i
End Sub

Private Sub VIO_CursorChange(CursorIndex As Long, XPos As Variant, YPos
As Variant, bTracking As Boolean)
VIOCursorX = XPos
End Sub

Private Sub VIO_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If (VIOCursorX / DataSampleRate) - DataPtr >= 0 Then
    MMControl1.Frames = Int((VIOCursorX / DataSampleRate) - DataPtr)
    MMControl1.Command = "Step"
Else
    MMControl1.Frames = Int(DataPtr - (VIOCursorX / DataSampleRate))
    MMControl1.Command = "Back"
End If
MMControl1.Frames = FrameJump.Text
DataPtr = VIOCursorX
End Sub

Public Sub Update_Scales()
VI.Plots(1).YAxis.Minimum = Scale_V_Min
VI.Plots(1).YAxis.Maximum = Scale_V_Max
VI.Plots(1).YAxis.AutoScale = Scale_V_Auto
VI.Plots(2).YAxis.Minimum = Scale_C_Min
VI.Plots(2).YAxis.Maximum = Scale_C_Max
VI.Plots(2).YAxis.AutoScale = Scale_C_Auto
OSC.Plots(1).XAxis.Minimum = Scale_O_Min
OSC.Plots(1).XAxis.Maximum = Scale_O_Max
OSC.Plots(1).XAxis.AutoScale = Scale_O_Auto
Call Update_Graphs
End Sub
```

```
Sub AVG_Calculation()
Dim V_Total, C_Total As Double

V_Total = 0#
C_Total = 0#

For i = 0 To DataFrames
   V_Total = V_Total + CDbl(V(i))
   C_Total = C_Total + CDbl(C(i))
Next i

AVG_Voltage = Str(Round(V_Total / DataFrames, 2)) & "  V"
AVG_Current = Str(Round(C_Total / DataFrames, 2)) & "  A"

End Sub
```

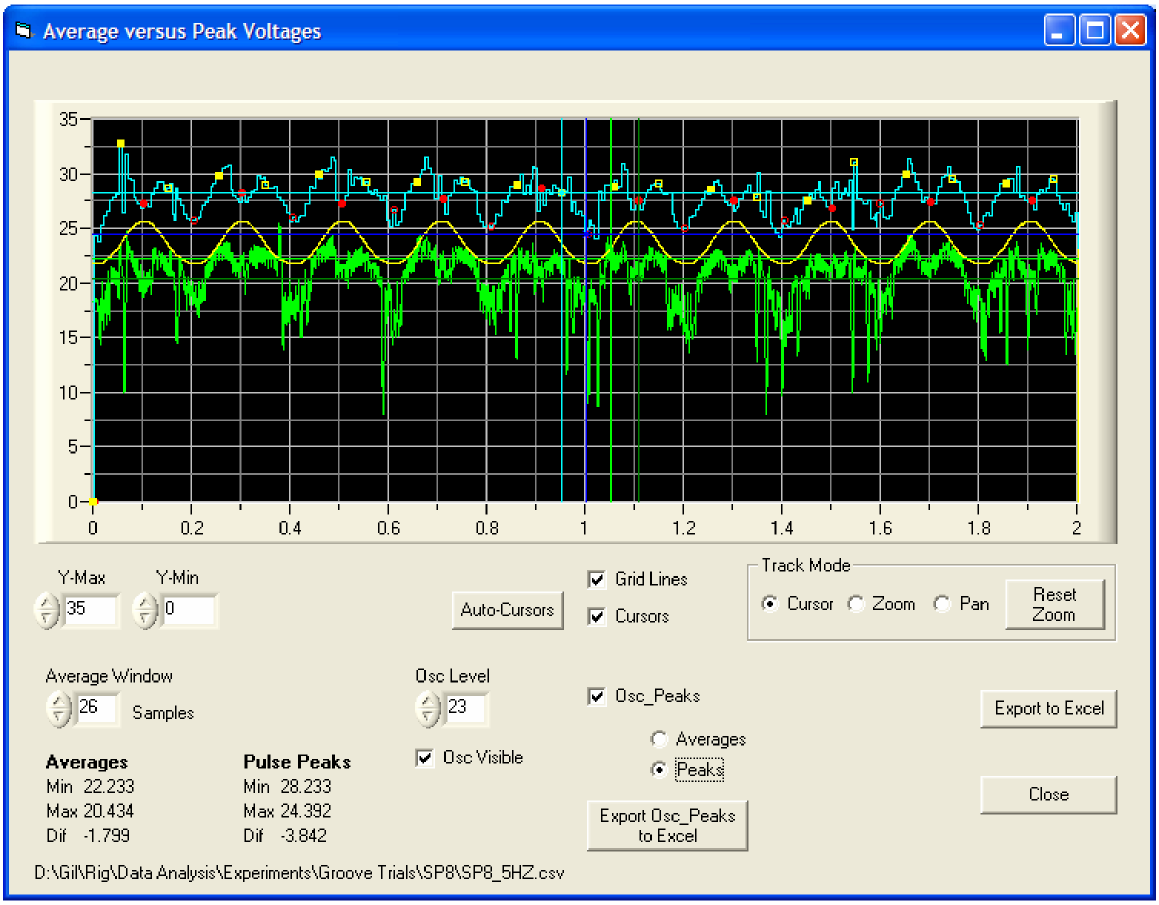


Figure G.2 – Average versus Peal Voltages screen and source code

"Avg_vs_Pk.frm"

```
Dim Avg(MaxDataSampled) As Single
Dim Pk(MaxDataSampled) As Single
Dim OscVal(MaxDataSampled) As Single

Private Sub AvgPk_Click()
If Option5.Value = True Then Call Cursors_Update
If Option4.Value = True Then
    Call Cursors_MinMax
    Call Scales_Update
End If
End Sub

Private Sub AvgWindow_ValueChanged(Value As Variant, PreviousValue As
Variant, ByVal OutOfRange As Boolean)
    Call Avg_Update
End Sub

Private Sub Command1_Click()
    Me.Hide
End Sub
```

```
Private Sub Command2_Click()
AvgPk.ClearData
AvgPk.TrackMode = cwGTrackDragAnnotation
Call Form_Activate
End Sub



Private Sub CursorsSt_Click()
If CursorsSt.Value = 1 Then
   AvgPk.Cursors(1).Visible = True
   AvgPk.Cursors(2).Visible = True
   AvgPk.Cursors(3).Visible = True
   AvgPk.Cursors(4).Visible = True
Else
   AvgPk.Cursors(1).Visible = False
   AvgPk.Cursors(2).Visible = False
   AvgPk.Cursors(3).Visible = False
   AvgPk.Cursors(4).Visible = False
End If
End Sub

Private Sub Factor_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
   Call Pk_Update
End Sub

Private Sub Export_Excel_Click()
Dim Excel_Time As Single

Screen.MousePointer = vbHourglass

Set ExlObj = CreateObject("excel.application")      ' Initialize the excel object
ExlObj.Workbooks.Add                                 ' Add an excel workbook

With ExlObj.ActiveSheet
   .cells(1, 1).Value = "Time"
   .cells(1, 2).Value = "Average"
   .cells(1, 3).Value = "Peaks"
   .cells(1, 4).Value = "Oscillation"
   Excel_Time = AvgPk.Axes(1).Minimum
   For i = 1 To (AvgPk.Axes(1).Maximum - AvgPk.Axes(1).Minimum) /
DataSampleRate
      .cells(i + 1, 1).Value = Excel_Time
      .cells(i + 1, 2).Value = Avg((AvgPk.Axes(1).Minimum / DataSampleRate) +
i)
      .cells(i + 1, 3).Value = Pk((AvgPk.Axes(1).Minimum / DataSampleRate) + i)
```

```
        .cells(i + 1, 4).Value = OscVal((AvgPk.Axes(1).Minimum /
DataSampleRate) + i)
        Excel_Time = Excel_Time + DataSampleRate
    Next i
End With
Screen.MousePointer = vbDefault
ExlObj.Visible = True


End Sub

Private Sub Export_OP_Excel_Click()
Dim Osc_PkArray(MaxGraphData, 1) As Single
Dim Osc_PkNegArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreLArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreRArray(MaxGraphData, 1) As Single
Dim j_idx, k_idx As Integer


Screen.MousePointer = vbHourglass

Set ExlObj = CreateObject("excel.application")      ' Initialize the excel object
ExlObj.Workbooks.Add                                 ' Add an excel workbook

With ExlObj.ActiveSheet
    .cells(1, 1).Value = "Time"
    .cells(1, 2).Value = "Osc_Pk_Left"
    .cells(1, 3).Value = "Time"
    .cells(1, 4).Value = "Osc_Pk_Right"
    .cells(1, 5).Value = "Time"
    .cells(1, 6).Value = "Osc_Mid_Left"
    .cells(1, 7).Value = "Time"
    .cells(1, 8).Value = "Osc_Mid_Right"

    For i = 0 To Osc_PksIdx
        .cells(i + 2, 1) = Osc_Pks(i) * DataSampleRate
        .cells(i + 2, 3) = Osc_PksNeg(i) * DataSampleRate

        If Osc_Pk_Op1 = True Then
            .cells(i + 2, 2) = Avg(Osc_Pks(i))
            .cells(i + 2, 4) = Avg(Osc_PksNeg(i))
        Else
            .cells(i + 2, 2) = Pk(Osc_Pks(i))
            .cells(i + 2, 4) = Pk(Osc_PksNeg(i))
        End If
    Next i
    j_idx = 2
    k_idx = 2
```

```
   For i = 0 To Osc_PksCentreIdx
      If Osc_Pk_Op1 = True Then
         If O(Osc_PksCentre(i)) > O(Osc_PksCentre(i) + 2) Then
            .cells(j_idx, 6) = Avg(Osc_PksCentre(i))
            .cells(j_idx, 5) = Osc_PksCentre(i) * DataSampleRate
            j_idx = j_idx + 1
         Else
            .cells(k_idx, 8) = Avg(Osc_PksCentre(i))
            .cells(k_idx, 7) = Osc_PksCentre(i) * DataSampleRate
            k_idx = k_idx + 1
         End If
      Else
         If O(Osc_PksCentre(i)) > O(Osc_PksCentre(i) + 2) Then
            .cells(j_idx, 6) = Pk(Osc_PksCentre(i))
            .cells(j_idx, 5) = Osc_PksCentre(i) * DataSampleRate
            j_idx = j_idx + 1
         Else
            .cells(k_idx, 8) = Pk(Osc_PksCentre(i))
            .cells(k_idx, 7) = Osc_PksCentre(i) * DataSampleRate
            k_idx = k_idx + 1
         End If
      End If
   Next i
End With
Screen.MousePointer = vbDefault
ExlObj.Visible = True
End Sub

Private Sub Form_Activate()
   Label_DataFileName.Caption = DataFileName
   AvgWindow.Value = Int(1 / (DataSampleRate * Pulse_Frequency))
   'Call Avg_Update
   Call Pk_Update
   Call Osc_Update
   Call Cursors_Update
   Call Cursors_MinMax
   Call Scales_Update
   Call Osc_Peaks_Update
End Sub

Private Sub Avg_Update()
Dim AvgSum, AvgIni As Single

AvgIni = 0
For i = 1 To AvgWindow.Value
   AvgSum = AvgSum + V(i)
Next i
AvgIni = AvgSum / AvgWindow.Value
```

```
For i = 0 To DataFrames
   If i <= AvgWindow.Value Then
      Avg(i) = AvgIni
   Else
      AvgSum = AvgSum + V(i)
      AvgSum = AvgSum - V(i - AvgWindow.Value)
      Avg(i) = AvgSum / AvgWindow.Value
   End If
Next i

With AvgPk
   .Plots(1).XAxis.Minimum = 0
   .Plots(1).XAxis.Maximum = DataFrames * DataSampleRate
   .Plots(1).PlotY Avg, 0, DataSampleRate
   .Cursors(1).XPosition = 0
   .Cursors(2).XPosition = 0
End With


End Sub

Private Sub Pk_Update()
Dim AvgCur, PkV As Single
Dim PkIdx As Integer
Dim PkDetect As Boolean
Dim i, j As Integer

AvgCur = 0
PkIdx = 0
PkV = 0
PkDetect = False

For i = 0 To DataFrames
   AvgCur = AvgCur + C(i)
Next i

AvgCur = AvgCur / DataFrames

For i = 0 To DataFrames
   If C(i) > AvgCur * 1.8 Then
      PkDetect = True
      If PkV < V(i) Then
         PkV = V(i)
         PkIdx = i
      End If
      If i > 0 Then Pk(i) = Pk(i - 1)
   Else
```

```
        If PkDetect = True Then
            PkDetect = False
            For j = PkIdx To i
                Pk(j) = PkV
            Next j
        End If
        If i > 0 Then Pk(i) = Pk(i - 1)
        PkV = 0
    End If

Next i

With AvgPk
    .Plots(2).PlotY Pk, 0, DataSampleRate
    .Cursors(3).XPosition = 0
    .Cursors(4).XPosition = 0
End With

End Sub

Private Sub Osc_Update()

For i = 0 To DataFrames
    OscVal(i) = O(i) + OscLevel.Value
Next i

AvgPk.Plots(3).PlotY OscVal, 0, DataSampleRate

End Sub

Private Sub Osc_Peaks_Update()
Dim Osc_PkArray(MaxGraphData, 1) As Single
Dim Osc_PkNegArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreLArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreRArray(MaxGraphData, 1) As Single
Dim j_idx, k_idx As Integer

If OscPksChk = 1 Then
    For i = 0 To Osc_PksIdx
        Osc_PkArray(i, 0) = Osc_Pks(i) * DataSampleRate
        Osc_PkNegArray(i, 0) = Osc_PksNeg(i) * DataSampleRate

        If Osc_Pk_Op1 = True Then
            Osc_PkArray(i, 1) = Avg(Osc_Pks(i))
            Osc_PkNegArray(i, 1) = Avg(Osc_PksNeg(i))
        Else
            Osc_PkArray(i, 1) = Pk(Osc_Pks(i))
            Osc_PkNegArray(i, 1) = Pk(Osc_PksNeg(i))
```

```
      End If
    Next i
    j_idx = 0
    k_idx = 0
    For i = 0 To Osc_PksCentreIdx
      If Osc_Pk_Op1 = True Then
        If O(Osc_PksCentre(i)) > O(Osc_PksCentre(i) + 2) Then
          Osc_PkCentreLArray(j_idx, 1) = Avg(Osc_PksCentre(i))
          Osc_PkCentreLArray(j_idx, 0) = Osc_PksCentre(i) * DataSampleRate
          j_idx = j_idx + 1
        Else
          Osc_PkCentreRArray(k_idx, 1) = Avg(Osc_PksCentre(i))
          Osc_PkCentreRArray(k_idx, 0) = Osc_PksCentre(i) *
DataSampleRate
          k_idx = k_idx + 1
        End If
      Else
        If O(Osc_PksCentre(i)) > O(Osc_PksCentre(i) + 2) Then
          Osc_PkCentreLArray(j_idx, 1) = Pk(Osc_PksCentre(i))
          Osc_PkCentreLArray(j_idx, 0) = Osc_PksCentre(i) * DataSampleRate
          j_idx = j_idx + 1
        Else
          Osc_PkCentreRArray(k_idx, 1) = Pk(Osc_PksCentre(i))
          Osc_PkCentreRArray(k_idx, 0) = Osc_PksCentre(i) *
DataSampleRate
          k_idx = k_idx + 1
        End If
      End If
    Next i
End If

AvgPk.Plots(4).PlotXY Osc_PkArray, bPlotPerRow = True
AvgPk.Plots(5).PlotXY Osc_PkNegArray, bPlotPerRow = True
AvgPk.Plots(6).PlotXY Osc_PkCentreLArray, bPlotPerRow = True
AvgPk.Plots(7).PlotXY Osc_PkCentreRArray, bPlotPerRow = True

End Sub

Private Sub GridLinesSt_Click()
If GridLinesSt.Value = 1 Then
    AvgPk.Plots(1).XAxis.Ticks.MajorGrid = True
    AvgPk.Plots(1).XAxis.Ticks.MinorGrid = True
    AvgPk.Plots(1).YAxis.Ticks.MajorGrid = True
    AvgPk.Plots(1).YAxis.Ticks.MinorGrid = True
Else
    AvgPk.Plots(1).XAxis.Ticks.MajorGrid = False
    AvgPk.Plots(1).XAxis.Ticks.MinorGrid = False
    AvgPk.Plots(1).YAxis.Ticks.MajorGrid = False
```

```
   AvgPk.Plots(1).YAxis.Ticks.MinorGrid = False
End If
End Sub

Private Sub Option4_Click()
If Option4.Value = True Then AvgPk.TrackMode = cwGTrackDragCursor
End Sub

Private Sub Option5_Click()
If Option5.Value = True Then AvgPk.TrackMode = cwGTrackZoomRectXY
End Sub

Private Sub Option6_Click()
If Option6.Value = True Then AvgPk.TrackMode = cwGTrackPanPlotAreaXY
End Sub

Private Sub Osc_Pk_Op1_Click()
Call Osc_Peaks_Update
End Sub

Private Sub Osc_Pk_Op2_Click()
Call Osc_Peaks_Update
End Sub

Private Sub OscLevel_ValueChanged(Value As Variant, PreviousValue As
Variant, ByVal OutOfRange As Boolean)
Call Osc_Update
End Sub

Private Sub Cursors_Update()
Dim Cursor_Pos As Single

   Cursor_Pos = (AvgPk.Axes(1).Maximum - AvgPk.Axes(1).Minimum)
   If AvgPk.Cursors(1).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(1).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(1).XPosition = (Cursor_Pos / 1.8) +
AvgPk.Axes(1).Minimum
   End If
   If AvgPk.Cursors(2).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(2).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(2).XPosition = (Cursor_Pos / 1.9) +
AvgPk.Axes(1).Minimum
   End If
   If AvgPk.Cursors(3).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(3).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(3).XPosition = (Cursor_Pos / 2) + AvgPk.Axes(1).Minimum
   End If
```

```
   If AvgPk.Cursors(4).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(4).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(4).XPosition = (Cursor_Pos / 2.1) +
AvgPk.Axes(1).Minimum
   End If
End Sub

Private Sub OscPksChk_Click()
Call Osc_Peaks_Update
End Sub

Private Sub OscSt_Click()
If OscSt.Value = 1 Then
   AvgPk.Plots(3).Visible = True
Else
   AvgPk.Plots(3).Visible = False
End If
End Sub

Private Sub Cursors_MinMax()
AvgMax.Caption = Round(AvgPk.Cursors(1).YPosition, 3)
AvgMin.Caption = Round(AvgPk.Cursors(2).YPosition, 3)
AvgDif.Caption = Round((AvgPk.Cursors(1).YPosition -
AvgPk.Cursors(2).YPosition), 3)
PkMax.Caption = Round(AvgPk.Cursors(3).YPosition, 3)
PkMin.Caption = Round(AvgPk.Cursors(4).YPosition, 3)
PkDif.Caption = Round((AvgPk.Cursors(3).YPosition -
AvgPk.Cursors(4).YPosition), 3)
End Sub

Private Sub Scales_Update()
AvgPk.Axes(2).AutoScale = True
AvgPk.Axes(2).AutoScaleNow
YMax.Value = AvgPk.Axes(2).Maximum
YMin.Value = AvgPk.Axes(2).Minimum
End Sub

Private Sub YMax_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
AvgPk.Axes(2).Maximum = Value
End Sub

Private Sub YMin_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
AvgPk.Axes(2).Minimum = Value
End Sub
```

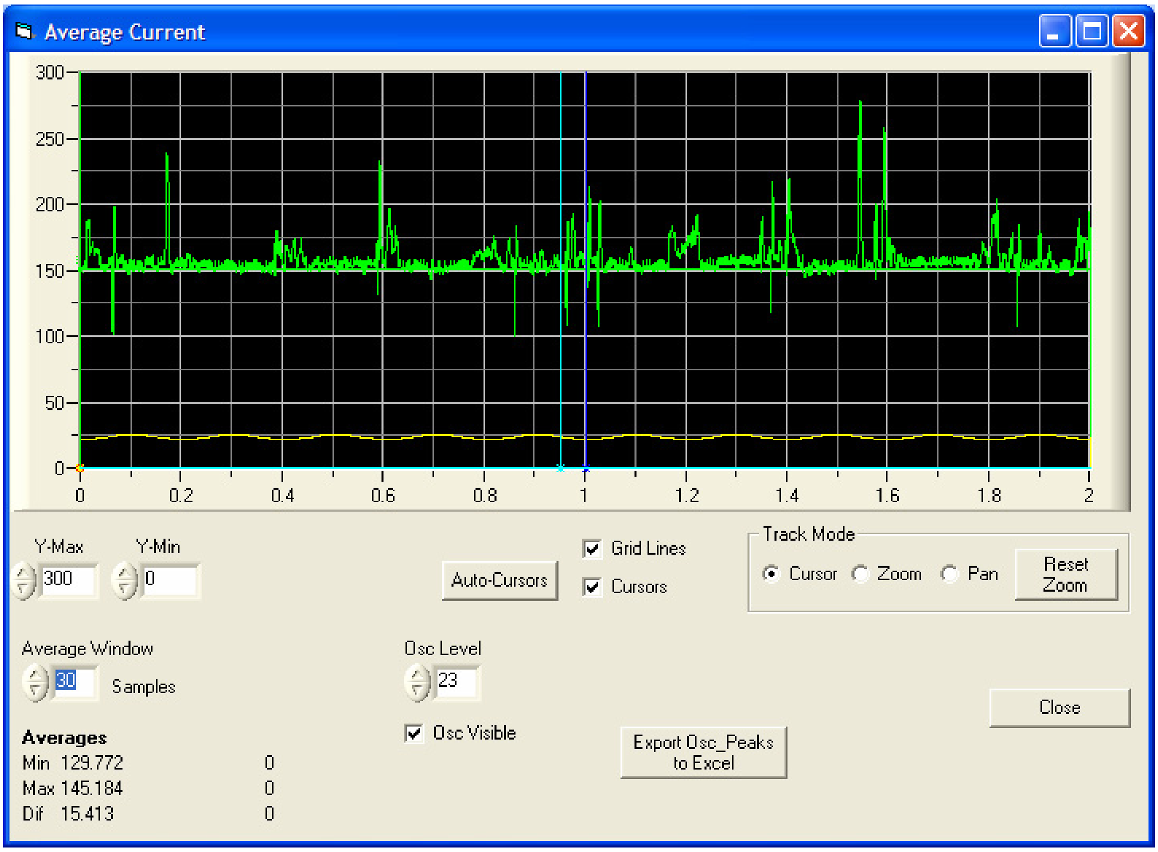


Figure G.3 – Average Current screen and source code

"Avg_Cur.frm"

```
Dim Avg(MaxDataSampled) As Single
Dim Pk(MaxDataSampled) As Single
Dim OscVal(MaxDataSampled) As Single

Private Sub AvgPk_Click()
If Option5.Value = True Then Call Cursors_Update
If Option4.Value = True Then
    Call Cursors_MinMax
    Call Scales_Update
End If
End Sub

Private Sub AvgWindow_ValueChanged(Value As Variant, PreviousValue As
Variant, ByVal OutOfRange As Boolean)
    Call Avg_Update
End Sub

Private Sub Command1_Click()
    Me.Hide
End Sub
```

```
Private Sub Command2_Click()
AvgPk.ClearData
AvgPk.TrackMode = cwGTrackDragAnnotation
Call Form_Activate
End Sub

Private Sub CursorsSt_Click()
If CursorsSt.Value = 1 Then
   AvgPk.Cursors(1).Visible = True
   AvgPk.Cursors(2).Visible = True
   AvgPk.Cursors(3).Visible = True
   AvgPk.Cursors(4).Visible = True
Else
   AvgPk.Cursors(1).Visible = False
   AvgPk.Cursors(2).Visible = False
   AvgPk.Cursors(3).Visible = False
   AvgPk.Cursors(4).Visible = False
End If
End Sub

Private Sub Export_OP_Excel_Click()

Dim Osc_PkArray(MaxGraphData, 1) As Single
Dim Osc_PkNegArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreLArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreRArray(MaxGraphData, 1) As Single
Dim j_idx, k_idx As Integer


Screen.MousePointer = vbHourglass

Set ExlObj = CreateObject("excel.application")      ' Initialize the excel object
ExlObj.Workbooks.Add                                ' Add an excel workbook

With ExlObj.ActiveSheet
   .cells(1, 1).Value = "Time"
   .cells(1, 2).Value = "Osc_Pk_Left"
   .cells(1, 3).Value = "Time"
   .cells(1, 4).Value = "Osc_Pk_Right"
   .cells(1, 5).Value = "Time"
   .cells(1, 6).Value = "Osc_Mid_Left"
   .cells(1, 7).Value = "Time"
   .cells(1, 8).Value = "Osc_Mid_Right"

   For i = 0 To Osc_PksIdx
      .cells(i + 2, 1) = Osc_Pks(i) * DataSampleRate
      .cells(i + 2, 3) = Osc_PksNeg(i) * DataSampleRate
```

```
        .cells(i + 2, 2) = Avg(Osc_Pks(i))
        .cells(i + 2, 4) = Avg(Osc_PksNeg(i))

    Next i
    j_idx = 2
    k_idx = 2
    For i = 0 To Osc_PksCentreIdx
        If O(Osc_PksCentre(i)) > O(Osc_PksCentre(i) + 2) Then
            .cells(j_idx, 6) = Avg(Osc_PksCentre(i))
            .cells(j_idx, 5) = Osc_PksCentre(i) * DataSampleRate
            j_idx = j_idx + 1
        Else
            .cells(k_idx, 8) = Avg(Osc_PksCentre(i))
            .cells(k_idx, 7) = Osc_PksCentre(i) * DataSampleRate
            k_idx = k_idx + 1
        End If
    Next i
End With
Screen.MousePointer = vbDefault
ExlObj.Visible = True

End Sub

Private Sub Form_Activate()
    AvgWindow.Value = Int(1 / (DataSampleRate * Pulse_Frequency))
    Call Osc_Update
    Call Cursors_Update
    Call Cursors_MinMax
    Call Scales_Update
    Call Osc_Peaks_Update
End Sub

Private Sub Avg_Update()
Dim AvgSum, AvgIni As Single

AvgIni = 0
For i = 1 To AvgWindow.Value
    AvgSum = AvgSum + C(i)
Next i
AvgIni = AvgSum / AvgWindow.Value

For i = 0 To DataFrames
    If i <= AvgWindow.Value Then
        Avg(i) = AvgIni
    Else
        AvgSum = AvgSum + C(i)
        AvgSum = AvgSum - C(i - AvgWindow.Value)
        Avg(i) = AvgSum / AvgWindow.Value
```

```
    End If
Next i

With AvgPk
    .Plots(1).XAxis.Minimum = 0
    .Plots(1).XAxis.Maximum = DataFrames * DataSampleRate
    .Plots(1).PlotY Avg, 0, DataSampleRate
    .Cursors(1).XPosition = 0
    .Cursors(2).XPosition = 0
End With


End Sub

Private Sub Osc_Update()

For i = 0 To DataFrames
    OscVal(i) = O(i) + OscLevel.Value
Next i

AvgPk.Plots(3).PlotY OscVal, 0, DataSampleRate

End Sub

Private Sub Osc_Peaks_Update()
Dim Osc_PkArray(MaxGraphData, 1) As Single
Dim Osc_PkNegArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreArray(MaxGraphData, 1) As Single

If OscPksChk = 1 Then
    For i = 0 To Osc_PksIdx
        Osc_PkArray(i, 0) = Osc_Pks(i) * DataSampleRate
        Osc_PkNegArray(i, 0) = Osc_PksNeg(i) * DataSampleRate

        If Osc_Pk_Op1 = True Then
            Osc_PkArray(i, 1) = Avg(Osc_Pks(i))
            Osc_PkNegArray(i, 1) = Avg(Osc_PksNeg(i))
        Else
            Osc_PkArray(i, 1) = Pk(Osc_Pks(i))
            Osc_PkNegArray(i, 1) = Pk(Osc_PksNeg(i))
        End If
    Next i
    For i = 0 To Osc_PksCentreIdx
        Osc_PkCentreArray(i, 0) = Osc_PksCentre(i) * DataSampleRate
        If Osc_Pk_Op1 = True Then
            Osc_PkCentreArray(i, 1) = Avg(Osc_PksCentre(i))
        Else
            Osc_PkCentreArray(i, 1) = Pk(Osc_PksCentre(i))
```

```
        End If
    Next i
End If
AvgPk.Plots(4).PlotXY Osc_PkArray, bPlotPerRow = True
AvgPk.Plots(5).PlotXY Osc_PkNegArray, bPlotPerRow = True
AvgPk.Plots(6).PlotXY Osc_PkCentreArray, bPlotPerRow = True

End Sub

Private Sub GridLinesSt_Click()
If GridLinesSt.Value = 1 Then
    AvgPk.Plots(1).XAxis.Ticks.MajorGrid = True
    AvgPk.Plots(1).XAxis.Ticks.MinorGrid = True
    AvgPk.Plots(1).YAxis.Ticks.MajorGrid = True
    AvgPk.Plots(1).YAxis.Ticks.MinorGrid = True
Else
    AvgPk.Plots(1).XAxis.Ticks.MajorGrid = False
    AvgPk.Plots(1).XAxis.Ticks.MinorGrid = False
    AvgPk.Plots(1).YAxis.Ticks.MajorGrid = False
    AvgPk.Plots(1).YAxis.Ticks.MinorGrid = False
End If
End Sub

Private Sub Option4_Click()
If Option4.Value = True Then AvgPk.TrackMode = cwGTrackDragCursor
End Sub

Private Sub Option5_Click()
If Option5.Value = True Then AvgPk.TrackMode = cwGTrackZoomRectXY
End Sub

Private Sub Option6_Click()
If Option6.Value = True Then AvgPk.TrackMode = cwGTrackPanPlotAreaXY
End Sub

Private Sub Osc_Pk_Op1_Click()
Call Osc_Peaks_Update
End Sub

Private Sub Osc_Pk_Op2_Click()
Call Osc_Peaks_Update
End Sub

Private Sub OscLevel_ValueChanged(Value As Variant, PreviousValue As
Variant, ByVal OutOfRange As Boolean)
Call Osc_Update
End Sub
```

```
Private Sub Cursors_Update()
Dim Cursor_Pos As Single

   Cursor_Pos = (AvgPk.Axes(1).Maximum - AvgPk.Axes(1).Minimum)
   If AvgPk.Cursors(1).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(1).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(1).XPosition = (Cursor_Pos / 1.8) +
AvgPk.Axes(1).Minimum
   End If
   If AvgPk.Cursors(2).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(2).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(2).XPosition = (Cursor_Pos / 1.9) +
AvgPk.Axes(1).Minimum
   End If
   If AvgPk.Cursors(3).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(3).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(3).XPosition = (Cursor_Pos / 2) + AvgPk.Axes(1).Minimum
   End If
   If AvgPk.Cursors(4).XPosition <= AvgPk.Axes(1).Minimum Or
AvgPk.Cursors(4).XPosition >= AvgPk.Axes(1).Maximum Then
      AvgPk.Cursors(4).XPosition = (Cursor_Pos / 2.1) +
AvgPk.Axes(1).Minimum
   End If
End Sub

Private Sub OscPksChk_Click()
Call Osc_Peaks_Update
End Sub

Private Sub OscSt_Click()
If OscSt.Value = 1 Then
   AvgPk.Plots(3).Visible = True
Else
   AvgPk.Plots(3).Visible = False
End If
End Sub

Private Sub Cursors_MinMax()
AvgMax.Caption = Round(AvgPk.Cursors(1).YPosition, 3)
AvgMin.Caption = Round(AvgPk.Cursors(2).YPosition, 3)
AvgDif.Caption = Round((AvgPk.Cursors(1).YPosition -
AvgPk.Cursors(2).YPosition), 3)
PkMax.Caption = Round(AvgPk.Cursors(3).YPosition, 3)
PkMin.Caption = Round(AvgPk.Cursors(4).YPosition, 3)
PkDif.Caption = Round((AvgPk.Cursors(3).YPosition -
AvgPk.Cursors(4).YPosition), 3)
End Sub
```

```
Private Sub Scales_Update()
AvgPk.Axes(2).AutoScale = True
AvgPk.Axes(2).AutoScaleNow
YMax.Value = AvgPk.Axes(2).Maximum
YMin.Value = AvgPk.Axes(2).Minimum
End Sub

Private Sub YMax_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
AvgPk.Axes(2).Maximum = Value
End Sub

Private Sub YMin_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
AvgPk.Axes(2).Minimum = Value
End Sub
```

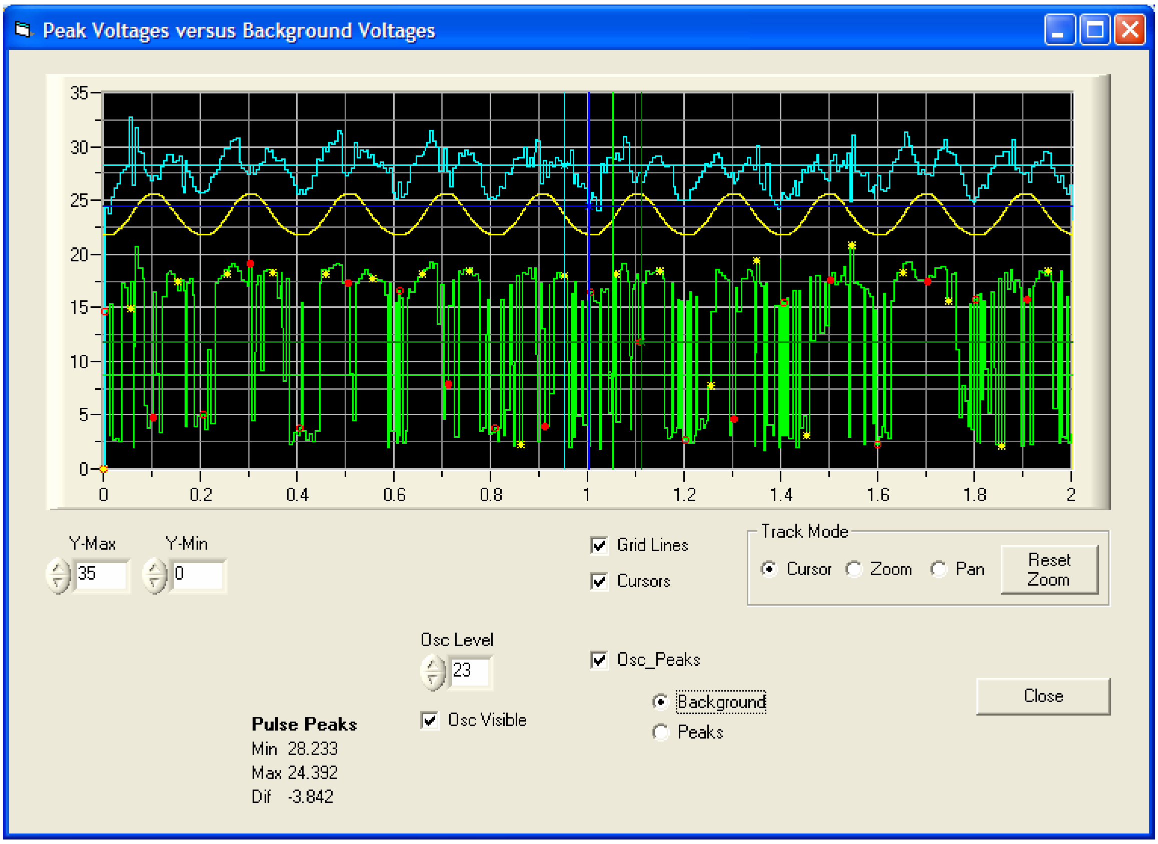


Figure G.4 – Peak Voltages versus Background Voltages screen and source code

"Pk_vs_Bk.frm"

```
Dim Bk(MaxDataSampled) As Single
Dim Pk(MaxDataSampled) As Single
Dim OscVal(MaxDataSampled) As Single

Private Sub Command1_Click()
    Me.Hide
End Sub

Private Sub Command2_Click()
PkBk.ClearData
PkBk.TrackMode = cwGTrackDragAnnotation
Call Form_Activate
End Sub

Private Sub CursorsSt_Click()
If CursorsSt.Value = 1 Then
    PkBk.Cursors(1).Visible = True
    PkBk.Cursors(2).Visible = True
    PkBk.Cursors(3).Visible = True
    PkBk.Cursors(4).Visible = True
Else
```

```
    PkBk.Cursors(1).Visible = False
    PkBk.Cursors(2).Visible = False
    PkBk.Cursors(3).Visible = False
    PkBk.Cursors(4).Visible = False
End If
End Sub

Private Sub Factor_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
    Call Pk_Update
End Sub

Private Sub Form_Activate()
    PkBk.Plots(1).XAxis.Minimum = 0
    PkBk.Plots(1).XAxis.Maximum = DataFrames * DataSampleRate

    Call Bk_Update
    Call Pk_Update
    Call Osc_Update
    Call Cursors_Update
    Call Cursors_MinMax
    Call Scales_Update
    Call Osc_Peaks_Update
End Sub

Private Sub Pk_Update()
Dim AvgCur, PkV As Single
Dim PkIdx As Integer
Dim PkDetect As Boolean
Dim i, j As Integer

AvgCur = 0
PkIdx = 0
PkV = 0
PkDetect = False

For i = 0 To DataFrames
    AvgCur = AvgCur + C(i)
Next i

AvgCur = AvgCur / DataFrames

For i = 0 To DataFrames
    If C(i) > AvgCur * 1.8 Then
        PkDetect = True
        If PkV < V(i) Then
            PkV = V(i)
            PkIdx = i
```

```
      End If
      If i > 0 Then Pk(i) = Pk(i - 1)
   Else
      If PkDetect = True Then
         PkDetect = False
         For j = PkIdx To i
            Pk(j) = PkV
         Next j
      End If
      If i > 0 Then Pk(i) = Pk(i - 1)
      PkV = 0
   End If

Next i

With PkBk
   .Plots(2).PlotY Pk, 0, DataSampleRate
   .Cursors(3).XPosition = 0
   .Cursors(4).XPosition = 0
End With

End Sub

Private Sub Bk_Update()
Dim AvgCur, BkV As Single
Dim BkIdx As Integer
Dim BkDetect As Boolean
Dim i, j As Integer

AvgCur = 0
BkIdx = 0
BkV = 50
BkDetect = False

For i = 0 To DataFrames
   AvgCur = AvgCur + C(i)
Next i

AvgCur = AvgCur / DataFrames

For i = 0 To DataFrames
   If C(i) < AvgCur Then
      BkDetect = True
      If BkV > V(i) Then
         BkV = V(i)
         BkIdx = i
      End If
      If i > 0 Then Bk(i) = Bk(i - 1)
```

```
    Else
        If BkDetect = True Then
            BkDetect = False
            For j = BkIdx To i
                Bk(j) = BkV
            Next j
        End If
        If i > 0 Then Bk(i) = Bk(i - 1)
        BkV = 50
    End If

Next i

With PkBk
    .Plots(1).PlotY Bk, 0, DataSampleRate
    .Cursors(1).XPosition = 0
    .Cursors(2).XPosition = 0
End With

End Sub
Private Sub Osc_Update()

For i = 0 To DataFrames
    OscVal(i) = O(i) + OscLevel.Value
Next i

PkBk.Plots(3).PlotY OscVal, 0, DataSampleRate

End Sub

Private Sub Osc_Peaks_Update()
Dim Osc_PkArray(MaxGraphData, 1) As Single
Dim Osc_PkNegArray(MaxGraphData, 1) As Single
Dim Osc_PkCentreArray(MaxGraphData, 1) As Single

If OscPksChk = 1 Then
    For i = 0 To Osc_PksIdx
        Osc_PkArray(i, 0) = Osc_Pks(i) * DataSampleRate
        Osc_PkNegArray(i, 0) = Osc_PksNeg(i) * DataSampleRate

        If Osc_Pk_Op1 = True Then
            Osc_PkArray(i, 1) = Bk(Osc_Pks(i))
            Osc_PkNegArray(i, 1) = Bk(Osc_PksNeg(i))
        Else
            Osc_PkArray(i, 1) = Pk(Osc_Pks(i))
            Osc_PkNegArray(i, 1) = Pk(Osc_PksNeg(i))
        End If
    Next i
```

```
    For i = 0 To Osc_PksCentreIdx
        Osc_PkCentreArray(i, 0) = Osc_PksCentre(i) * DataSampleRate
        If Osc_Pk_Op1 = True Then
            Osc_PkCentreArray(i, 1) = Bk(Osc_PksCentre(i))
        Else
            Osc_PkCentreArray(i, 1) = Pk(Osc_PksCentre(i))
        End If
    Next i
End If
PkBk.Plots(4).PlotXY Osc_PkArray, bPlotPerRow = True
PkBk.Plots(5).PlotXY Osc_PkNegArray, bPlotPerRow = True
PkBk.Plots(6).PlotXY Osc_PkCentreArray, bPlotPerRow = True

End Sub

Private Sub GridLinesSt_Click()
If GridLinesSt.Value = 1 Then
    PkBk.Plots(1).XAxis.Ticks.MajorGrid = True
    PkBk.Plots(1).XAxis.Ticks.MinorGrid = True
    PkBk.Plots(1).YAxis.Ticks.MajorGrid = True
    PkBk.Plots(1).YAxis.Ticks.MinorGrid = True
Else
    PkBk.Plots(1).XAxis.Ticks.MajorGrid = False
    PkBk.Plots(1).XAxis.Ticks.MinorGrid = False
    PkBk.Plots(1).YAxis.Ticks.MajorGrid = False
    PkBk.Plots(1).YAxis.Ticks.MinorGrid = False
End If
End Sub

Private Sub Option4_Click()
If Option4.Value = True Then PkBk.TrackMode = cwGTrackDragCursor
End Sub

Private Sub Option5_Click()
If Option5.Value = True Then PkBk.TrackMode = cwGTrackZoomRectXY
End Sub

Private Sub Option6_Click()
If Option6.Value = True Then PkBk.TrackMode = cwGTrackPanPlotAreaXY
End Sub

Private Sub Osc_Pk_Op1_Click()
Call Osc_Peaks_Update
End Sub

Private Sub Osc_Pk_Op2_Click()
Call Osc_Peaks_Update
End Sub
```

```
Private Sub OscLevel_ValueChanged(Value As Variant, PreviousValue As
Variant, ByVal OutOfRange As Boolean)
Call Osc_Update
End Sub

Private Sub Cursors_Update()
Dim Cursor_Pos As Single

   Cursor_Pos = (PkBk.Axes(1).Maximum - PkBk.Axes(1).Minimum)
   If PkBk.Cursors(1).XPosition <= PkBk.Axes(1).Minimum Or
PkBk.Cursors(1).XPosition >= PkBk.Axes(1).Maximum Then
      PkBk.Cursors(1).XPosition = (Cursor_Pos / 1.8) + PkBk.Axes(1).Minimum
   End If
   If PkBk.Cursors(2).XPosition <= PkBk.Axes(1).Minimum Or
PkBk.Cursors(2).XPosition >= PkBk.Axes(1).Maximum Then
      PkBk.Cursors(2).XPosition = (Cursor_Pos / 1.9) + PkBk.Axes(1).Minimum
   End If
   If PkBk.Cursors(3).XPosition <= PkBk.Axes(1).Minimum Or
PkBk.Cursors(3).XPosition >= PkBk.Axes(1).Maximum Then
      PkBk.Cursors(3).XPosition = (Cursor_Pos / 2) + PkBk.Axes(1).Minimum
   End If
   If PkBk.Cursors(4).XPosition <= PkBk.Axes(1).Minimum Or
PkBk.Cursors(4).XPosition >= PkBk.Axes(1).Maximum Then
      PkBk.Cursors(4).XPosition = (Cursor_Pos / 2.1) + PkBk.Axes(1).Minimum
   End If
End Sub

Private Sub OscPksChk_Click()
Call Osc_Peaks_Update
End Sub

Private Sub OscSt_Click()
If OscSt.Value = 1 Then
   PkBk.Plots(3).Visible = True
Else
   PkBk.Plots(3).Visible = False
End If
End Sub

Private Sub Cursors_MinMax()
PkMax.Caption = Round(PkBk.Cursors(3).YPosition, 3)
PkMin.Caption = Round(PkBk.Cursors(4).YPosition, 3)
PkDif.Caption = Round((PkBk.Cursors(3).YPosition -
PkBk.Cursors(4).YPosition), 3)
End Sub

Private Sub Scales_Update()
```

```
PkBk.Axes(2).AutoScale = True
PkBk.Axes(2).AutoScaleNow
YMax.Value = PkBk.Axes(2).Maximum
YMin.Value = PkBk.Axes(2).Minimum
End Sub

Private Sub YMax_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
PkBk.Axes(2).Maximum = Value
End Sub

Private Sub YMin_ValueChanged(Value As Variant, PreviousValue As Variant,
ByVal OutOfRange As Boolean)
PkBk.Axes(2).Minimum = Value
End Sub
```

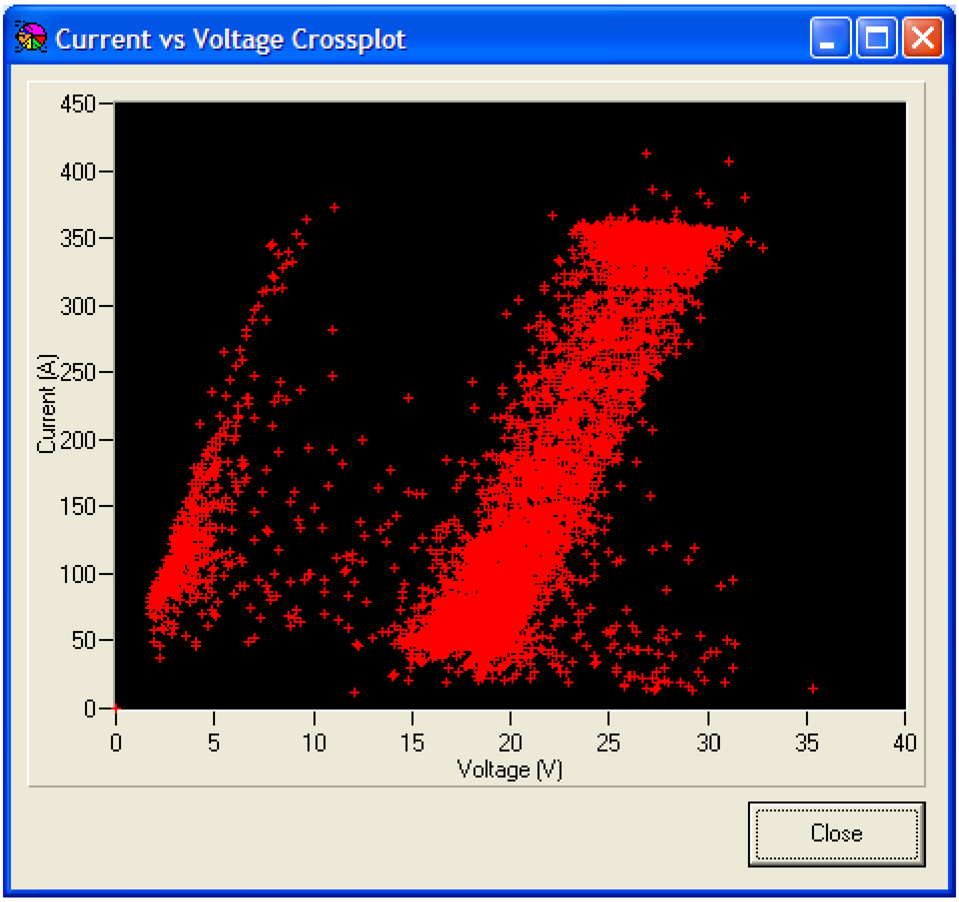


Figure G.5 – Current versus Voltage Crossplot

"Analysis.frm"

```
Private Sub CloseAnalysis_Click()
Unload Me
End Sub

Private Sub Form_Load()

    For i = 0 To MaxFrames
        CrossPlot2D(i, 0) = V(i)
        CrossPlot2D(i, 1) = C(i)
    Next i

    For i = 1 To Samples_P_Graph
        Plot1D_1(i) = V(DataPtr + i)
        Plot1D_2(i) = C(DataPtr + i)
    Next i

    CrossGraph2D.PlotXY CrossPlot2D, bPlotPerRow = True

End Sub
```

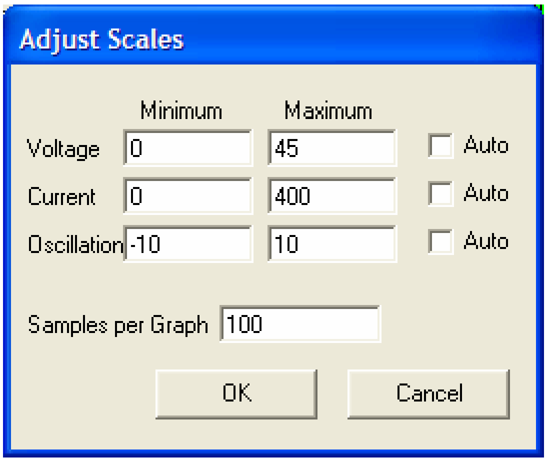


Figure G.6 – Adjust Scales screen and source code

"Adjust_Scales.frm"

```
Private Sub Command1_Click()
Scale_V_Min = Voltage_Min
Scale_V_Max = Voltage_Max
Scale_V_Auto = Voltage_Auto.Value
Scale_C_Min = Current_Min
Scale_C_Max = Current_Max
Scale_C_Auto = Current_Auto.Value
Scale_O_Min = Oscillation_Min
Scale_O_Max = Oscillation_Max
Scale_O_Auto = Oscillation_Auto.Value
Samples_P_Graph = SamplesPerGraph

Call MainWindow.Update_Scales
Adjust_Scales.Hide

End Sub

Private Sub Command2_Click()
Adjust_Scales.Hide
End Sub

Private Sub Form_Paint()
Voltage_Min = Scale_V_Min
Voltage_Max = Scale_V_Max
Current_Min = Scale_C_Min
Current_Max = Scale_C_Max
Oscillation_Min = Scale_O_Min
Oscillation_Max = Scale_O_Max

If Scale_V_Auto = True Then
   Voltage_Auto.Value = 1
   Voltage_Min.Enabled = False
   Voltage_Max.Enabled = False
```

```
Else
   Voltage_Auto.Value = 0
   Voltage_Min.Enabled = True
   Voltage_Max.Enabled = True
End If

If Scale_C_Auto = True Then
   Current_Auto.Value = 1
   Current_Min.Enabled = False
   Current_Max.Enabled = False
Else
   Current_Auto.Value = 0
   Current_Min.Enabled = True
   Current_Max.Enabled = True
End If

If Scale_O_Auto = True Then
   Oscillation_Auto.Value = 1
   Oscillation_Min.Enabled = False
   Oscillation_Max.Enabled = False
Else
   Oscillation_Auto.Value = 0
   Oscillation_Min.Enabled = True
   Oscillation_Max.Enabled = True
End If

SamplesPerGraph = Samples_P_Graph

End Sub


Private Sub Voltage_Auto_Click()
If Voltage_Auto.Value = 1 Then
   Voltage_Min.Enabled = False
   Voltage_Max.Enabled = False
Else
   Voltage_Min.Enabled = True
   Voltage_Max.Enabled = True
End If
End Sub

Private Sub Current_Auto_Click()
If Current_Auto.Value = 1 Then
   Current_Min.Enabled = False
   Current_Max.Enabled = False
Else
   Current_Min.Enabled = True
   Current_Max.Enabled = True
```

```
End If

End Sub

Private Sub Oscillation_Auto_Click()
If Oscillation_Auto.Value = 1 Then
   Oscillation_Min.Enabled = False
   Oscillation_Max.Enabled = False
Else
   Oscillation_Min.Enabled = True
   Oscillation_Max.Enabled = True
End If

End Sub
```

“General.bas”

```
Declare Function mciSendString Lib "winmm.dll" Alias _
   "mciSendStringA" (ByVal lpstrCommand As String, ByVal _
   lpstrReturnString As Any, ByVal uReturnLength As Long, ByVal _
   hwndCallback As Long) As Long


Public Const AppName = "WeldData"
Public Const MaxDataSampled = 40000
Public Const MaxGraphData = 1000 'Even numbers
Public Const Header = 11

Public DataPath As String
Public DataFileName As String
Public VideoFileName As String
Public DataFrames As Integer
Public VideoFrames As Integer
Public MaxFrames As Integer
Public DataSampleRate As Single

Public i, j As Integer
Public k As Single

Public DataPtr, PrevDataPtr As Integer

Public V(MaxDataSampled) As Single
Public C(MaxDataSampled) As Single
Public O(MaxDataSampled) As Single

Public VIGraphData(MaxGraphData, 1) As Single
Public OSCGraphData(MaxGraphData, 1) As Single
Public VIOCursorX As Variant
```

```
Public CrossPlot2D(MaxDataSampled, 1) As Single
Public CrossPlot3D(MaxDataSampled, 2) As Single
Public Plot1D_1(MaxGraphData) As Single
Public Plot1D_2(MaxGraphData) As Single

Public Osc_Pks(MaxDataSampled) As Integer
Public Osc_PksNeg(MaxDataSampled) As Integer
Public Osc_PksCentre(MaxDataSampled) As Integer
Public Osc_PksIdx As Integer
Public Osc_PksNegIdx As Integer
Public Osc_PksCentreIdx As Integer
Public Osc_Frequency As Single
Public OscMin As Single
Public OscMax As Single

Public Pulse_Frequency As Single

Public Scale_V_Min As Integer
Public Scale_V_Max As Integer
Public Scale_V_Auto As Boolean
Public Scale_C_Min As Integer
Public Scale_C_Max As Integer
Public Scale_C_Auto As Boolean
Public Scale_O_Min As Integer
Public Scale_O_Max As Integer
Public Scale_O_Auto As Boolean
Public Samples_P_Graph As Integer

Public VideoAdjustScalePoints(2, 2) As Integer
Public VideoMeasurePoints(2, 2) As Integer
Public VideoAdjustScaleRatio As Single
Public VidBaseSuppLines_H_Idx As Integer
Public VidBaseSuppLines_V_Idx As Integer
Public VidBaseSuppLines_H_Pos(10) As Integer
Public VidBaseSuppLines_V_Pos(10) As Integer
Public VidBaseSuppLines_H_Selected As Integer
Public VidBaseSuppLines_V_Selected As Integer

Sub main()
Scale_V_Min = 0
Scale_V_Max = 45
Scale_V_Auto = False
Scale_C_Min = 0
Scale_C_Max = 400
Scale_C_Auto = False
Scale_O_Min = -10
Scale_O_Max = 10
```

```
Scale_O_Auto = False
Samples_P_Graph = 100

VidBaseSuppLines_H_Idx = 0
VidBaseSuppLines_H_Idx = 0
VidBaseSuppLines_H_Selected = 0
VidBaseSuppLines_V_Selected = 0

MainWindow.Show

End Sub
```

Appendix H. Results of the 3rd phase of experimentation

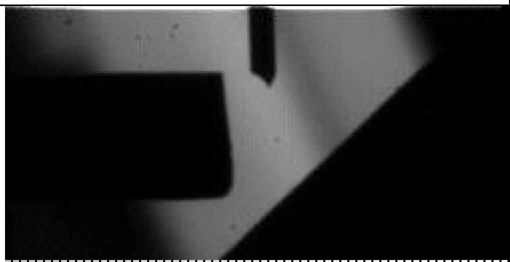
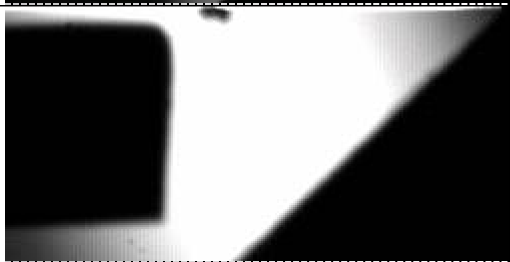
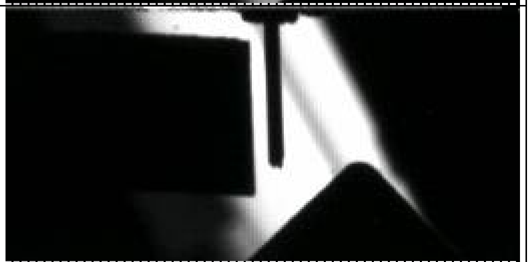
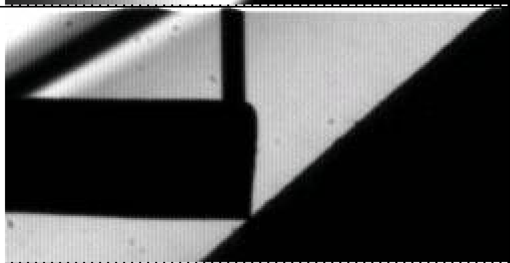
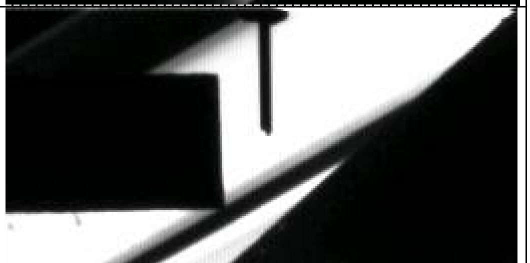
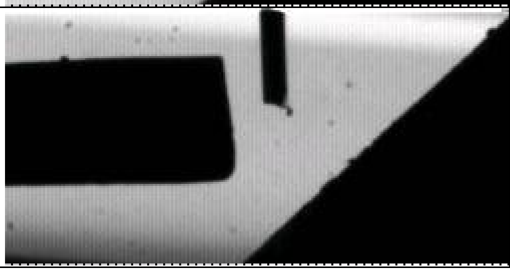
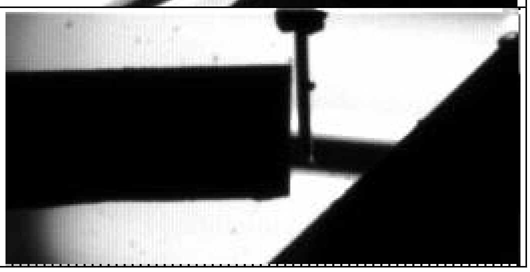
Trial	5 mm scale	10 mm scale
C1		
C2		
C3		
C4		
C5		

Figure H.1 - 5 mm and 10 mm calibration scales for WeldData arc image measurements

Table H.1 - GMAW-P peak voltage average values of trials C1 to C3

Trial	Osc Freq (Hz)	Osc_Pk_ Left (V)	Osc_Pk_ Right (V)	Osc_Mid_ Left (V)	Osc_Mid_ Right (V)	CRight-CLeft (V)	Left-Right (V)
C1.1	5	31.95	26.98	29.30	29.93	0.63	4.97
C1.2	10	31.17	26.47	29.28	29.14	-0.14	4.70
C1.3	15	31.39	26.98	29.90	29.45	-0.45	4.41
C1.4	20	31.99	27.19	29.73	29.53	-0.21	4.80
C1.5	25	33.00	28.22	30.81	30.67	-0.14	4.78
C2.1	5	36.39	33.46	34.95	35.32	0.38	2.93
C2.2	10	34.56	30.94	33.12	33.14	0.02	3.62
C2.3	15	35.64	32.01	34.12	34.25	0.13	3.63
C2.4	20	35.72	31.78	33.53	33.62	0.09	3.95
C2.5	25	35.40	31.32	33.47	33.52	0.05	4.07
C3.1	1	30.40	28.37	29.79	29.91	0.11	2.04
C3.2	3	31.64	27.30	29.54	29.69	0.15	4.35
C3.3	5	32.78	28.17	30.52	30.58	0.06	4.61
C3.4	7	31.94	27.44	30.05	29.74	-0.31	4.51
C3.5	9	32.84	28.63	30.95	30.99	0.04	4.20

Table H.2 – GMAW voltage average values of trials C4 and C5

Trial	Osc Freq (Hz)	Osc_Pk_ Left (V)	Osc_Pk_ Right (V)	Osc_Mid_ Left (V)	Osc_Mid_ Right (V)	CRight-CLeft (V)	Left-Right (V)
C4.1	5	27.15	26.11	26.04	26.94	0.90	1.04
C4.2	15	27.43	25.77	26.12	27.07	0.95	1.66
C4.3	25	27.30	25.65	25.86	27.12	1.25	1.64
C5.1	3	27.20	26.24	25.94	26.69	0.75	0.96
C5.2	6	26.87	25.87	26.04	26.83	0.78	1.00
C5.3	9	27.00	25.85	25.86	26.74	0.87	1.15
C5.4	12	27.31	25.52	26.02	26.91	0.89	1.80
C5.5	15	27.20	25.42	25.83	26.70	0.87	1.77

Table H.3 - GMAW current average values of trials C4 and C5

Trial	Osc Freq (Hz)	Osc_Pk_ Left (A)	Osc_Pk_ Right (A)	Osc_Mid_ Left (A)	Osc_Mid_ Right (A)	CRight-CLeft (A)	Left-Right (A)
C4.1	5.00	149.96	170.69	166.75	148.65	-18.10	20.73
C4.2	15.00	147.76	175.01	155.09	166.78	11.69	27.25
C4.3	25.00	148.69	175.22	162.12	165.69	3.57	26.53
C5.1	3.00	150.45	171.47	172.77	160.15	-12.62	21.02
C5.2	6.00	152.43	173.17	163.68	162.72	-0.96	20.74
C5.3	9.00	152.69	175.75	163.49	165.87	2.38	23.07
C5.4	12.00	149.65	179.86	159.49	168.10	8.61	30.20
C5.5	15.00	151.91	180.71	159.21	167.94	8.74	28.80

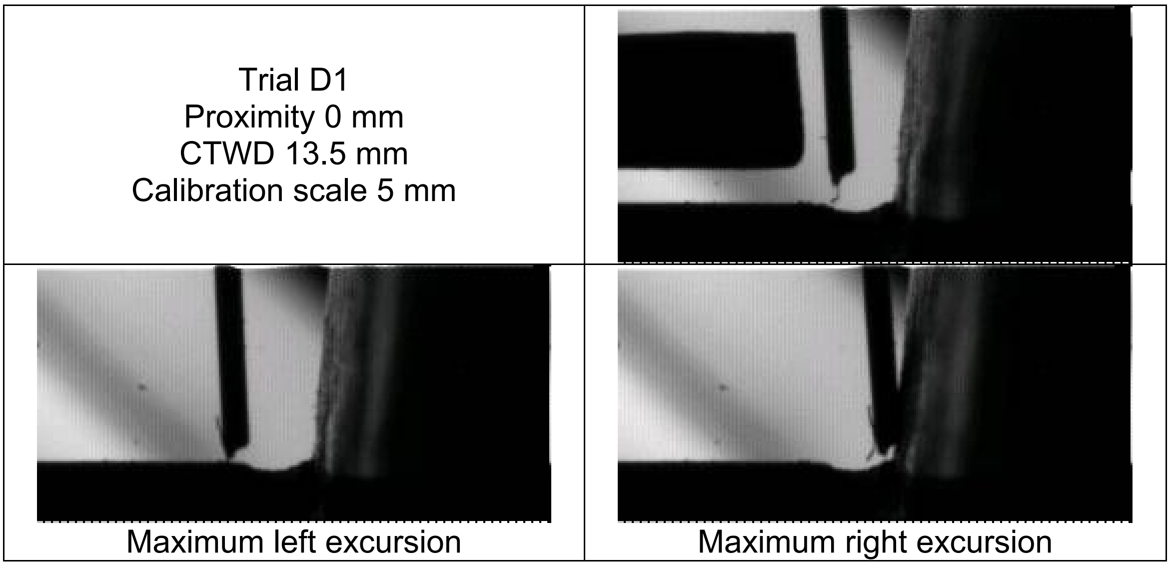


Figure H.2 - Calibration scale (top) and sidewall distances (bottom) of trial D1

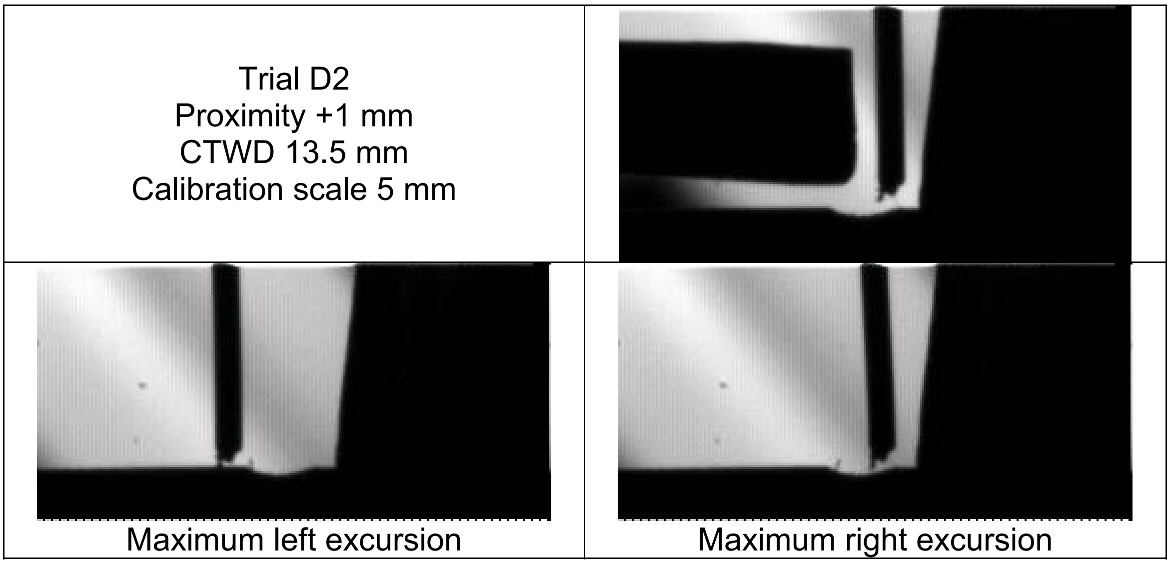


Figure H.3 - Calibration scale (top) and sidewall distances (bottom) of trial D2

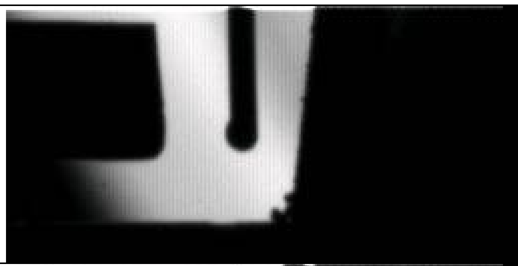
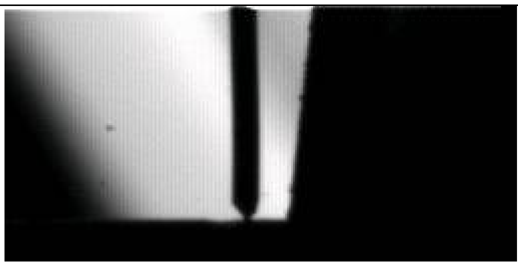
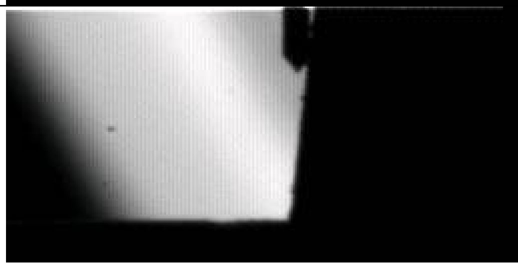
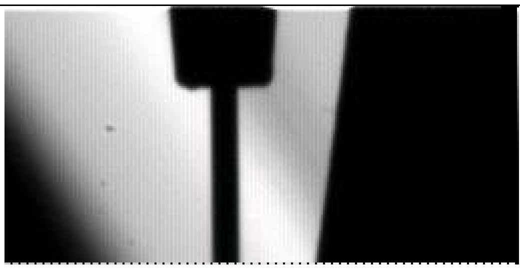
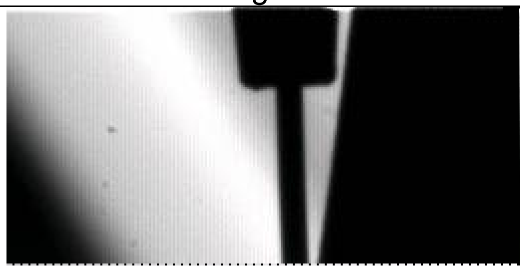
Trial D3 Proximity -1 mm CTWD 13.5 mm Calibration scale 5 mm	
 Maximum left excursion	 Maximum right excursion
 Contact-tip at maximum left	 Contact-tip at maximum right

Figure H.4 - Calibration scale (top) and sidewall distances (wire tip middle and contact-tip bottom) of trial D3

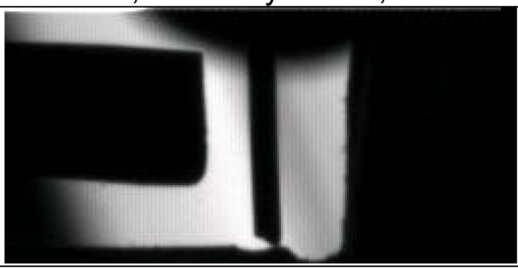
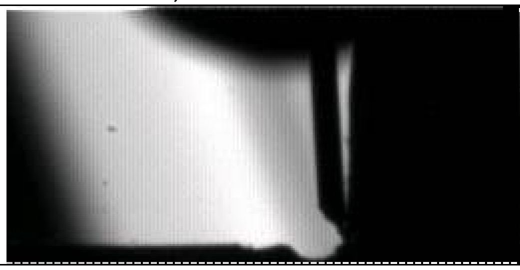
Trial D4 ; Proximity 0 mm ; CTWD 13 mm to 17 mm ; Calibration scale 5 mm	
	

Figure H.5 - Calibration scale (left) and sidewall distance (right) of trial D4

Table H.4 - GMAW-P peak voltage average values of trials D1 to D3

Trial	Osc Freq (Hz)	Osc_Pk_Left (V)	Osc_Pk_Right (V)	Osc_Mid_Left (V)	Osc_Mid_Right (V)
D1.1	5.00	28.25	26.14	29.46	29.66
D1.2	10.00	29.39	24.85	28.85	29.47
D1.3	15.00	29.31	25.39	28.56	28.57
D1.4	20.00	30.27	26.97	29.49	29.36
D1.5	25.00	30.10	28.75	29.84	29.64
D2.1	5.00	29.26	27.90	28.79	28.95
D2.2	10.00	29.11	28.48	28.90	29.16
D2.3	15.00	29.07	28.69	29.02	29.11
D2.4	20.00	29.49	29.62	29.44	29.72
D2.5	25.00	30.13	30.27	29.96	30.21
D3.1	5.00	30.71	21.81	27.45	27.41
D3.2	10.00	30.76	22.15	28.81	28.75
D3.3	15.00	30.35	21.08	28.28	28.46
D3.4	20.00	31.62	21.10	28.23	28.87
D3.5	25.00	31.16	20.50	27.61	27.48

Table H.5 - GMAW-P peak voltage average values of trial D4

Trial	CTWD (mm)	Osc_Pk_Left (V)	Osc_Pk_Right (V)	Osc_Mid_Left (V)	Osc_Mid_Right (V)
D4.1	13	29.83	21.84	27.96	28.34
D4.2	14	30.12	24.30	28.58	29.36
D4.3	15	32.61	25.11	30.81	30.52
D4.4	16	33.86	24.26	30.50	31.66
D4.5	17	34.92	25.65	32.36	32.06

Figure H.6 – Weld metal penetrations for trials D1 to D3

Trial	Oscillation (Hz)	Bottom penetration (mm)	Sidewall penetration (mm)
D1.1	5	1.65	0.74
D1.2	10	1.72	0.65
D1.3	15	1.84	0.37
D1.4	20	1.68	0.28
D1.5	25	2	0.56
D2.1	5	2.07	0.23
D2.2	10	1.98	0.14
D2.3	15	1.75	0.25
D2.4	20	1.75	0.21
D2.5	25	0.93	0.16
D3.1	5	0.42	0.74
D3.2	10	1.02	0.67
D3.3	15	1	0.79
D3.4	20	0.93	0.72
D3.5	25	0.63	0.67

Figure H.7 – Weld metal penetrations for trial D4

Trial	CTWD (mm)	Bottom penetration (mm)	Sidewall penetration (mm)
D4.1	13	1.41	1.28
D4.2	14	1.89	1.37
D4.3	15	1.63	1.72
D4.4	16	0.88	0.98
D4.5	17	1	0.65

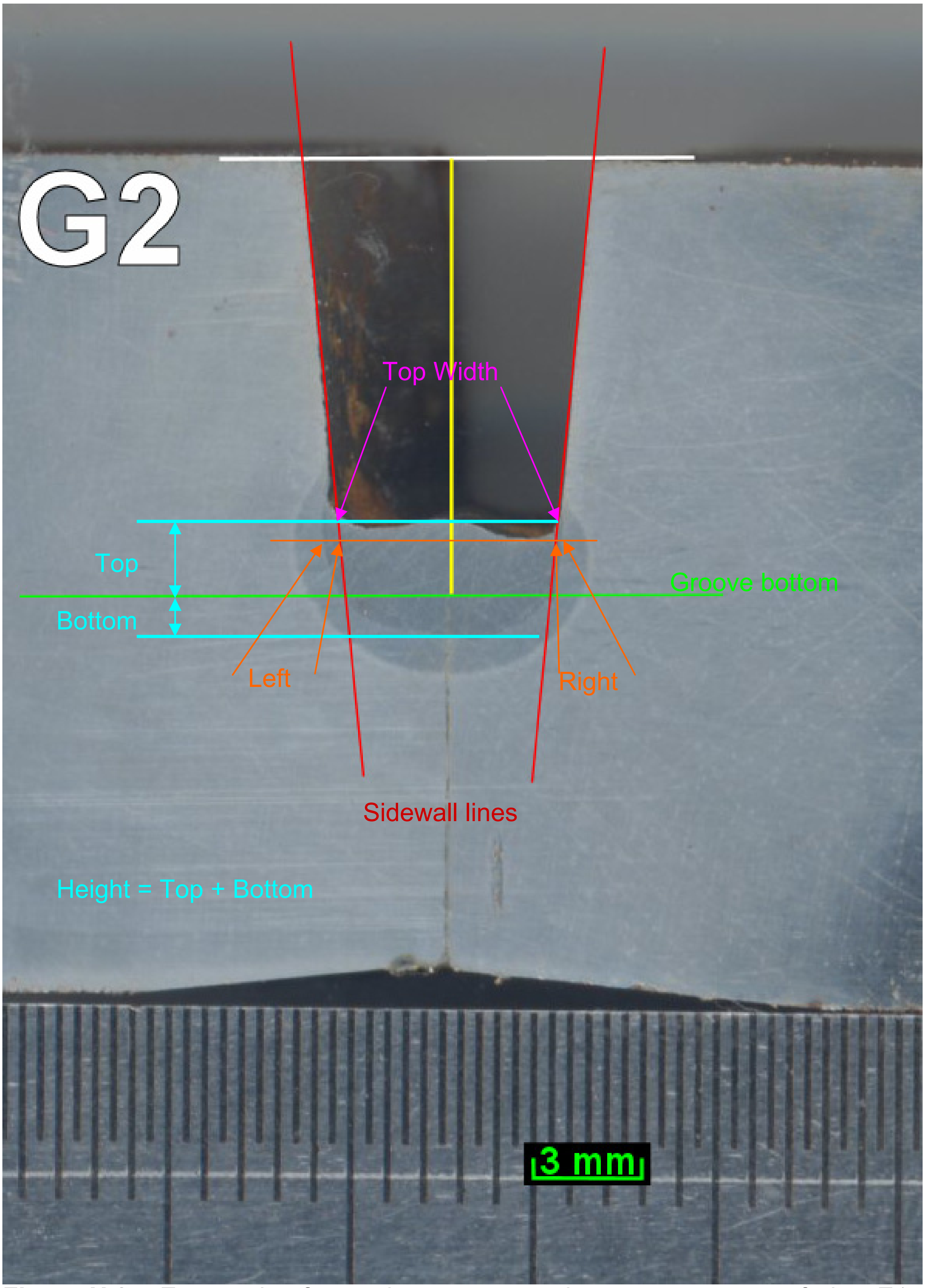


Figure H.8 – Explanation for weld metal penetration measurements of trials E1 to E8

Trial E1 ; Proximity -0.5 mm ; Groove width 2.5 mm ; Oscillation width 2.5 mm

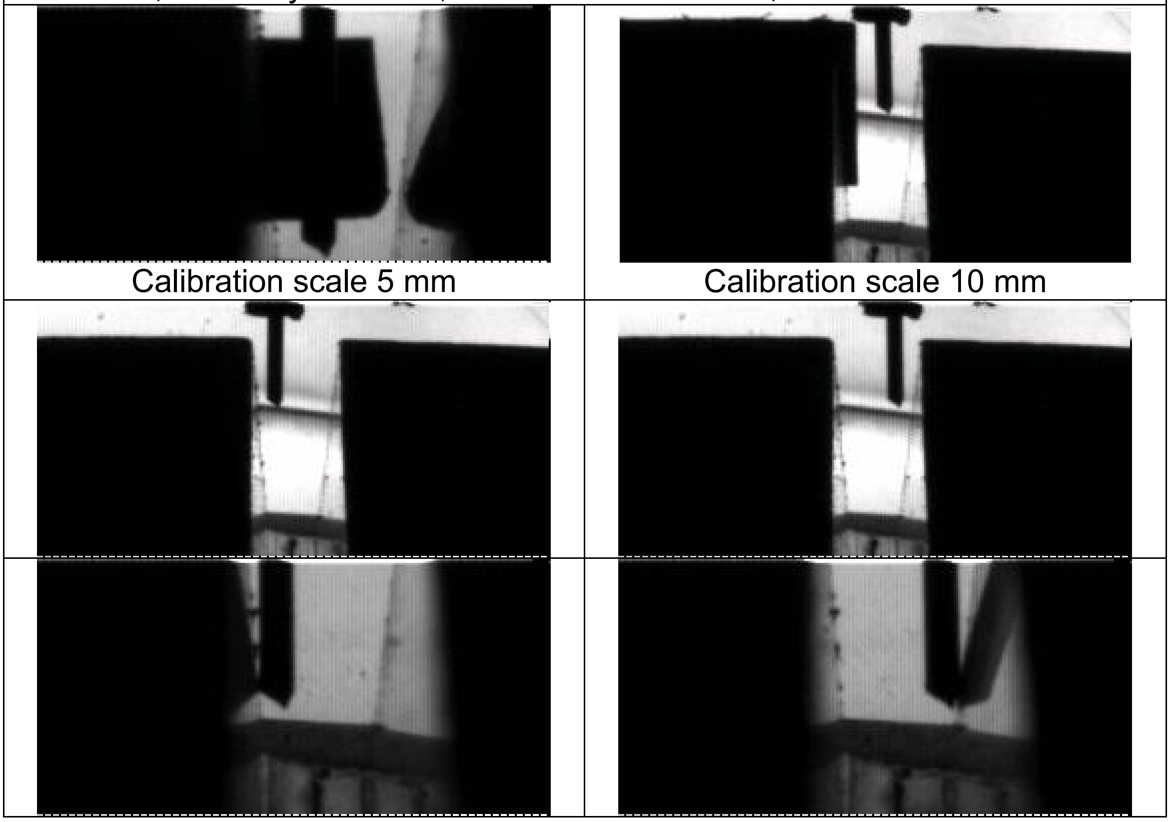


Figure H.9 - Calibration scale (top) and sidewall distances of trial E1

Table H.6 – Weld metal penetration and bead dimensions of trial E1

		Penetration (mm)				Bead dimensions (mm)		
Trial	**Osc Freq. (Hz)**	**Bottom**	**Left**	**Right**	**Top**	**Top width**	**Height**	**Width/Depth Avg. 0.948**
E1.1	5	0	0.21	0.59	4.1	4.55	4.1	1.0179
E1.2	10	0	0.42	0.59	4.2	4.47	4.2	0.9911
E1.3	15	0	0.55	0.63	4.1	4.47	4.1	1.03
E1.4	20	0	0.16	0.46	4.36	4.26	4.36	0.953
E1.5	25	0	0.03	0.08	4.89	3.59	4.89	0.7479

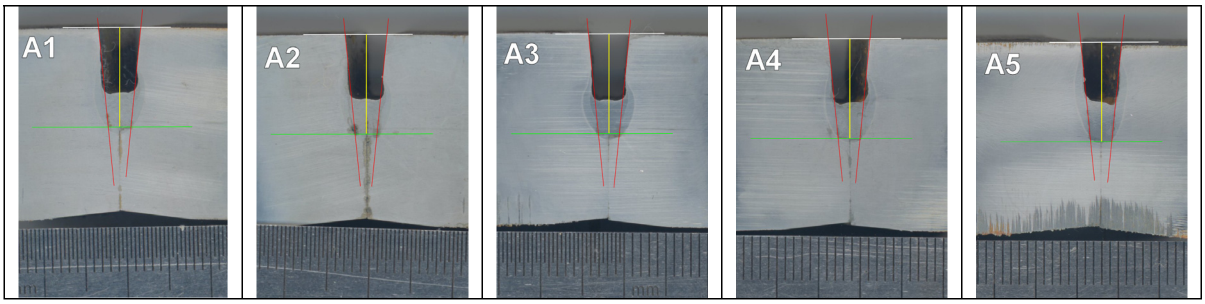


Figure H.10 – Bead profiles from trials E1.1 to E1.5

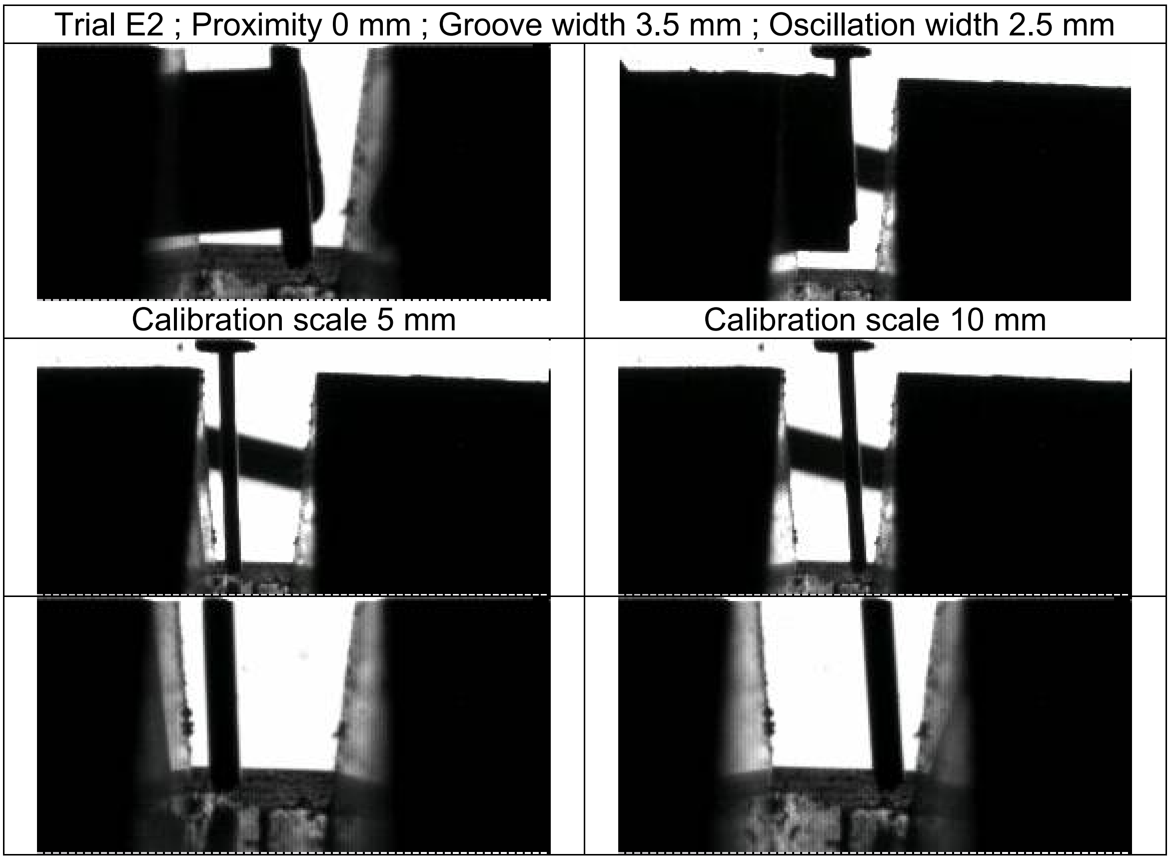


Figure H.11 - Calibration scale (top) and sidewall distances of trial E2

Table H.7 – Weld metal penetration and bead dimensions of trial E2

		Penetration (mm)				Bead dimensions (mm)		
Trial	**Osc Freq. (Hz)**	**Bottom**	**Left**	**Right**	**Top**	**Top width**	**Height**	**Width/Depth Avg. 1.4325**
E2.1	5	1.56	0.55	0.55	2.58	5.74	4.14	1.3865
E2.2	10	1.1	0.55	0.25	2.83	5.59	3.93	1.4224
E2.3	15	1.06	0.59	0.17	2.76	5.65	3.82	1.4791
E2.4	20	0.56	0.29	0.08	3.33	5.61	3.89	1.4422

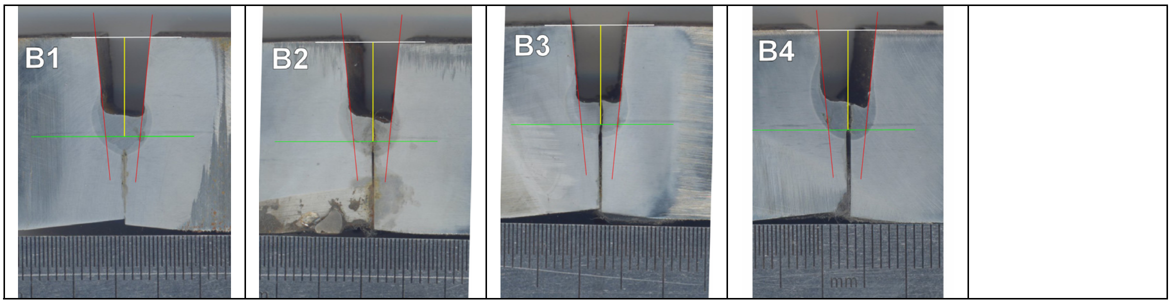


Figure H.12 – Bead profiles from trials E2.1 to E2.4

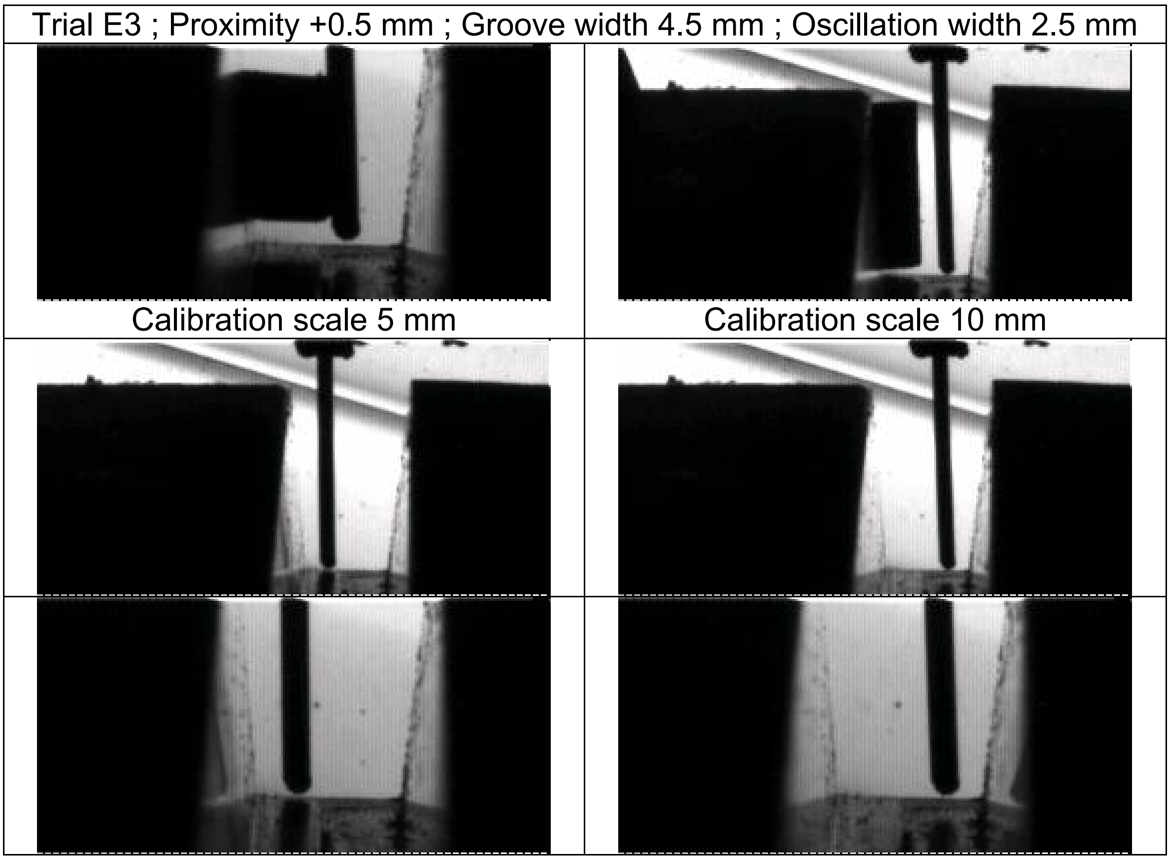


Figure H.13 - Calibration scale (top) and sidewall distances of trial E3

Table H.8 – Weld metal penetration and bead dimensions of trial E3

		Penetration (mm)				Bead dimensions (mm)		
Trial	**Osc Freq. (Hz)**	**Bottom**	**Left**	**Right**	**Top**	**Top width**	**Height**	**Width/Depth Avg. 1.5626**
E3.1	5	1.32	0.26	0.42	2.56	5.65	3.88	1.4562
E3.2	10	1.43	0.29	0.29	2.23	5.65	3.66	1.5437
E3.3	15	1.44	0.38	0.33	2.2	5.7	3.64	1.5659
E3.4	20	1.32	0.46	0.29	2.38	5.99	3.7	1.6189
E3.5	25	1.07	0.51	0.25	2.48	5.78	3.55	1.6282

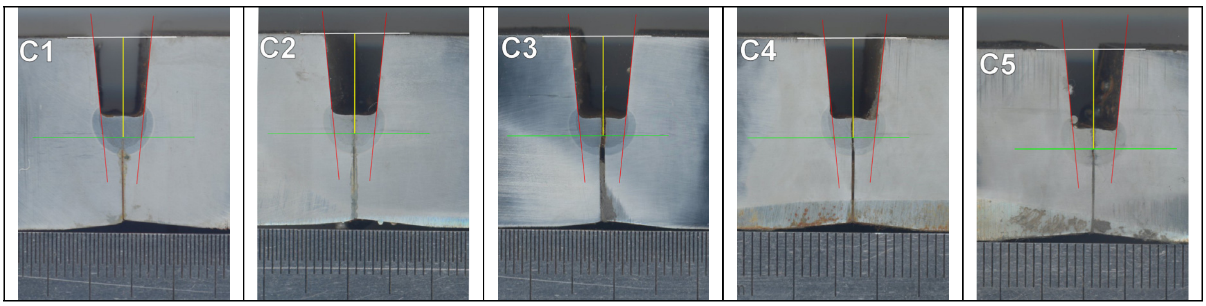


Figure H.14 – Bead profiles from trials E3.1 to E3.5

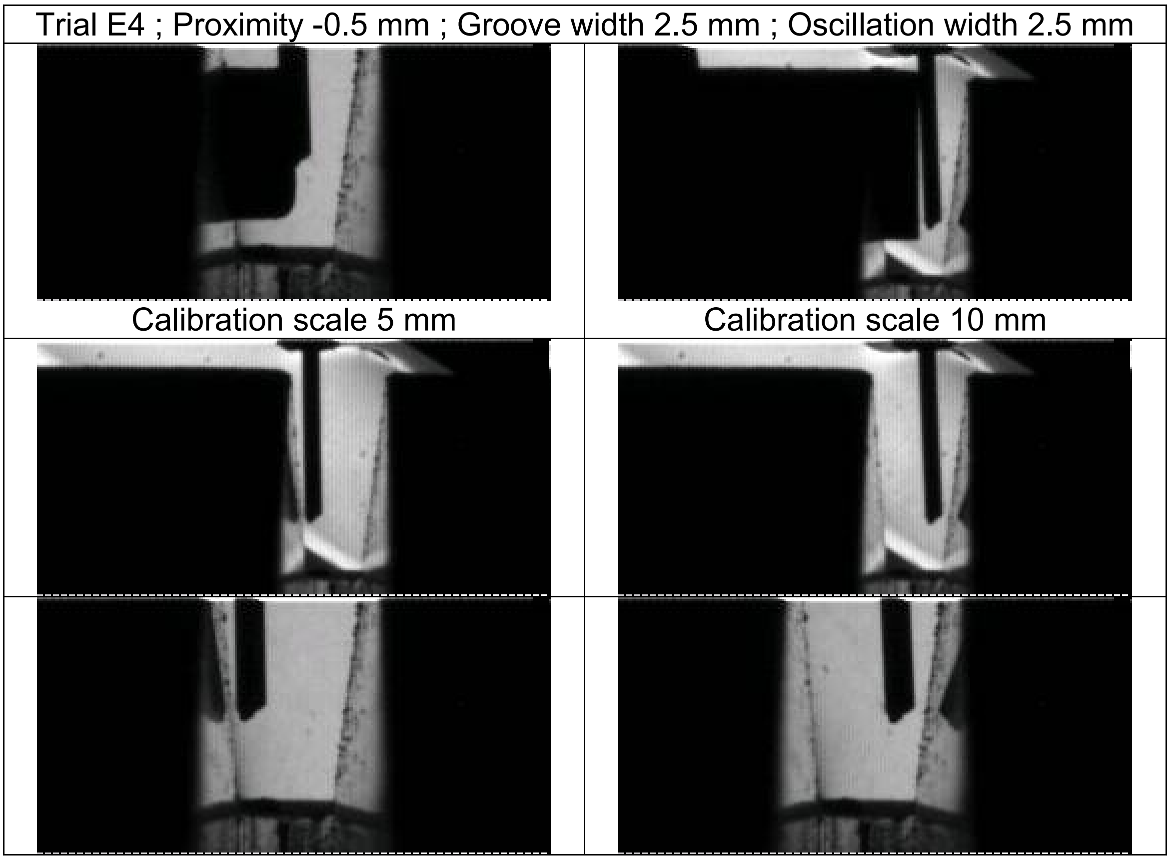


Figure H.15 - Calibration scale (top) and sidewall distances of trial E4

Table H.9 – Weld metal penetration and bead dimensions of trial E4

		Penetration (mm)				Bead dimensions (mm)		
Trial	Osc Freq. (Hz)	Bottom	Left	Right	Top	Top width	Height	Width/Depth Avg. 1.0858
E4.1	5	0.67	0.38	0.63	3.51	4.55	4.18	1.0885
E4.2	10	0.46	0.5	0.51	3.72	4.51	4.18	1.0789
E4.3	15	0.46	0.55	0.46	3.68	4.55	4.14	1.099
E4.4	20	0.15	0.55	0.51	3.87	4.47	4.02	1.1119
E4.5	25	0	0.6	0.51	4.33	4.55	4.33	1.0508

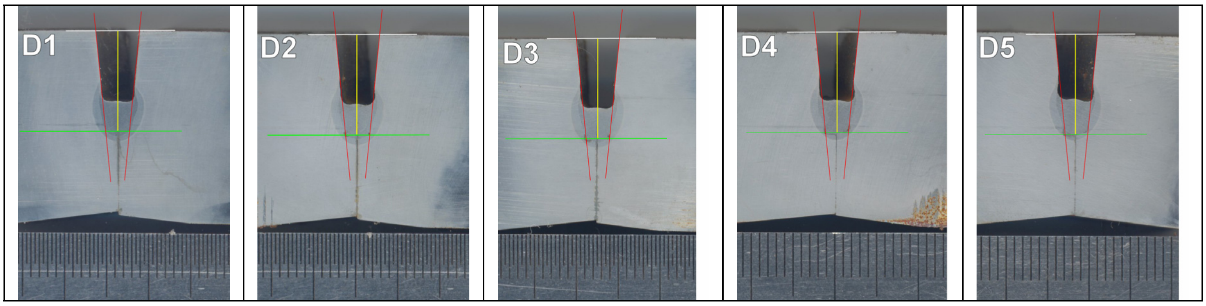


Figure H.16 – Bead profiles from trials E4.1 to E4.5

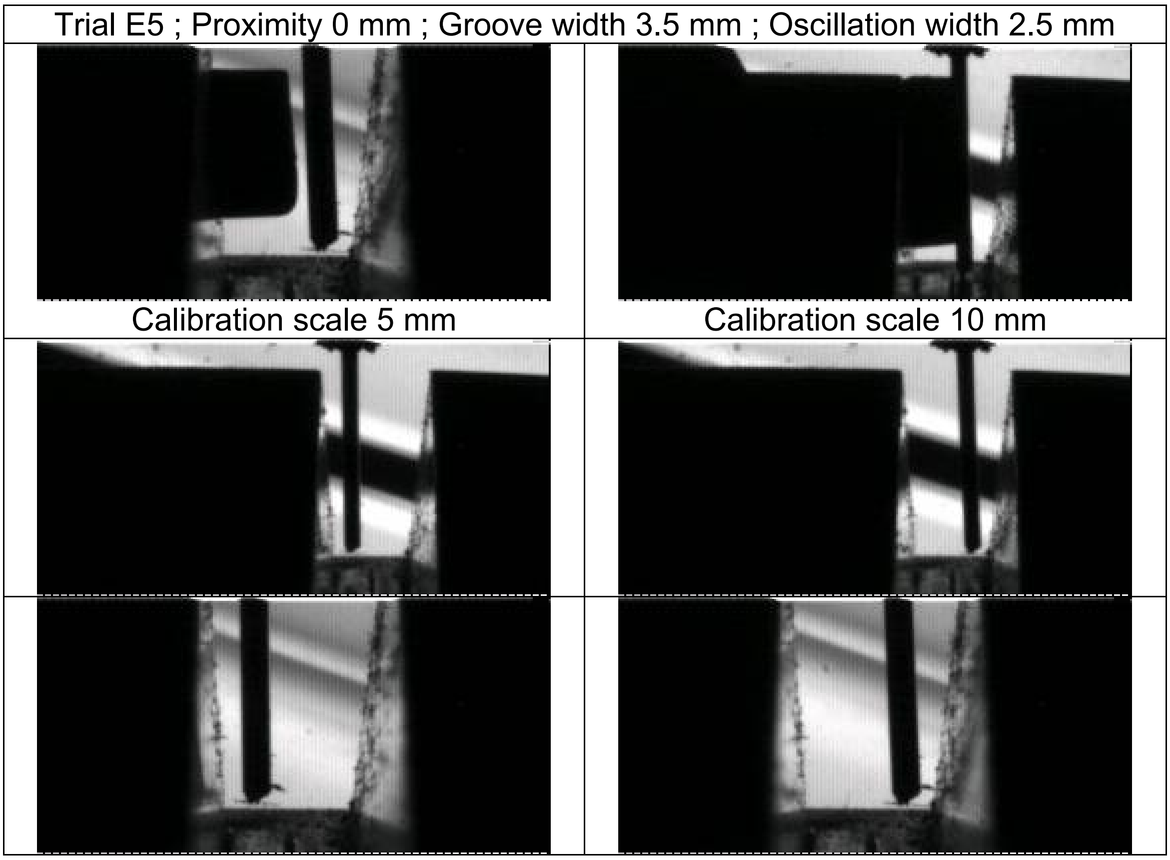


Figure H.17 - Calibration scale (top) and sidewall distances of trial E5

Table H.10 – Weld metal penetration and bead dimensions of trial E5

		Penetration (mm)				Bead dimensions (mm)		
Trial	**Osc Freq. (Hz)**	**Bottom**	**Left**	**Right**	**Top**	**Top width**	**Height**	**Width/Depth Avg. 1.3517**
E5.1	5	1.44	0.59	0.38	2.75	5.44	4.19	1.2983
E5.2	10	1.44	0.5	0.46	2.58	5.4	4.02	1.3433
E5.3	15	1.4	0.63	0.59	2.88	5.69	4.28	1.3294
E5.4	20	1.1	0.51	0.38	2.8	5.4	3.9	1.3846
E5.5	25	0.78	0.47	0.51	3.07	5.4	3.85	1.4026

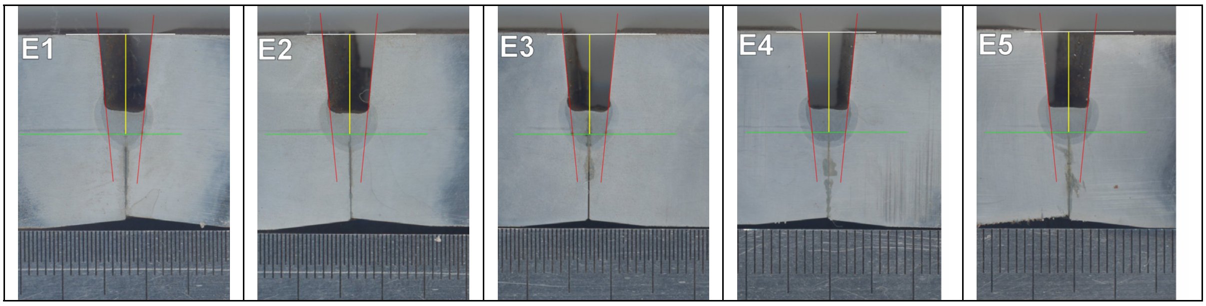


Figure H.18 – Bead profiles from trials E5.1 to E5.5

Trial E6 ; Proximity +0.15 mm ; Groove width 5 mm ; Oscillation width 3.7 mm

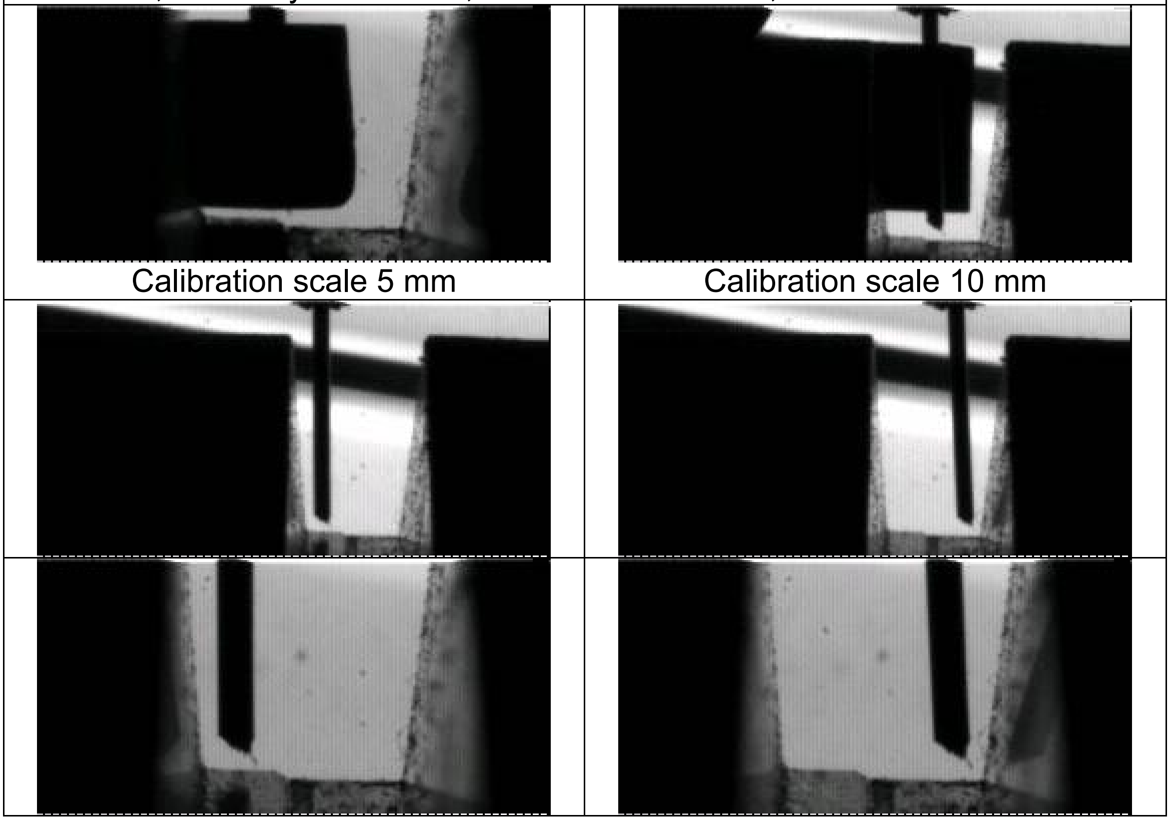


Figure H.19 - Calibration scale (top) and sidewall distances of trial E6

Table H.11 – Weld metal penetration and bead dimensions of trial E6

		Penetration (mm)				Bead dimensions (mm)		
Trial	**Osc Freq. (Hz)**	**Bottom**	**Left**	**Right**	**Top**	**Top width**	**Height**	**Width/Depth Avg. 1.8599**
E6.1	5	0.85	0.29	0.46	2.17	6.07	3.02	2.0099
E6.2	10	1.36	0.38	0.38	2.1	6.1	3.46	1.763
E6.3	15	0.94	0.38	0.47	2.43	6.2	3.37	1.8398
E6.4	20	0.55	0.42	0.29	2.8	6.12	3.35	1.8269

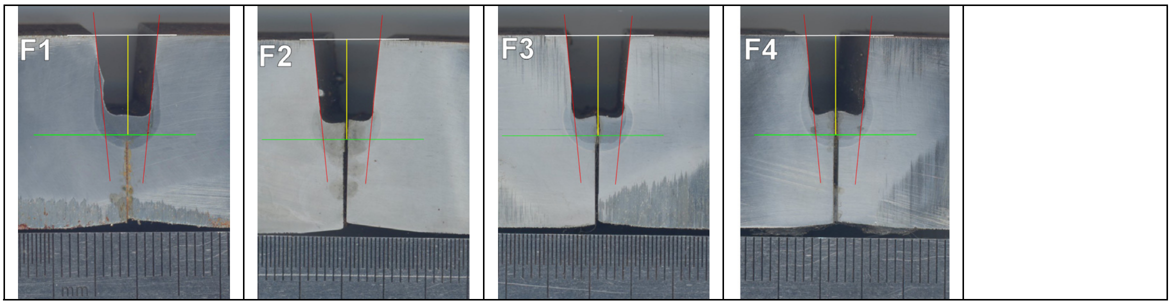


Figure H.20 – Bead profiles from trials E6.1 to E6.4

Trial E7 ; Proximity +0.65 mm ; Groove width 6 mm ; Oscillation width 3.7 mm

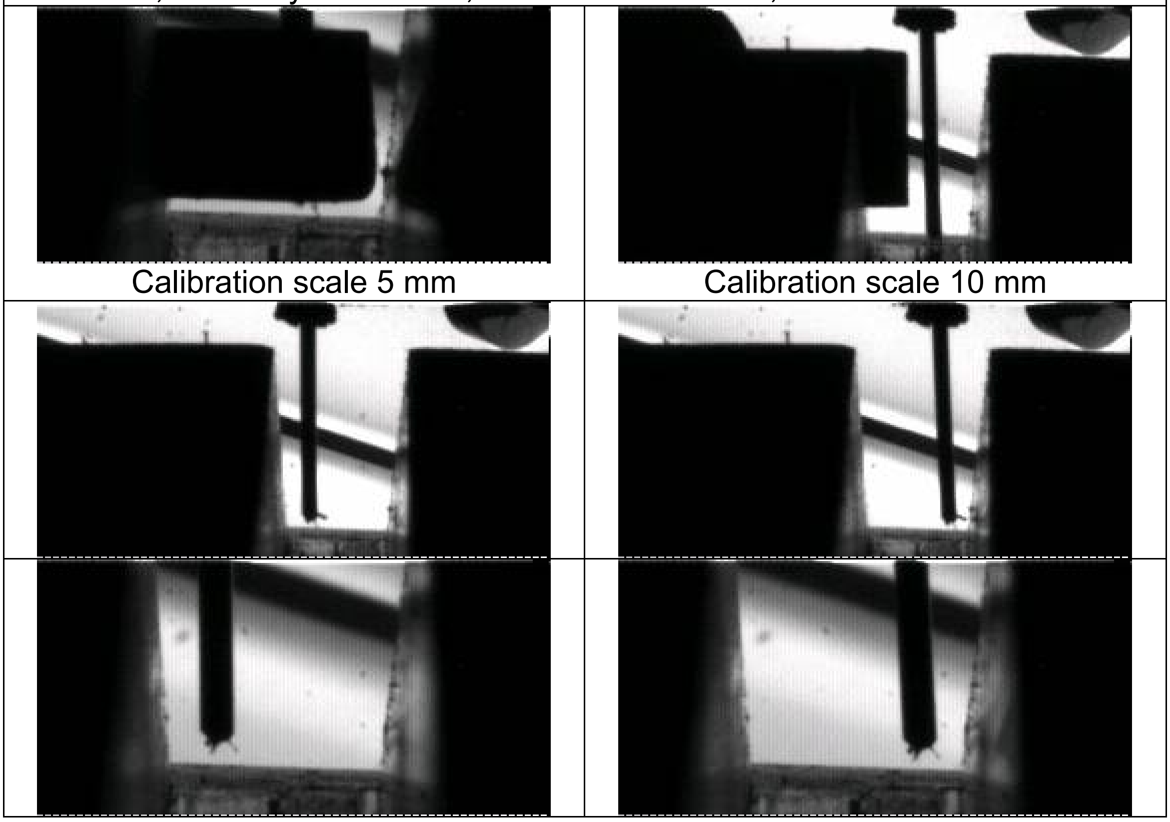


Figure H.21 - Calibration scale (top) and sidewall distances of trial E7

Table H.12 – Weld metal penetration and bead dimensions of trial E7

		Penetration (mm)				Bead dimensions (mm)		
Trial	**Osc Freq. (Hz)**	**Bottom**	**Left**	**Right**	**Top**	**Top width**	**Height**	**Width/Depth Avg. 2.0383**
E7.1	5	1.31	0.46	0.29	2.11	6.83	3.42	1.9971
E7.2	10	1.14	0.42	0.08	2.1	6.54	3.24	2.0185
E7.3	15	1.31	0.38	0.34	1.86	6.62	3.17	2.0883
E7.4	20	1.36	0.38	0.51	2.04	6.96	3.4	2.0471
E7.5	25	1.11	0.3	0.51	2.1	6.55	3.21	2.0405

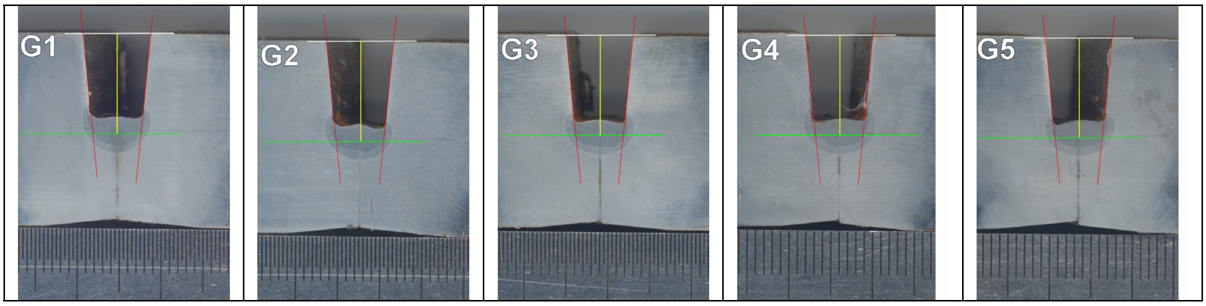


Figure H.22 – Bead profiles from trials E7.1 to E7.5

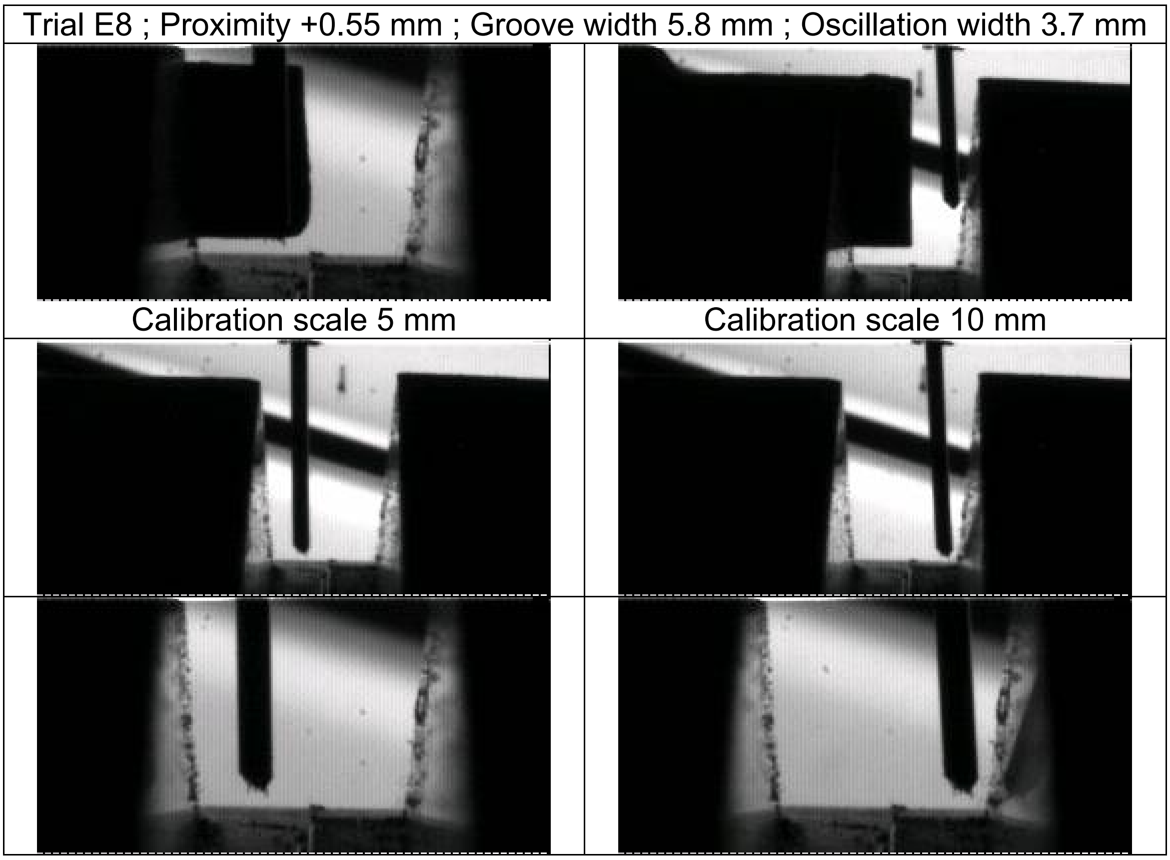


Figure H.23 - Calibration scale (top) and sidewall distances of trial E8

Table H.13 – Weld metal penetration and bead dimensions of trial E8

		Penetration (mm)				Bead dimensions (mm)		
Trial	Osc Freq. (Hz)	Bottom	Left	Right	Top	Top width	Height	Width/Depth Avg. 1.9252
E8.1	5	1.06	0.5	0.46	2.54	6.76	3.6	1.8778
E8.2	10	1.09	0.21	0.5	2.11	6.41	3.2	2.0031
E8.3	15	1.35	0.34	0.55	2.07	6.58	3.42	1.924
E8.4	20	1.19	0.26	0.42	2.25	6.5	3.44	1.8895
E8.5	25	0.85	0.25	0.47	2.52	6.51	3.37	1.9318

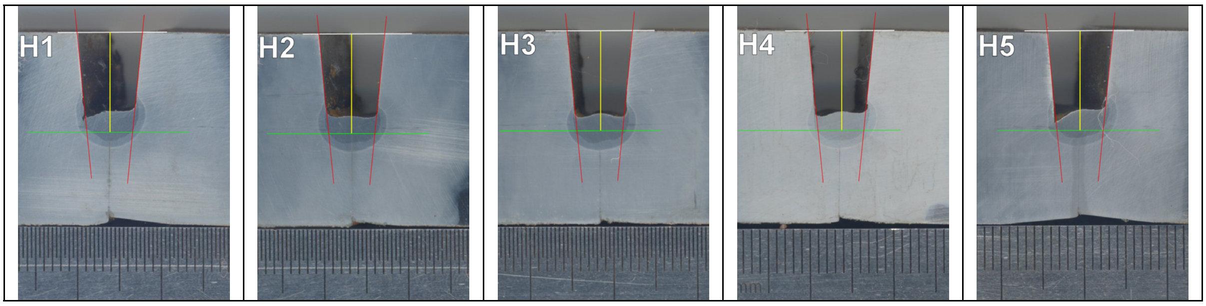


Figure H.24 – Bead profiles from trials E8.1 to E8.5

Table H.14 – Peak voltage average values from trials E1 to E8

Trial	Osc Freq (Hz)	Osc_Pk_Left (V)	Osc_Pk_Right (V)	Osc_Mid_Left (V)	Osc_Mid_Right (V)
E1.1	5.00	20.94	23.85	27.38	25.78
E1.2	10.00	21.08	24.19	27.27	25.33
E1.3	15.00	20.39	23.21	26.67	25.36
E1.4	20.00	19.59	23.88	27.62	26.43
E1.5	25.00	21.16	25.13	27.74	27.42
E2.1	5.00	22.81	26.72	28.32	27.63
E2.2	10.00	21.43	28.28	27.90	26.46
E2.3	15.00	21.21	27.75	27.96	26.74
E2.4	20.00	19.83	30.15	28.71	27.16
E2.5	25.00	26.48	26.59	25.66	26.59
E3.1	5.00	26.71	26.51	27.19	27.51
E3.2	10.00	26.48	26.84	28.26	28.13
E3.3	15.00	26.09	26.42	27.73	27.76
E3.4	20.00	26.11	27.04	28.02	27.73
E3.5	25.00	25.97	26.66	27.64	28.01
E4.1	5.00	22.67	21.95	26.69	26.04
E4.2	10.00	22.26	23.38	26.86	26.51
E4.3	15.00	22.43	22.29	26.57	26.54
E4.4	20.00	22.14	22.57	27.01	26.90
E4.5	25.00	21.95	20.48	26.19	26.38
E5.1	5.00	24.41	27.47	28.87	28.80
E5.2	10.00	24.29	26.50	28.38	28.20
E5.3	15.00	25.51	26.63	29.22	29.26
E5.4	20.00	25.08	26.22	28.14	28.14
E5.5	25.00	25.17	26.54	28.23	28.71
E6.1	5.00	25.35	25.27	28.80	28.44
E6.2	10.00	24.05	24.83	28.77	28.63
E6.3	15.00	23.43	24.89	29.09	29.16
E6.4	20.00	20.83	25.79	30.12	29.58
E6.5	25.00	26.11	27.33	27.48	27.33
E7.1	5.00	25.88	26.40	28.59	28.31
E7.2	10.00	25.82	26.81	28.29	28.31
E7.3	15.00	26.01	26.40	28.23	28.27
E7.4	20.00	27.29	26.28	29.36	29.44
E7.5	25.00	27.28	25.83	28.48	28.82
E8.1	5.00	27.61	25.78	29.13	29.46
E8.2	10.00	27.23	24.76	27.91	28.13
E8.3	15.00	27.11	24.76	28.08	28.41
E8.4	20.00	28.92	24.17	28.93	28.75
E8.5	25.00	29.63	25.38	29.01	29.05