

# ProcessWare Manual

*ArcWare*

3HAC 5715-1  
BaseWare OS 3.2



The information in this document is subject to change without notice and should not be construed as a commitment by ABB Robotics Products AB. ABB Robotics Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Robotics Products AB be liable for incidental or consequential damages arising from use of this document or of the software and hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Robotics Products AB's written permission, and contents thereof must not be imparted to a third party nor be used for any unauthorized purpose. Contravention will be prosecuted.

Additional copies of this document may be obtained from ABB Robotics Products AB at its then current charge.

© ABB Robotics Products AB

Article number: 3HAC 7677-1  
Issue: M2000/Rev.1

ABB Robotics Products AB  
S-721 68 Västerås  
Sweden

## CONTENTS

	Page
<b>Programming ArcWare .....</b>	<b>3</b>
Programming Arc Welding .....	3
Manual functions when program execution has been stopped .....	8
Manual functions during program execution.....	15
<b>Special ArcWare windows .....</b>	<b>18</b>
Window: Production .....	18
Window: Program Test .....	19
Window when executing .....	20
<b>Arc Welding .....</b>	<b>21</b>
Activating arc welding parameters .....	21
Defining arc welding systems.....	22
Defining measurements .....	23
Defining arc welding functions .....	24
Defining arc welding equipment .....	27
Defining weldguide arc welding sensor .....	30
<b>ArcL - Arc welding with linear motion ArcL1 ArcL2 .....</b>	<b>33</b>
Example .....	33
Arguments .....	34
Program execution.....	37
Limitations.....	38
Fault management.....	38
Example .....	39
Syntax .....	40
Related information .....	40
<b>ArcC - Arc welding with circular motion ArcC1 ArcC2 .....</b>	<b>41</b>
Example .....	41
Arguments .....	42
Program execution.....	45
Limitations.....	46
Error management .....	46
Example .....	47
Syntax .....	48
Related information .....	48
<b>seamdata - Seam data .....</b>	<b>49</b>
Description.....	49
The welding sequence .....	50
Components .....	51

	Page
Structure .....	56
Related information .....	57
<b>welddata - Weld data .....</b>	<b>59</b>
Description .....	59
Example .....	60
The welding sequence .....	61
Components .....	62
Examples .....	64
Structure .....	66
Related information .....	66
<b>weavedata - Weave data .....</b>	<b>67</b>
Description .....	67
Components .....	67
Limitations .....	72
Structure .....	73
Related information .....	73
<b>ArcRefresh - Refresh arcweld data .....</b>	<b>75</b>
Example .....	75
Syntax .....	75
Related information .....	76
<b>ArcKill - Kill the AW process .....</b>	<b>77</b>
Example .....	77
Syntax .....	78
Related information .....	78
<b>arcdata - Arc process data .....</b>	<b>79</b>
Description .....	79
Components .....	79
Example .....	80
Structure .....	81
Related information .....	81

## Programming ArcWare

Before you start to program arc welding instructions, you must configure the arc welding system and any external axes.

This is described in Chapter 12 of this manual, System Parameters.

### Programming Arc Welding

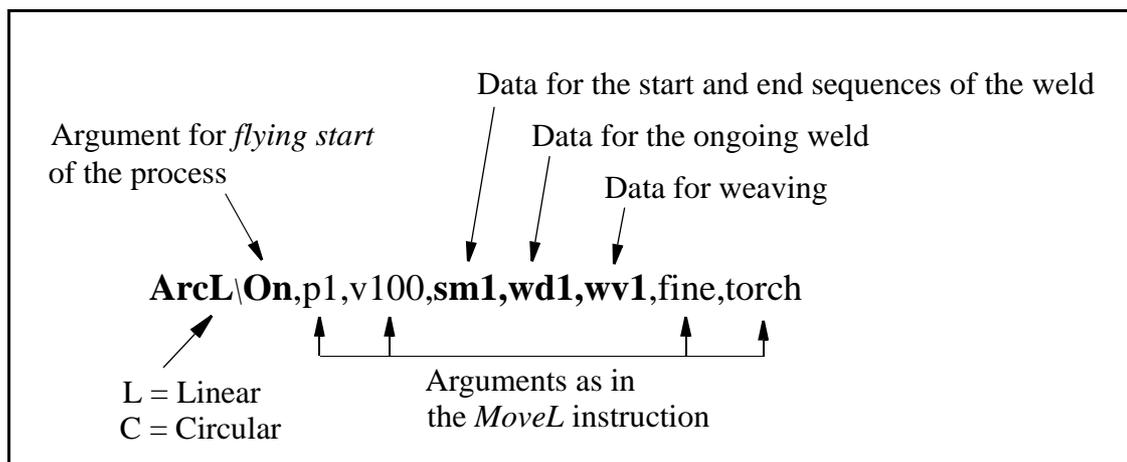
#### Program structure

When there are several seams to be welded on an object, the welding sequence may be of critical importance for the quality of the object. The risk of deformation due to thermal stress can be reduced by choosing the seam welding sequence. It is often best to make a specific routine for this, *object routine*, with all the seams specified in the correct order. When the object is placed in a positioner, its orientation can also be specified in the object routine.

The object routine can call a welding routine for each seam to be welded.

#### Arc welding instructions

An arc welding instruction basically contains the same types of information as a positioning instruction. However, each arc welding instruction includes a further three arguments, *seam*, *weld* and *weave*, that serve as data for the arc welding process (data types: *seamdata*, *welddata* and *weavedata*).



The speed argument in the instruction, *v100*, is only valid during step-wise execution (forward or backward) and the welding process will in this case automatically be inhibited. Otherwise, during normal execution, the *process speed* in different phases of the process is included as components of seam and weld data.

For more information on programming this type of instruction, see *Programming arc welding instructions* on page 5.

## Defining arc welding data

Before starting to program arc welding instructions, you should define the arc welding data that is to be used. This data is divided into three types:

- *seamdata*; describes how the seam is to be started and ended,
- *welddata*; describes the actual welding phase,
- *weavedata*; describes how any weaving is to be carried out.

The exact components of the above data depend on the configuration of the robot at the time.

Normally, data is stored as a part of the program. However, when data is to remain in the memory regardless of which program is loaded, it is stored in a system module.

- Open the *Program Data Types* window by choosing **View: Data Types**.
- Select the type *seamdata*, *welddata* or *weavedata* and press Enter .
- Press the function key *New*.

A window appears, displaying the name of the data (see Figure 1).

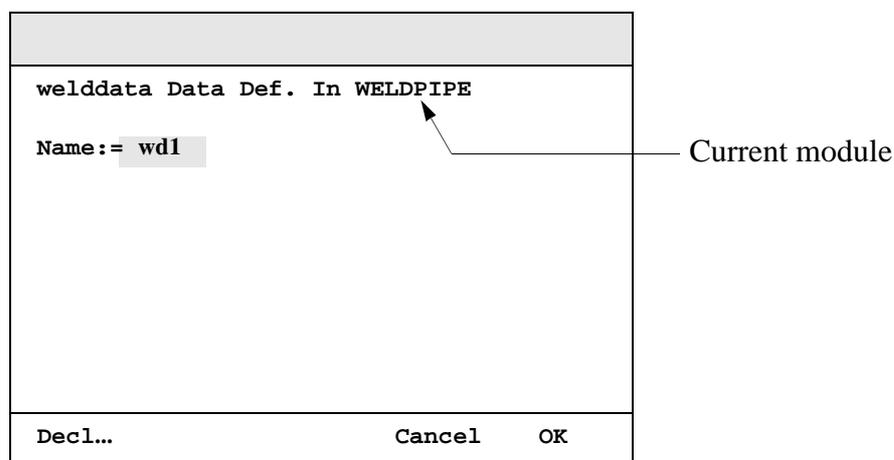


Figure 1 Definition of weld data.

- If you want to change the name press Enter  and specify a new name.
- Press the function key *Decl*.

A dialog box appears, displaying the basic weld data declaration (see Figure 2).

welddata Data Definition		
Name:=	wdl...	
Type:=	Persistent <input type="checkbox"/>	Global <input type="checkbox"/>
In Module:=	WELDPIPE	
=====2(7)		
wdl:		
weld_speed:=	0	(num)
weld_voltage:=	0	(num)
weld_wirefeed:=	0	(num)
org_weld_speed:=	0	(num)
org_weld_voltage:=	0	(num)
Cancel		OK

Figure 2 Definition of weld data.

- If you wish to save the data in another module, select the field *In Module* and press Enter . Choose the module in which the data is to be saved.
- Select the lower part of the window by pressing the List  key.
- Select the appropriate component in the data and specify the desired value. More information on the individual components can be found in RAPID Reference Manual.
- Choose **OK** to terminate the definition.

**Tip** It is sometimes easier to create new data by copying and modifying existing data.

---

### Programming arc welding instructions

- Jog the robot to the desired destination position.
- Call up the instruction pick list by choosing **IPL1: Motion & Process**.
- Select the instruction *ArcL* or *ArcC*.

The instruction will be added directly to the program, as illustrated in Figure 3. The arguments are set in relation to the last arc welding instruction that was programmed.

File	Edit	View	IPL1	IPL2
Program Instr			WELDPIPE/main	
			Motion&Proc	
			===== 1(2)	
ArcL\On,*,v100,sml,wdl,wvl,z->			1	ActUnit
ArcL\Off,*,v100,sml,wdl,wvl,z->			2	ArcC
			3	ArcKill
			4	ArcL
			5	ArcL\Off
			6	ArcL\On
			7	ArcRefresh
			8	DeactUnit
			9	More ↓
Copy	Paste	OptArg	ModPos	Test->

Figure 3 An arc welding instruction is added directly to the program.

The instruction is now ready for use. However, if an argument needs to be changed, one data can be replaced by another. The following example will illustrate this:

- Select the argument you wish to change (*sm1* in this example).
- Press Enter .

The window used to program instruction arguments appears. The selected argument is marked with a ? in front of it (see Figure 4). The lower part of the box displays all available seam data that can be selected.

Instruction Arguments				
ArcL\On, *, v100, ?sm1, wd1, wv1, z10, torch;				
Seam	sm1			
				1 (2)
New...	sm1	sm2		
sm3	sm4			
Next	Func	More...	Cancel	OK

Figure 4 The dialog box used to change seamdata.

- Select the desired seam data.
- Move to the next argument (weld data, *wd1*) by pressing *Next*.
- Repeat for all arguments.
- Choose **OK** to confirm the change.

The instruction is now ready for use.

### Example of an arc welding instruction

The seam illustrated in Figure 5 is to be welded. The seam line is represented by the thick line in the figure.

The xxxxx characters between points p10 and p20 designate a *flying start*. In other words, preparations (e.g. gas preflowing) for welding are carried out on the way to the starting-point, p20. The weld is terminated at point p80.

The weld data, *wd1*, applies until position p50 is reached, where a transition to *wd2* takes place.

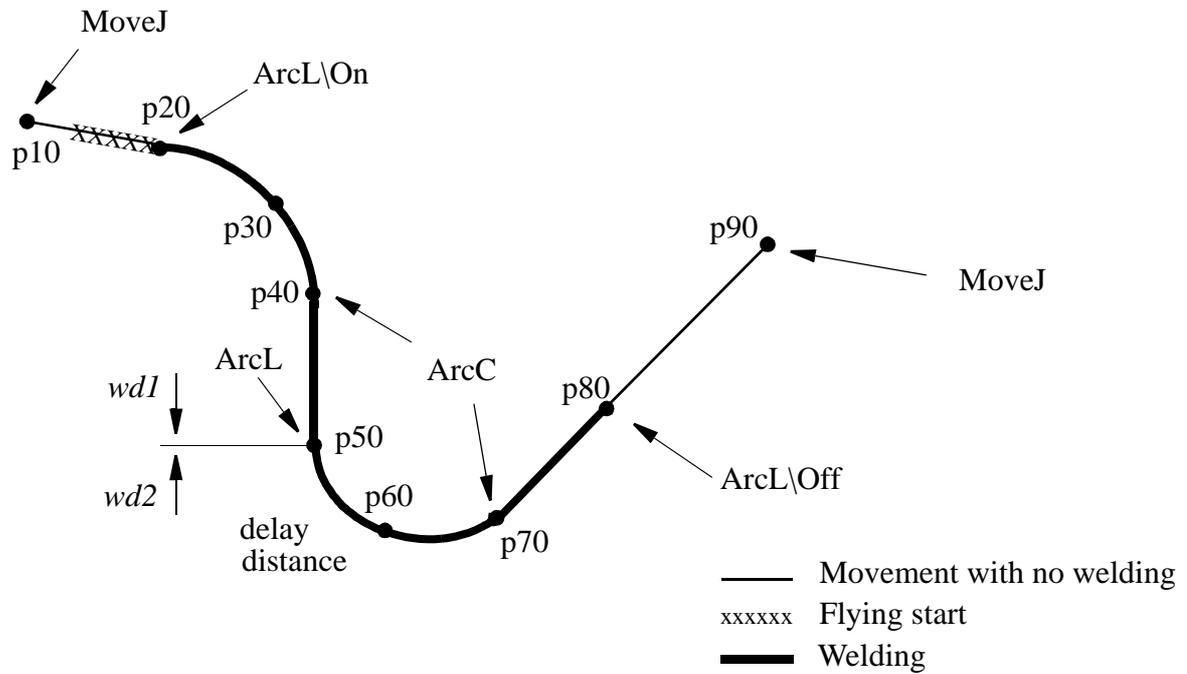


Figure 5 An example of an arc welding instruction.

The programming sequence for this weld could be written as follows:

```

MoveJ p10, v100, z10, torch;
ArcL\On, p20, v100, sm1, wd1, wv1, fine, torch;
ArcC p30, p40, v100, sm1, wd1, wv1, z10, torch;
ArcL p50, v100, sm1, wd1, wv1, z10, torch;
ArcC p60, p70, v100, sm1, wd2, wv1, z10, torch;
ArcL\Off, p80, v100, sm1, wd2, wv1, fine, torch;
MoveJ p90, v100, z10, torch;

```

If the seam is to be coordinated with an external axis, an argument of the type *workobject* must be included in all arc welding instructions except for the start instruction. For more information, see RAPID Reference Manual.

---

---

## Manual functions when program execution has been stopped

Arc welding functions in *Program Window*:

- Weld data tuning
- Weave data tuning
- Communicate with seamtracker sensor
- Process blocking
- Manual wirefeed
- Manual gas on/off
- Select arc welding system
- Changing tuning increments

Arc welding functions in *Production Window*:

- Manual wirefeed
- Manual gas on/off
- Select arc welding system
- Changing tuning increments

---

### Weld data tuning

Certain weld data components (*weld\_speed*, *weld\_wirefeed* and *weld\_voltage*) can be tuned using the weld data tuning function.

There are two stored values for the tunable data, namely, the present value and the so called “original value”. This allows you to be able to see how much the value has changed and to be able to revert to the original value.

When tuning, it is always the present value that is changed.

The original value can also be updated (i.e. it can be set to the same value as the present value).

The same changes can also be made in the *Program Data* window.

- Select **Arcweld: Weld Tuning**.

A dialog box appears where the latest used (present) welding data is displayed (see Figure 6).

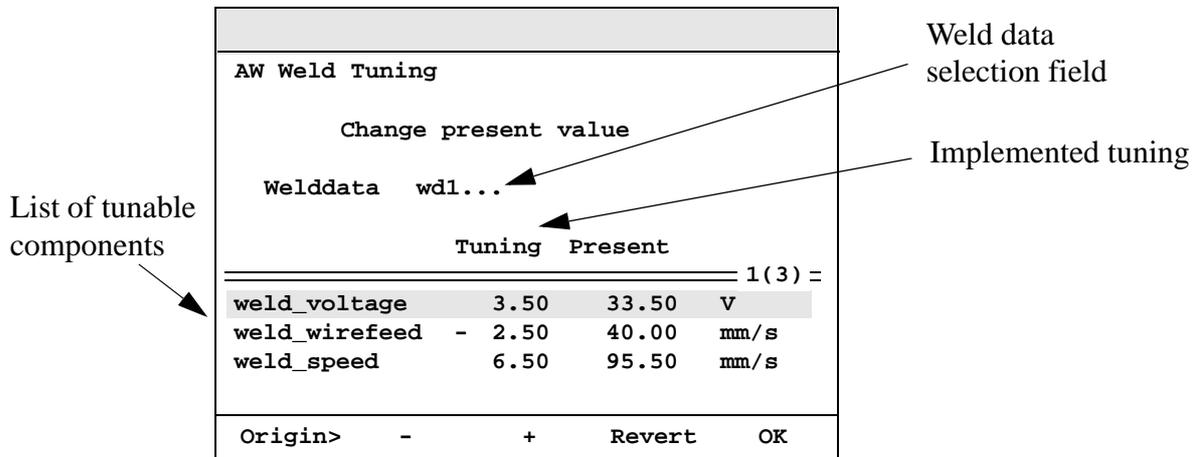


Figure 6 Dialog box for weld data tuning.

### Select welding data to be tuned

- Select the weld data selection field and press Enter .
- Select the desired welding data in the dialog box displayed (see Figure 6) and press **OK**.

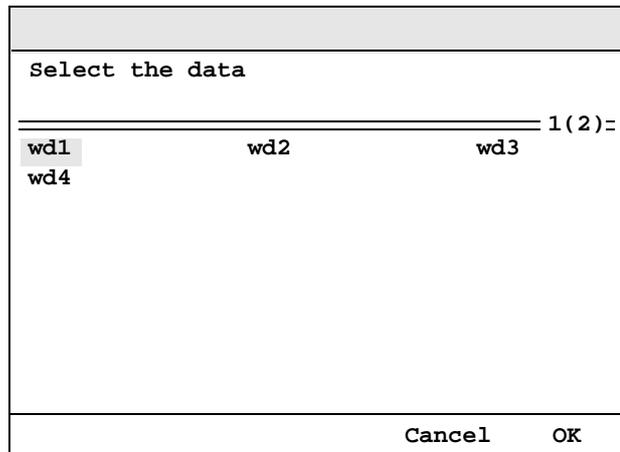


Figure 7 Dialog box for weld data tuning (select data).

### Tuning present values

Select the values to be tuned, in the lower part of the box, and press the function keys “-” or “+”. Each time these keys are pressed, the value will decrease/increase in increments. The tuning increment can be preset. For adjustment see *Changing tuning increments* on page 15.

### Resetting the present tuning value

- Press **Revert** to reset the tuning value.  
The present value will be reset to the original value.

*Updating the original value to the present value*

- Press **Origin**.

A dialog box will be displayed where the present value and the original value are displayed (see Figure 8).

AW Weld Tuning			
Update origin value			
Welddata wdl...			
Present Origin			
			1(3) =
weld_voltage	33.50	30.00	V
weld_wirefeed	40.00	42.50	mm/s
weld_speed	95.50	89.00	mm/s
Pres>	Update		OK

Figure 8 Dialog box for weld data tuning (origin value).

- Press **Update**.  
The original value will be reset to the present value.
- Press **Pres** to revert to the first dialog box.
- Press **OK** to terminate the dialog.

**Note** If more than one system is configured in the robot, the dialog for selection of AW systems can be used to select the corresponding datatype (*welddata*, *welddata1* or *welddata2*).

**Weave data tuning**

The weave data tuning dialog has exactly the same functions as the weld data tuning dialog.

The data types supported for tuning are *weavedata*, *weavedata1* and *weavedata2*, depending on the number of configured systems.

The tunable components are: *weave\_width*, *weave\_height* and *weave\_bias*.

**weave\_width**

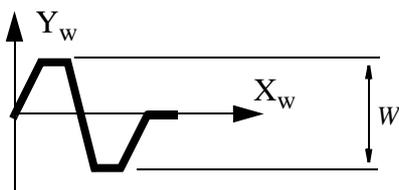


Figure 9 Weaving width (W).

**weave\_height**

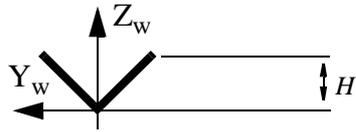


Figure 10 Weaving height (H).

**weave\_bias**

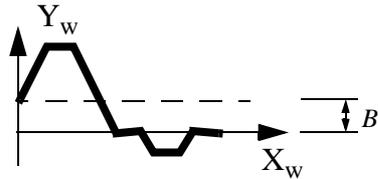


Figure 11 Weaving bias (B).

**Communicate with seamtracker sensor**

**Note** The seamtracker sensor communication functions can only be used if the robot is configured to use a seamtracker sensor.

- Select **Arcweld: Sensor**.

The following dialog box is displayed (see Figure 12).

<b>AW Manual Sensor On/Off</b>	
Joint: <input type="text" value="7"/>	
<b>Sensor coordinates</b>	
Y:	4.0
Z:	9.0
Gap:	0.8
(On)	Off
OK	

Figure 12 Dialog box for communication with seamtracker sensor.

This dialog has two functions:

- to switch the sensor on/off
- to get the current sensor data by selecting a joint number and pressing **On**.

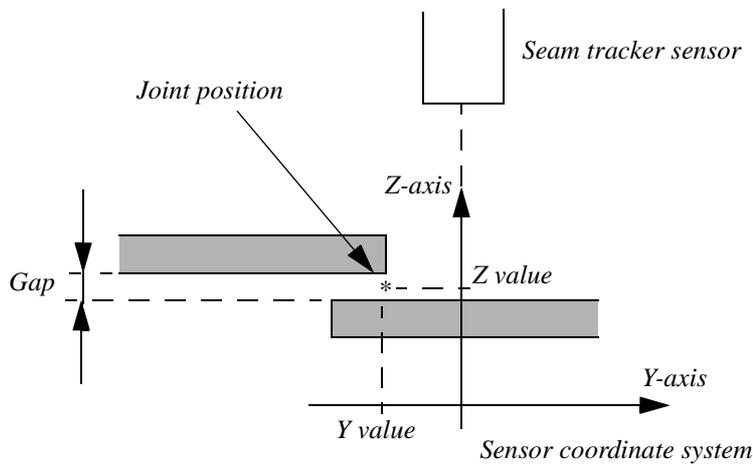


Figure 13 Sensor data for a selected joint number.

### Process blocking

- Select **Arcweld: Blocking**.

The following dialog box is displayed (see Figure 14).

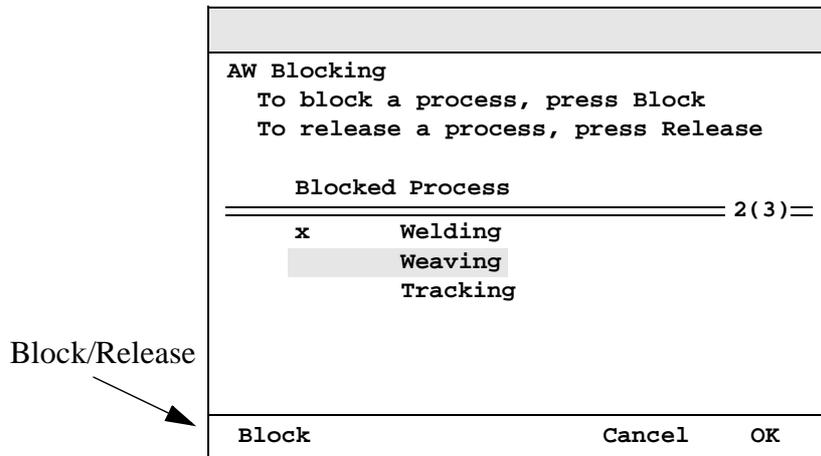


Figure 14 Dialog box for process blocking.

- Select the desired process by using the arrow keys.
- Select **Block** or **Release**.
- Choose **OK** to confirm.

Blocking can also be implemented by activating the digital process blocking inputs.

The parts of the process that have been blocked will be shown in the *Program Test* and *Program Run* windows.

Blocking that is implemented in the above dialog, is active only in the Manual operating mode. It is, however, possible to allow blocking in Auto mode if the arcwelding system parameter *auto inhib* is TRUE.

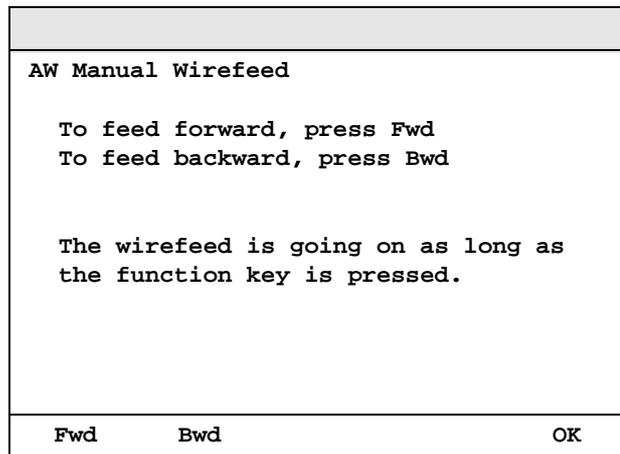
**Note** If more than one system is configured in the robot, blocking from the dialog will affect all systems. The digital process blocking inputs will only affect the corresponding system.

---

### Manual wirefeed

- Select **Arcweld: Manual wirefeed**.

The following dialog box will be displayed (see Figure 15).



AW Manual Wirefeed		
To feed forward, press Fwd To feed backward, press Bwd		
The wirefeed is going on as long as the function key is pressed.		
Fwd	Bwd	OK

Figure 15 Dialog box for manual wirefeed.

- Select **Fwd** or **Bwd**.

The wire will be fed forward or backward at 50 mm/s, as long as the function button is pressed in.

- Choose **OK** to terminate the dialog.

**Note** If more than one system is configured in the robot, the dialog for selection of AW systems can be used to select the corresponding wirefeed equipment.

---

### Manual gas on/off

- Select **Arcweld: Gas on/off**.

The following dialog box will be displayed (see Figure 16).

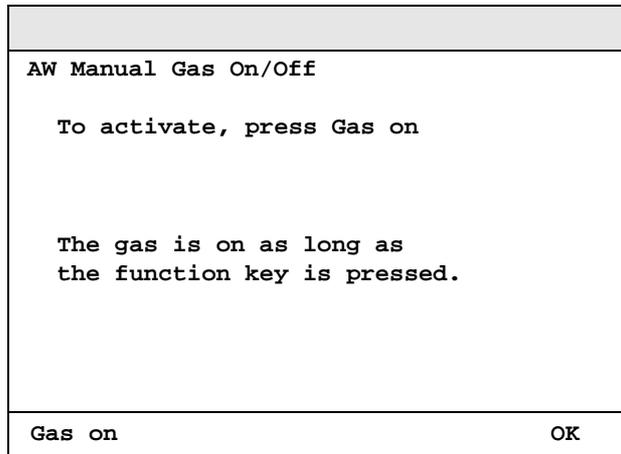


Figure 16 Dialog box for gas on/off.

- Select **Gas on**.

The gas valve is open as long as the function button is pressed in.

- Choose **OK** to terminate the dialog.

**Note** If more than one system is configured in the robot, the dialog for selection of AW systems can be used to select the corresponding gas valve.

### Select arc welding system

Up to 3 arc welding systems can exist at the same time in the robot.

- Select **Arcweld: Select AW system**.

The following dialog box is displayed (see Figure 17).

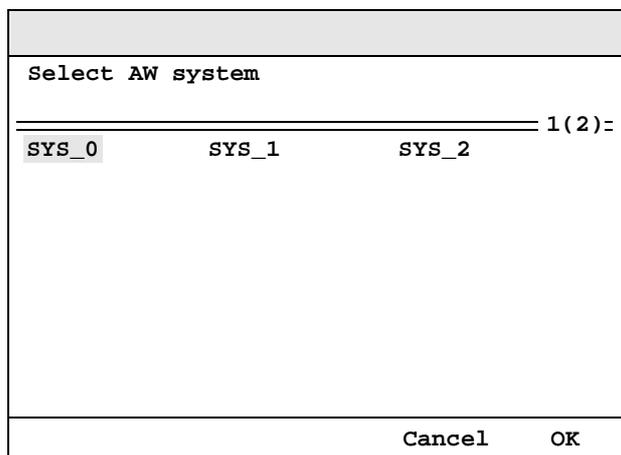


Figure 17 Dialog box for select arcwelding system.

- Select the desired system by using the arrow keys.

- Choose **OK** to confirm the decision.  
If **Cancel** is chosen, the previous arc welding system is retained as the current system. When a system has been selected as the current system, all other manual functions will operate on this system.

### Changing tuning increments

- Select **Arcweld: Increments**.

The following dialog box will be displayed (see Figure 18).

AW Tuning increments		
Welding speed:	0.10	mm/s
Wirefeed:	0.20	mm/s
Distance:	0.20	mm
Voltage:	0.50	Volts
OK		

Figure 18 Dialog box for changing tuning increments.

- Move to the desired field by using the arrow keys.
- Change the value by using the number keys.
- Choose **OK** to terminate the dialog.

### Manual functions during program execution

Arc welding functions in *Program Window*:

- Weld data tuning
- Weave data tuning
- Measured value display

Arc welding functions in *Production Window*:

- Weld data tuning
- Weave data tuning
- Measured value display

When program execution starts, the latest used function will be displayed.

**Note** This chapter refers only to the *Program Window*. The functions are, however, exactly the same in the *Production Window*.

**Data tuning**

Certain data can be tuned while it is active (i.e. when the program is executing). However, only the present values can be tuned. The original values can be tuned only when program execution has been stopped.

- Press **Start** to start the program.

A window will be displayed with a list of tunable data of the type that was latest selected (see Figure 19).

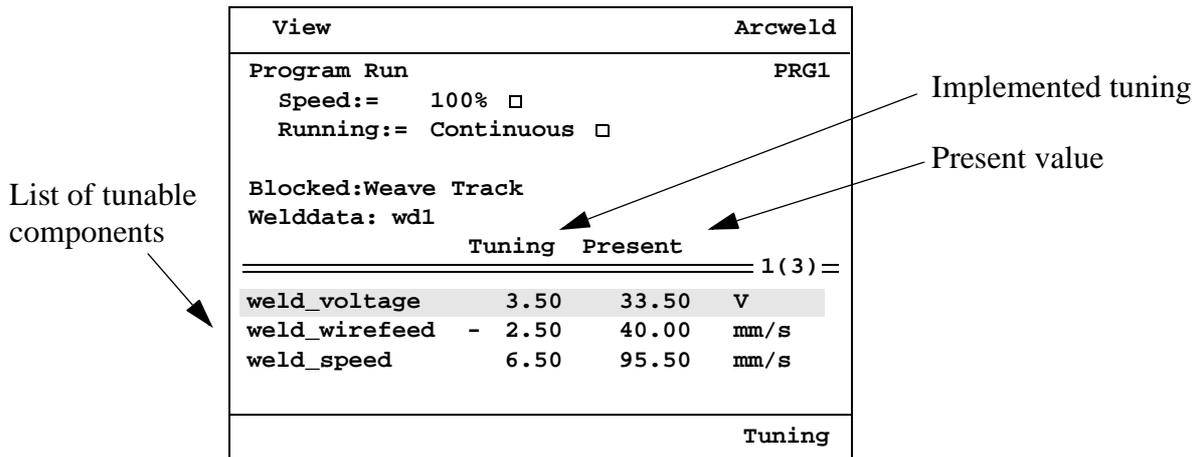


Figure 19 Window for weld data tuning during program execution.

**Selecting the data type to be tuned**

- Select **Arcweld** and enter the desired type (**Weld Tuning** or **Weave Tuning**) using the number keys.

**Note** This chapter refers only to **Weld Tuning**. However, the functions are exactly the same for **Weave Tuning**.

**Tuning the data**

- Press **Tuning**.

The following dialog box is displayed (see Figure 20).

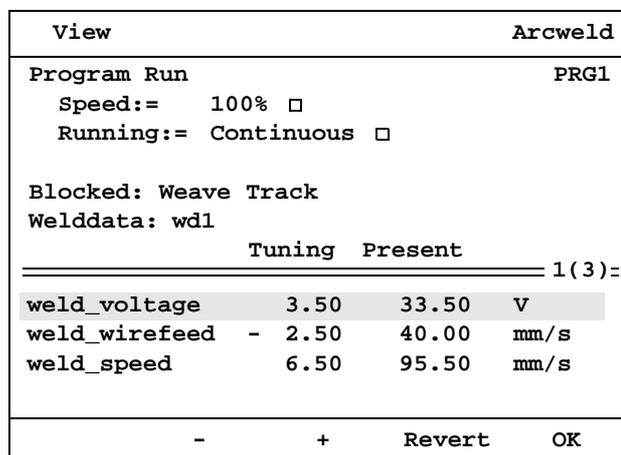


Figure 20 Window for weld data tuning during program execution.

- Select a data component for tuning and press the function keys “-” or “+” to reduce/increase the value.

### **Resetting the current tuning value**

- Press **Revert** to reset the tuning value.  
The present value is reset to the original value.

## **Measurement values**

- Select **Arcweld: Meas. values**.

The following dialog box is displayed (see Figure 21).

View		Arcweld	
Program Run		PRG1	
Speed:=	100% <input type="checkbox"/>		
Running:=	Continuous <input type="checkbox"/>		
Blocked:Weave Track			
Measurement values			
		Present	
		===== 1(2)=	
Voltage	29.70	V	
Current	320.00	A	
		-%	100%
		+%	25%

Figure 21 Window for measurement values.

The measurement values are displayed only if:

- The system is configured so it uses the analog inputs for voltage and current (the arcwelding system parameters *voltage meas* and *current meas*). In this case external measuring equipment must be used.
- The system is an ARCITEC system.

---



---

## Special ArcWare windows

---



---

### Window: Production

File	Edit	View	Arcweld
Production Info			WELDPIPE
Routine	:	main	
Status	:	Stopped	
Speed:=		100 <input type="checkbox"/> %	
Running mode:=		Continuous <input type="checkbox"/>	
Blocked:		Weave Track	
			1(26) ==
>> !Init data			
counter:=0;			
!Go to start position			
MoveL pstart,vfast,fine,weldgun1;			
WaitUntil DInput(ready)=1;			
Start	FWD	BWD	

### Menu: Arcweld

#### Arcweld

- |                       |
|-----------------------|
| 1 Manual wirefeed...  |
| 2 Gas on/off...       |
| 3 Select AW system... |
| 4 Increments...       |

#### Command

**Manual wirefeed**

**Gas on/off**

**Select AW system**

**Increments**

#### Used to:

feed the wire forwards or backwards

activate/deactivate the gas manually

choose arcwelding system

change the tuning increments

---



---

**Window: Program Test**

File	Edit	View	Test	Arcweld
Program Test		WELDPIPE/main		
Speed:	100%			
Running:=	Continuous			
Blocked:	Weave Track			
				1(26)
<b>&gt;&gt; !Init data</b>				
counter:=0;				
!Go to start position				
MoveL pstart,vfast,fine,weldgun1;				
WaitUntil DInput(ready)=1;				
Start		(Modpos)	Instr	>

**Menu: Arcweld****Arcweld**

- |                       |
|-----------------------|
| 1 Weld Tuning...      |
| 2 Weave Tuning...     |
| 3 Sensor...           |
| 4 Blocking...         |
| 5 Manual wirefeed...  |
| 6 Gas on/off...       |
| 7 Select AW system... |
| 8 Increments...       |

Command**Weld Tuning****Weave Tuning****Sensor****Blocking****Manual wirefeed****Gas on/off****Select AW system****Increments**Used to:

tune weld data

tune weave data

communicate with seamtracker sensor

block certain parts of the process

feed the wire forwards or backwards

activate/deactivate the gas manually

choose arcwelding system

change the tuning increments

**Window when executing**

Test		Arcweld	
Program Run	WELDPIPE		
Speed:	100%		
Running:=	Continuous		
Blocked:	Weave Track		
Welddata:	welddata1		
	Tuning	Present	
			1(3)
weld_speed	+ 5.5	105.5	mm/s
weld_wirefeed	- 0.15	2.30	m/min
weld_voltage	+ 3.50	33.50	Volts
			Tuning

*Menu: Arcweld*

**Arcweld**

- |   |
|---|
| <p>1 Weld Tuning...</p> <p>2 Weave Tuning...</p> <p>3 Meas. values...</p> |
|---|

Command:

**Weld Tuning**

**Weave Tuning**

**Meas. values**

Used to:

tune current weld data

tune current weave data

read measurement values

---

---

## Arc Welding

The arc welding topic contains parameters that define the arc welding functions:

- The units used when the parameters are entered
  - The process functions used
  - The equipment used
  - The weldguide sensor used
- Choose **Topics: Arc Weld**.

---

### Activating arc welding parameters

**Note** When an arc welding parameter is activated, the program memory will be erased. So, make sure that you have saved all programs on diskette.

Make sure that all input and output signals referenced by the Arc Weld configuration are defined in the topic *IO Signals*, before any parameters are activated.

Arc welding data (*seamdata*, *welddata* and *weavedata*) is always adapted to the current configuration. If, for instance, an analog output is defined to control the weld voltage, the *Voltage* component will be included in the *welddata*. Changing these parameters will thus also affect the RAPID program as a whole, which may mean that any programs created with a different set of parameters cannot be run. Activating this change requires a special procedure.

It is not always necessary to convert the AW data from one configuration to another. When opening a program, mismatching AW data in the program being opened is always pointed out.

The RAPID converter is a tool to prepare a RAPID program in the current configuration for later use, and later also to make it usable again in any configuration.

In order to be able to run old programs, they must be stored in a special format:

- Open the FileManager window.
- Select the program to be converted.
- Start conversion by using the command **Options: RAPID Converters**.
- The program will be stored in the new format with the extension XRG (instead of PRG). This XRG file is to be used in any later configuration to re-shape a PRG file, i.e. a working RAPID program.
- Change the parameters as desired.
- Perform a normal warm start by choosing **File: Restart**.
- Choose **File: Restart** in the *Service window*.
- When the confirmation dialog appears, enter the digits 2, 5 and 8.
- The text above the fifth function key will then change to **P-start**. Activate the parameters by pressing this key.

- Convert the program back to the normal format again, using the command **Options: RAPID Converters**, in the FileManager as described above.

### Defining arc welding systems

Up to 5 arc welding systems can be activated simultaneously in the same robot installation. This may be required when, for example:

- more than one piece of process equipment is connected;
- two different electrode dimensions are used (different feeding systems must be used for this to happen);
- more than one process is used, e.g. TIG and MIG/MAG.

If more than one arc welding system is defined, a new set of instructions and data types is activated for each system. The first additional system (which is defined by the sequence of defined system in a configuration file) is connected to instructions and data types with the suffix 1, and the second to the suffix 2. In other words, *ArcL1* and *ArcC1* would be connected to *seamdata1*, *welddata1*, *weavedata1*.

- Choose **Topic: Arc Weld**.
- Choose **Types: Arc Weld System**.

All defined arc welding systems will be displayed, as shown in Figure 22.

File	Edit	Topics	Types
System Parameters		Arc Weld	
Arc Weld System			
Name		1(6)	
AWSYS1			
Add			

Figure 22 All defined arc welding systems are displayed.

- Select the arc welding system to be changed and press Enter , or add a new one by pressing **Add**.
- Select the desired parameter and change its value.
- Press **OK** to confirm.

<u>Parameter</u>	<u>Description</u>
<i>name</i> <sup>1</sup>	The name of the system (max. 16 characters).
<i>units</i> <sup>1</sup>	The name of the group of measurements to be used. These groups are defined in <i>Defining measurements</i> on page 23.
<i>functions</i> <sup>1</sup>	The name of the group of functions to be used. These groups are defined in <i>Defining arc welding functions</i> on page 24.
<i>equipment</i> <sup>1</sup>	The name of the equipment to be used. This is defined in <i>Defining arc welding equipment</i> on page 27.
<i>weldguide</i>	The name of the weldguide definition to be used. This is defined in <i>Defining weldguide arc welding sensor</i> on page 30.

---

## Defining measurements

The welding speed, dimensions and the like can be specified in different units. These are grouped together under a common name and coupled to the appropriate arc welding system.

- Choose **Topic: Arc Weld**.
- Choose **Types: Arc Weld Unit**.

All defined groups of measurements (max. 5) will be displayed, as shown in Figure 23.

File	Edit	Topics	Types
System Parameters		Arc Weld	
Arc Weld Unit			
Name		1(2)	
unit_mm			
unit_inch			
Add	Delete		

Figure 23 All groups of measurements are displayed.

- Select the group of units to be changed and press Enter  , or add a new one by pressing **Add**.
- Select the desired parameter and change its value.
- Press **OK** to confirm.

---

1. These parameters must always be defined.



- Select the group of functions to be changed and press Enter , or add a new one by pressing *Add*.
- Select the desired parameter and change its value.
- Press **OK** to confirm.

<u>Parameter</u>	<u>Description</u>
<i>name</i> <sup>1</sup>	The name of the function group (max. 16 characters).
<i>restart</i>	<p>Specifies whether the weld is to be restarted in the event of a welding defect. This restart can be done in three different ways:</p> <ul style="list-style-type: none"> <li>- Automatically (the number of times specified in the parameter <i>no of retries</i>)</li> <li>- Program controlled (using the routine's error handler)</li> <li>- Manually (when the error has been remedied, the program can be started in the normal way).</li> </ul> <p>If a restart is chosen (yes), the robot automatically reverses to a position before the position where it was interrupted (as specified in the parameter <i>restart distance</i>). The weld is then restarted and program execution continues.</p>
<i>restart distance</i>	The distance that the robot must reverse (relative to the position where it was interrupted) before a restart will take place (0–150 mm).
<i>regain speed</i>	The speed at which the robot moves to the position where it was interrupted.
<i>no of retries</i>	The number of automatic restart attempts.
<i>scrape</i>	Specifies if the robot is to weave at the actual weld start (scrape start). This weaving is automatically interrupted when the arc is ignited.
<i>opt ignition scrape</i>	Specifies scrape type at weld start. This function will not affect the scrape type at restart. The scrape types are specified in <i>seamdata</i> .
<i>scrape width</i>	The width of the weave pattern for a scrape start.
<i>scrape direction</i>	The angle of direction of the weave for a scrape start. It is specified in degrees, where 0° implies a weave that is carried out at right angles to the direction of the weld.
<i>scrape cycle time</i>	The time (in seconds) it takes for a complete weave cycle for a scrape start.
<i>ignition data</i>	<p>Specifies if ignition data is to be used at the start of the weld phase. At the start it is often beneficial to define higher weld data values for a better ignition. The values to be used are specified in <i>seamdata</i>.</p> <p>If the ignition data parameter is changed, the contents of <i>seamdata</i> will also change and therefore it must be activated as in <i>Activating arc welding parameters</i> on page 21.</p>

---

1. These parameters must always be defined

<b><i>ignition move delay</i></b>	Specifies whether a move delay is to be used from the time the arc is considered stable at ignition until the heating phase is started.
<b><i>delay distance</i></b>	Specifies whether a delay distance (after the <i>destination</i> position) is to be used for a changeover to new weld data in the next arc welding instruction.
<b><i>preheating</i></b>	Specifies whether preheating is to be used at the start of the weld phase.  When the arc is ignited, the seam will generally not have reached the correct temperature. Preheating can thus be used at the start of the weld to define higher weld data values. The values to be used are specified in <i>seamdata</i> .  If the preheating parameter is changed, the contents of <i>seamdata</i> will also change and therefore it must be activated as in <i>Activating arc welding parameters</i> on page 21.
<b><i>weave</i></b>	Specifies whether a weave pattern is to be added to the basic path. In this case, the weave starts when preheating is started at the beginning of the weld phase. The way the weave is to be carried out is described in <i>weavedata</i> .  If coordinated interpolation is used, a notch filter for the external axis may have to be used. See Chapter 12, System Parameters.
<b><i>weldguide</i></b>	Specifies whether Weldguide is to be used.
<b><i>weave sync</i></b>	Specifies whether synchronisation pulses are to be transmitted at the end positions of the weave.
<b><i>track</i></b>	Specifies that a tracker is connected to the sensor interface.
<b><i>crater fill</i></b>	Specifies whether a crater fill is to be used in the final phase. This means that the end crater that can form in the completed weld will be filled in with extra filler material. Exactly how the crater fill is to be carried out is described in <i>seamdata</i> .  If the Crater fill parameter is changed, the contents of <i>seamdata</i> will also change and therefore it must be activated as in <i>Activating arc welding parameters</i> on page 21.
<b><i>burnback</i></b>	Specifies whether burnback is to be used in the final phase. It is used in MIG/MAG welding and means that the power supply switches on for a short while after the electrode feed has been turned off. The end of the weld electrode is then melted and transferred to the molten metal in the weld deposit. In this way, the electrode will separate from the molten metal and not stick to it when it starts to harden. Exactly how the burnback is to be carried out is described in <i>seamdata</i> .  If the Burnback parameter is changed, the contents of <i>seamdata</i> will also change and therefore it must be activated as in <i>Activating arc welding parameters</i> on page 21.

<b><i>burnback voltage</i></b>	Specifies whether a specific burnback voltage should be used in the burnback phase. If not specified, burnback will be performed with the voltage used in the previous welding phase. If the Burnback voltage parameter is changed, the contents of <i>seamdata</i> will also change and it must therefore be activated as in <i>Activating arc welding parameters</i> on page 21.
<b><i>rollback</i></b>	Specifies whether rollback is to be used in the final phase. It is used in TIG welding and means that the cold wire is reversed before the molten metal hardens, to prevent the wire sticking. Exactly how the rollback is to be carried out is described in <i>seamdata</i> . If the Rollback parameter is changed, the contents of <i>seamdata</i> will also change and therefore it must be activated as in <i>Activating arc welding parameters</i> on page 21.
<b><i>rollback wirefeed</i></b>	Specifies whether a specific rollback wirefeed speed should be used in the rollback phase. If not specified, a wirefeed speed of 10 mm/s will be used. If the Rollback wirefeed speed parameter is changed, the contents of <i>seamdata</i> will also change and it must therefore be activated as in <i>Activating arc welding parameters</i> on page 21.
<b><i>override</i></b>	Specifies whether override is to be used.
<b><i>precond</i></b>	Specifies whether preconditions is to be used. If precond is on, the gas supervision and water supervision signals are verified before welding is started.
<b><i>auto inhib</i></b>	Specifies whether inhibition will be allowed in AUTO-mode or not.

---

## Defining arc welding equipment

The equipment used in arc welding must be defined. In addition, the signals must be connected to the desired arc welding functions. These are grouped together under a common name and then connected to the appropriate arc welding system.

- First, define the names of the signals to be used for communication between the robot and welding equipment. See Chapter 12, System Parameters.
- Choose **Topic: Arc Weld**.
- Choose **Types: Arc Weld Equipment**.

All defined equipment (max. 5) will be displayed, as shown in Figure 25.

File	Edit	Topics	Types
System Parameters		Arc Weld	
Arc Weld Equipment			
Name		1(2)	
TIG			
MIG			
Add			

Figure 25 All defined equipment is displayed.

- Select the equipment to be changed and press Enter  , or add new equipment by pressing **Add**.
- Select the desired parameter and change its value.
- Press **OK** to confirm.

<u>Parameter</u>	<u>Description</u>
<i>name</i> <sup>1</sup>	The name of the equipment (max. 16 characters).
<i>manual wirefeed</i>	Digital input signal for manual wire feed (active <i>High</i> ).
<i>weld inhibit</i>	Digital input signal for program execution without welding (active <i>High</i> ).
<i>weave inhibit</i>	Digital input signal for program execution without weaving (active <i>High</i> ).
<i>superv inhibit</i>	Digital input signal for program execution without any supervision. (active <i>High</i> ).
<i>stop process</i>	Digital input signal for stopping program execution. This signal only affects arc welding instructions.  A high signal means that program execution will stop as soon as an arc welding instruction is executed.
<i>ign timeout</i>	The maximum time (in seconds) permitted for igniting the welding arc.
<i>arc ok delay</i>	The time it takes the welding arc to stabilise at the start of a weld. The arc is only considered ignited after the <i>arc supervision</i> signal has been high for the specified number of seconds.
<i>arc supervision</i> <sup>1</sup>	Digital input signal for supervision of the welding arc.  A high signal means that the welding arc is ignited.
<i>volt supervision</i>	Digital input signal for supervision of the voltage.  A high signal means that the voltage is OK.

---

1. These parameters must always be defined

<i>current supervision</i>	Digital input signal for supervision of the current. A high signal means that the current is OK.
<i>water supervision</i>	Digital input signal for supervision of the cooling water. A high signal means that the cooling water is OK.
<i>wire supervision</i>	Digital input signal for supervision of the wire feed. A high signal means that the wire feed is OK.
<i>gas supervision</i>	Digital input signal for supervision of the protective gas. A high signal means that the protective gas is OK.
<i>torch supervision</i>	Digital input signal for supervision of the torch. A high signal means that the torch is OK.
<i>wirestick status</i>	Digital input signal for supervision of the wire stick status. A high signal means that an error has occurred.
<i>arc weld status</i>	Digital output signal for indication of welding defects. A high signal means that an error has occurred. If a normal program stop occurs in the middle of a weld, no high signal will be generated.
<i>gas control</i>	Digital output signal for control of the gas flow (active <i>High</i> ).
<i>power control</i> <sup>1</sup>	Digital output signal for control of the weld voltage (active <i>High</i> ).
<i>arc preset time</i>	Delays the <i>power control</i> signal. This allows the analog reference signals enough time to stabilise before the weld is started.
<i>wirefeed</i>	Digital output signal for activation of the wire feed (active <i>High</i> ).
<i>wirefeed backward</i>	Digital output signal for backward activation of the wire feed (active <i>High</i> ).
<i>weldschedule port</i>	Group of digital output signals used to transfer program numbers to the welding equipment.
<i>schedule strobe</i>	Digital output signal used for handshaking if, when program numbers are transferred to the welding equipment, <i>schedule port type</i> has been defined as <i>Pulse</i> (active <i>High</i> ).
<i>schedule port type</i>	Type of port used to transfer program numbers to the welding equipment: <i>Binary</i> = Binary-coded group of digital output signals <i>Pulse</i> = Program numbers are sent in the form of a number of pulses on the Weldschedule port signal which should then comprise two digital signals. They are pulsed in tens on one of the outputs and in ones on the other. <i>CAN</i> = Weldschedule number written on the CAN bus (dedicated for the Arcitec system).
<i>weld voltage</i>	Analog output signal for analog voltage reference.

---

1. These parameters must always be defined

<i>voltage adjust</i>	Analog output signal for relative adjustment of the analog voltage reference.
<i>wire feed</i>	Analog output signal for analog wire feed reference.
<i>current</i>	Analog output signal for analog current reference.
<i>current adjust</i>	Analog output signal for relative adjustment of the analog current reference.
<i>voltage meas</i>	Analog input signal for voltage measurement.
<i>current meas</i>	Analog input signal for current measurement.
<i>process stopped</i>	Digital output signal used to indicate that the weld has been interrupted.  A high signal means that the weld has been interrupted either because of a welding defect or because of a normal program stop.
<i>supervision arc</i>	Digital output signal for indication of welding arc errors.  A high signal means that an error has occurred.
<i>supervision voltage</i>	Digital output signal for indication of voltage errors.  A high signal means that an error has occurred.
<i>supervision current</i>	Digital output signal for indication of current errors.  A high signal means that an error has occurred.
<i>supervision water</i>	Digital output signal for indication of cooling water errors.  A high signal means that an error has occurred.
<i>supervision gas</i>	Digital output signal for indication of protective gas errors.  A high signal means that an error has occurred.
<i>supervision torch</i>	Digital output signal for indication of torch errors.  A high signal means that an error has occurred.
<i>supervision wirefeed</i>	Digital output signal for indication of wire feed errors.  A high signal means that an error has occurred.

---

## Defining weldguide arc welding sensor

The weldguide used in arc welding must be defined. In addition, the signals must be coupled to the desired arc welding functions. These are grouped together under a common name and then coupled to the appropriate arc welding system.

- First, define the names of the signals to be used for communication between the robot and Weldguide. See Chapter 12, System Parameters.
- Define in which functions weldguide is to be used (parameter weldguide, weave sync). See *Defining arc welding functions* on page 24.
- Choose **Topic: Arc Weld**.
- Choose **Types: Arc Weld Weldguide**.

All defined weldguides (max. 5) will be displayed, as shown in Figure 26.

File	Edit	Topics	Types
System Parameters			Arc Weld
Arc Weld Weldguide			
Name			1(2)
AWWG1			
AWWG2			
Add			

Figure 26 All defined weldguides are displayed.

- Select the weldguide to be changed and press Enter  , or add new weldguide by pressing **Add**.
- Select the desired parameter and change its value.
- Press **OK** to confirm.

<u>Parameter</u>	<u>Description</u>
<b><i>name</i></b> <sup>1</sup>	The name of the weldguide sensor (max. 16 characters).
<b><i>wg ready</i></b>	Digital input signal indicating that the weldguide tracker is ready (active High).
<b><i>wg inhib</i></b>	Digital output signal for stopping the weldguide tracker (active High).
<b><i>wg track</i></b>	Digital output signal for starting the weldguide tracker (active High).
<b><i>wg left sync</i></b>	Digital output signal indicating that the welding torch is at the left hand joint side of the weave pattern (active High).
<b><i>wg right sync</i></b>	Digital output signal indicating that the welding torch is at the right hand joint side of the weave pattern (active High).
<b><i>wg height corr</i></b>	Group of digital input signals for height corrections.
<b><i>wg side corr</i></b>	Group of digital input signals for side corrections.
<b><i>wg data valid</i></b>	Digital input signal indicating that weldguide is giving height and side corrections (active High).
<b><i>wg data ack</i></b>	Digital output signals for acknowledgement of data valid. The acknowledgement output pulse must be least 10 ms wide (active High).

---

1. These parameters must always be defined.



---



---

## ArcL                      Arc welding with linear motion

### ArcL1

### ArcL2

*ArcL* (*Arc Linear*) is used to weld along a straight seam. The instruction controls and monitors the entire welding process as follows:

- The tool centre point is moved linearly to the specified destination position.
- All phases of the welding process, such as the start and end phases, are controlled.
- The welding process is monitored continuously.

The only difference between *ArcL*, *ArcL1* and *ArcL2* is that they are connected to different process systems configured in the System Parameters. Although *ArcL* is used in the examples, *ArcL1* or *ArcL2* could equally well be used.

---

### Example

```
MoveJ . . . .
ArcL  \On, p1, v100, seam1, weld5, noweave, fine, gun1;
ArcL  \Off, p2, v100, seam1, weld5, noweave, fine, gun1;
MoveJ . . . .
```

This welds a straight seam between points *p1* and *p2*, as illustrated in Figure 27.

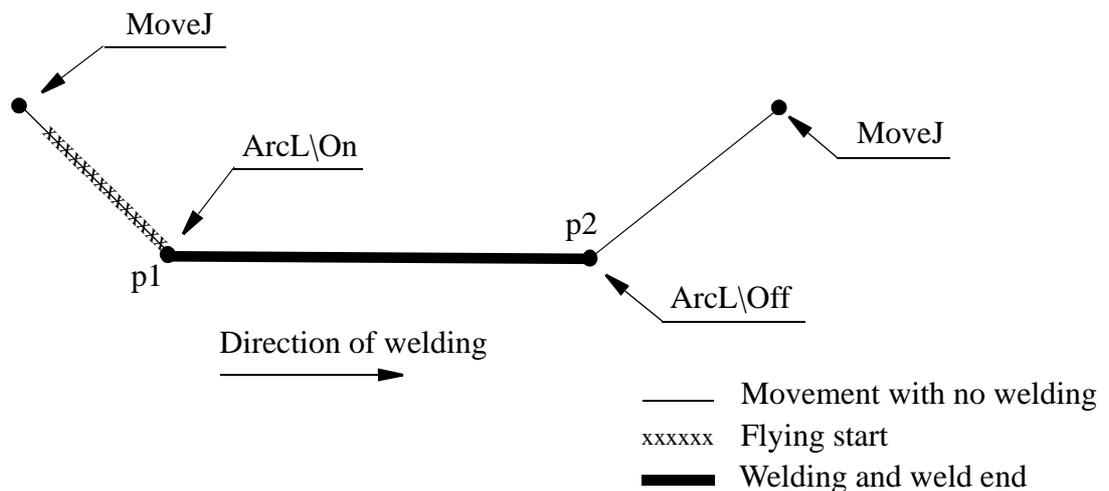


Figure 27 Welding with flying start.

On the way to *p1*, preparations for the weld start, such as gas preflowing, are carried out. The process and the actual weld movement then start at position *p1* and end at *p2*. The start and end processes are determined by *seam1* and the welding process by *weld5*. Weaving data is carried out according to *noweave*. (No weaving if the *weave\_shape* component value is zero.)

*V100* specifies the speed attained during the flying start to *p1*.

---

**Arguments**

**ArcL** [**\On**] | [**\Off**] **ToPoint** **Speed** [**\T**] **Seam** **Weld** **Weave** **Zone**  
 [**\Z**] **Tool** [**\WObj**] [**\SeamName**]

**[\On]**

Data type: *switch*

The argument *\On* is used to obtain a *flying start* (see Figure 27) which, in turn, results in shorter cycle times.

The argument *\On* may only be used in the first of the arc welding instructions to result in a seam. As the end instructions cannot include the argument *\On*, welding with a flying start must include at least two instructions.

The start preparations at a flying start, e.g. gas purging, are carried out on the way to the weld start position.

When the argument *\On* is not used, the weld starts at the position before the *ArcL* instruction (see Figure 28) and the robot remains stationary at the previous position whilst *all* weld start activities are in progress.

Whether or not a flying start is used, the start position for the weld will always be a stop point – regardless of what is specified in the *Zone* argument for that position.

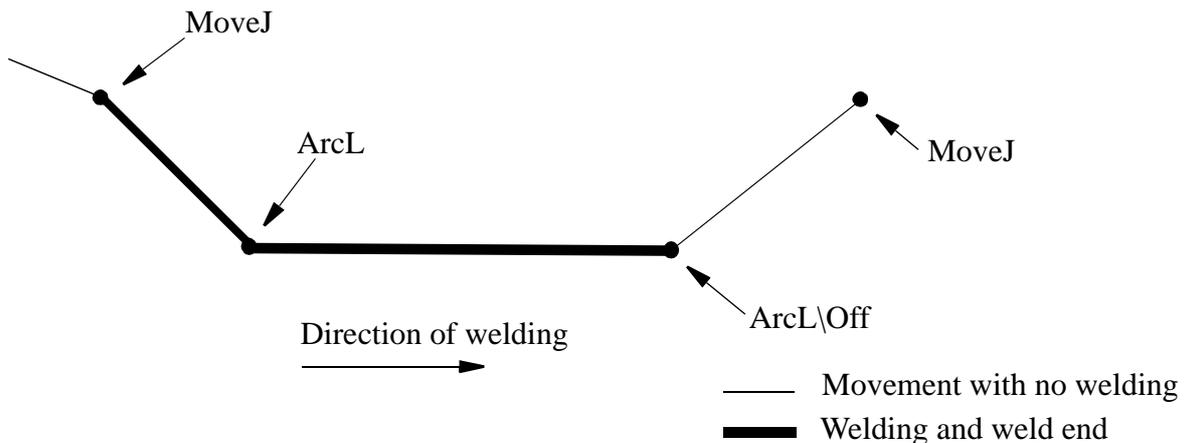


Figure 28 If welding is started without the argument *\On*, the weld is begun at the previous position.

**[\Off]**

Data type: *switch*

If the argument *\Off* is used, welding ends when the robot reaches the destination position. Regardless of what is specified in the *Zone* argument, the destination position will be a stop point.

If an *ArcL* instruction without the argument *\Off* is followed by *MoveJ*, for example, welding will end, but in an uncontrolled fashion. Logical instructions, such as *Set doI*, however, can be used between two arc welding instructions without ending the welding process.

**ToPoint**Data type: *robtarget*

The destination position of the robot and external axes. This is either defined as a named position or stored directly in the instruction (indicated by an \* in the instruction).

**Speed**Data type: *speeddata*

The speed of the TCP is controlled by the argument *Speed* in the following cases:

- When the argument *\On* is used (weld start preparations at a flying start).
- When the program is run instruction-by-instruction (no welding).

The speed of the TCP *during welding* is the same as for the arguments *Seam* and *Weld*. (See Figure 29)

Speed data also describes the speed of the tool's reorientation and the speed of any uncoordinated external axes.

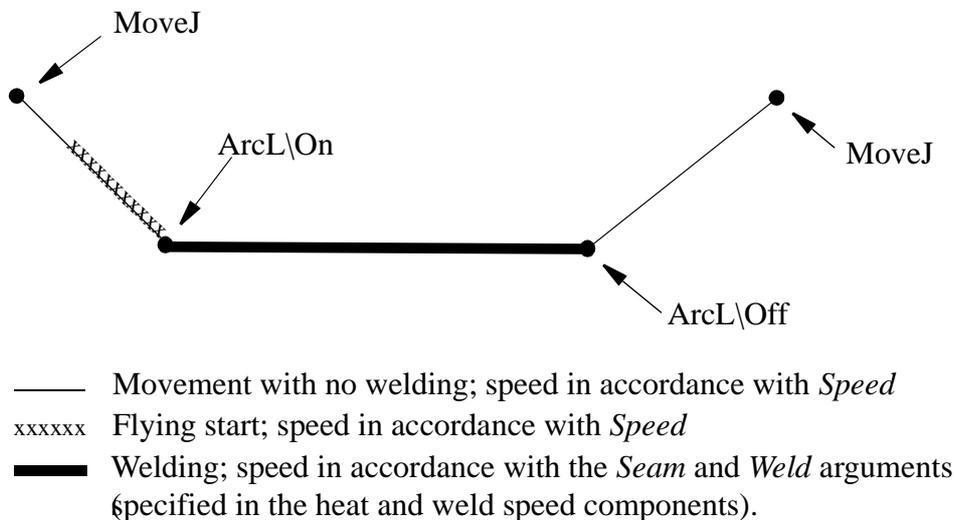


Figure 29 The speed at different phases of the welding process.

**[ \T ]**

(Time)

Data type: *num*

The argument *\T* is used to specify the total time of movement in seconds directly in the instruction. Time is thus substituted for the speed specified in the arguments *Speed*, *Seam* and *Weld*.

This argument can be used when, for example, one or more uncoordinated external axes participate in the movement. Uncoordinated external axes should, however, be avoided since, if used, the program becomes more difficult to adjust. Use coordinated external axes instead. Weaving is deactivated during execution of ArcX instructions with *\T* arguments.

**Seam** Data type: *seamdata*

Seam data describes the start and end phases of a welding process.

The argument *Seam* is included in all arc welding instructions so that, regardless of the position of the robot when the process is interrupted, a proper weld end and restart is achieved.

Normally the same seam data is used in all instructions of a seam.

**Weld** Data type: *welldata*

Weld data describes the weld phase of the welding process.

Weld data is often changed from one instruction to the next along a seam.

**Weave** Data type: *weavedata*

Weave data describes the weaving that is to take place during the heat and weld phases. Welding without weaving is obtained by specifying, for example, the weave data *noweave*. (No weaving if the *weave\_shape* component value is zero.)

**Zone** Data type: *zonedata*

Zone data defines how close the axes must be to the programmed position before they can start moving towards the next position.

In the case of a fly-by point, a corner path is generated past that position. In the case of a stop point (*fine*), the movement is interrupted until all axes have reached the programmed point.

A stop point is always generated automatically at the start position of a weld (even in the case of a *flying start*) and at a *controlled* weld end position. Fly-by points, such as *z10*, should be used for all other weld positions.

Weld data changes over to the next arc welding instruction at the centre point of the corner path (if not delayed by the *delay\_distance* component in the *Weld* argument).

[ \Z ] (Zone) Data type: *num*

This argument is used to specify the positional accuracy of the robot's TCP directly in the instruction. The size of the zone is specified in mm and is thus substituted in the corresponding zone specified in the zone data. The \Z argument is also useful when trimming individual corner paths.

**Tool** Data type: *tooldata*

The tool used in the movement. The TCP of the tool is the point moved to the specified destination position. The z-axis of the tool should be parallel with the torch.



---

## Limitations

When weaving, the distance between the programmed positions should be longer than the periodic time of weaving. If the distance is shorter and if there is a significant change of angle in the path, the weaving pattern will be distorted.

---

## Fault management

The process is supervised by a number of signal inputs. If anything abnormal is detected, program execution will stop. If, however, an error handler is programmed, the errors defined below can be remedied without stopping production. See the example in the *RestoPath* instruction.

<u>Error constant (ERRNO value)</u>	<u>Description</u>
AW_START_ERR	Start condition error; torch, gas or water supervision
AW_IGNI_ERR	Ignition error; arc supervision
AW_WELD_ERR	Weld error; arc supervision
AW_EQIP_ERR	Weld equipment error; voltage, current, water or gas supervision during welding
AW_WIRE_ERR	Wire stick error; wire stick supervision
AW_STOP_ERR	Welding interrupted using the stop process input

The process supervision is determined as a part of the process equipment configuration.

At the *start* of the process the robot checks that the following *preconditions* have been met:

- stop\_process
- water\_OK
- gas\_OK
- torch\_OK

If, after the start command is given, no approved start profile is indicated on the digital input, *arc\_OK*, within a predetermined time period, the process start will be interrupted.

When the process is started, all supervision inputs selected are monitored continuously:

- stop\_process, water\_OK, gas\_OK, arc\_OK, volt\_OK, curr\_OK, feed\_OK

The wirestick\_err supervision is checked at the end of the weld.

---

**Example**

```

MoveL ...
ArcL  \On, *, v100, seam1, weld5, weave1, fine, gun1\Wobj:=wobj1;
ArcL  *, v100, seam1,weld5, weave1, z10, gun1\Wobj:=wobj1;
ArcL  *, v100, seam1,weld5, weave1, z10, gun1\Wobj:=wobj1;
ArcL  \Off, *, v100, seam1,weld3, weave3, fine, gun1\Wobj:=wobj1;
MoveL ...

```

In this example, a weld is performed in which weld data and weave data are changed in the final part of the weld, which is illustrated in Figure 30. Note that an arc welding instruction must be used to change the direction of the path despite the fact that no weld data is changed.

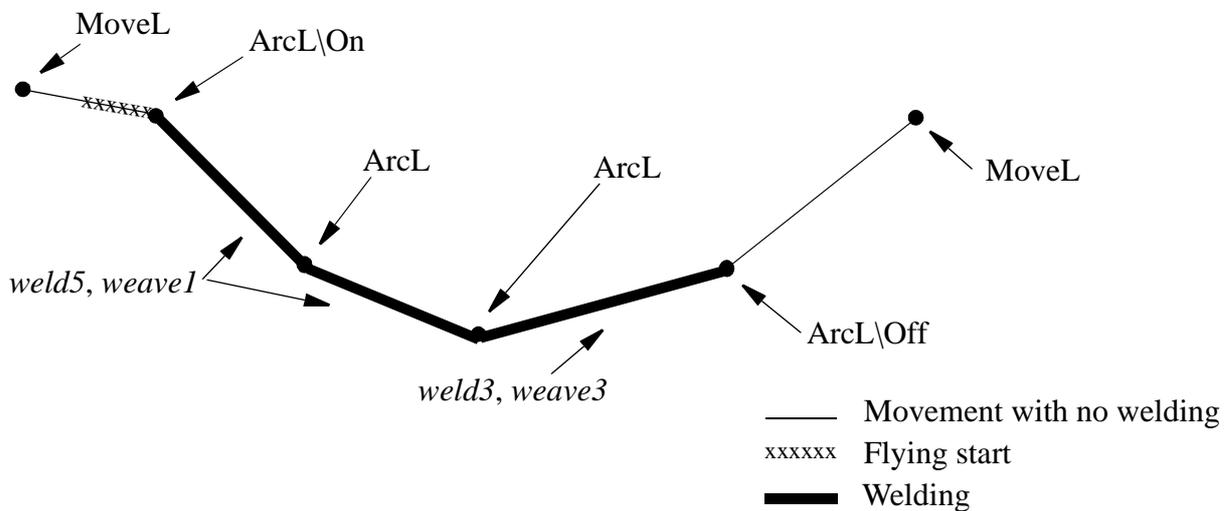


Figure 30 The direction and weld data can be changed by programming several ArcL instructions.

It is assumed, in this example, that a coordinated external axis is used in the movement. In this case, the *wobj1* work object must be specified in the instruction.

---

## Syntax

ArcL

```
[ '\On', ' ] | [ '\Off', ' ]
[ ToPoint ':=' ] < expression (IN) of robtarg > ', '
[ Speed ':=' ] < expression (IN) of speeddata >
  [ ( '\ T ':=' < expression (IN) of num > ) ] ', '
[ Seam ':=' ] < persistent (PERS) of seamdata > ', '
[ Weld ':=' ] < persistent (PERS) of welldata > ', '
[ Weave ':=' ] < persistent (PERS) of weavedata > ', '
[ Zone ':=' ] < expression (IN) of zonedata >
  [ '\ Z ':=' < expression (IN) of num > ] ', '
[ Tool ':=' ] < persistent (PERS) of tooldata >
[ '\ SeamName ':=' < expression (IN) of string > ]
[ '\ WObj ':=' < persistent (PERS) of wobjdata > ', ';
```

---

## Related information

	<u>Described in:</u>
Performing a circular weld	Instructions - <i>ArcC</i>
Other positioning instructions	RAPID Summary - <i>Motion</i>
Definition of speed	Data Types - <i>speeddata</i>
Definition of zone data	Data Types - <i>zonedata</i>
Definition of tools	Data Types - <i>tooldata</i>
Definition of work objects	Data Types - <i>wobjdata</i>
Definition of seam data	Data Types - <i>seamdata</i>
Definition of weld data	Data Types - <i>welldata</i>
Definition of weave data	Data Types - <i>weavedata</i>
Installation parameters for welding equipment and welding functions	System Parameters - <i>Arc Welding</i>
Movements in general	Motion Principles
Coordinate systems	Motion Principles - <i>Coordinate Systems</i>
Process phases and sub-activities	RAPID Summary - <i>Arc Welding</i>

---



---

## ArcC      Arc welding with circular motion

### ArcC1

### ArcC2

*ArcC* (*Arc Circular*) is used to weld along a circular path. The instruction controls and monitors the entire welding process as follows:

- The tool centre point is moved in a circle to the specified destination position.
- All phases, such as the start and end phases, of the welding process are controlled.
- The welding process is monitored continuously.

The only difference between *ArcC*, *ArcC1* and *ArcC2* is that they are connected to different process systems configured in the System Parameters. Although *ArcC* is used in the examples, *ArcC1* or *ArcC2* could equally well be used.

---

### Example

```
MoveL . . . .
ArcL \On, p1, v100, seam1, weld5, noweave, fine, gun1;
ArcC \Off, p2, p3, v100, seam1, weld5, noweave, fine, gun1;
MoveL . . . .
```

This welds a circular seam between points *p1* and *p3* (via point *p2*) as illustrated in Figure 31.

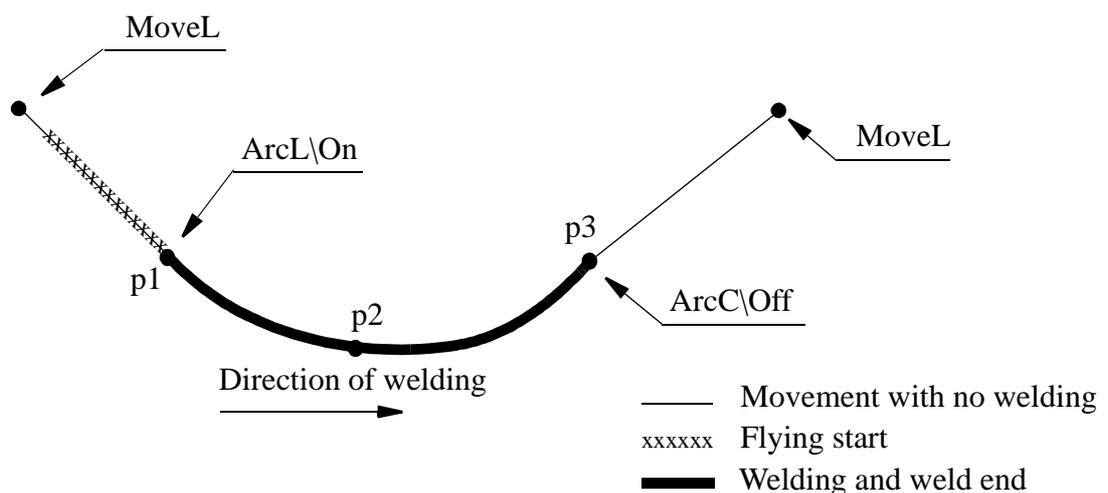


Figure 31 Welding with flying start.

On the way to *p1*, preparations for the weld start, such as gas preflowing, are carried out. The process and the actual weld movement then start at position *p1* and end at *p3*. The start and end processes are determined by *seam1* and the welding process by *weld5*. Weaving data is carried out according to *noweave*. (No weaving if the *weave\_shape* component value is zero.)

*V100* specifies the speed attained during the flying start to *p1*.

---

**Arguments**

**ArcC** [**\On**] | [**\Off**] **CirPoint ToPoint Speed** [**\T**] **Seam Weld Weave Zone** [**\Z**] **Tool** [**\WObj**] [**\SeamName**]

**[\On]**

Data type: *switch*

The argument `\On` is used to get a *flying start* (see Figure 31) which, in turn, results in shorter cycle times.

The argument `\On` may only be used in the first of the arc welding instructions to result in a seam. As the end instructions cannot include the argument `\On`, welding with a flying start must include at least two instructions.

The start preparations at a flying start, e.g. gas purging, are carried out on the way to the weld start position.

When the argument `\On` is not used, the weld starts at the position before the *ArcC* instruction (see Figure 32) and the robot remains stationary at the previous position whilst *all* weld start activities are in progress.

Whether or not a flying start is used, the start position for the weld will always be a stop point – regardless of what is specified in the *Zone* argument for that position.

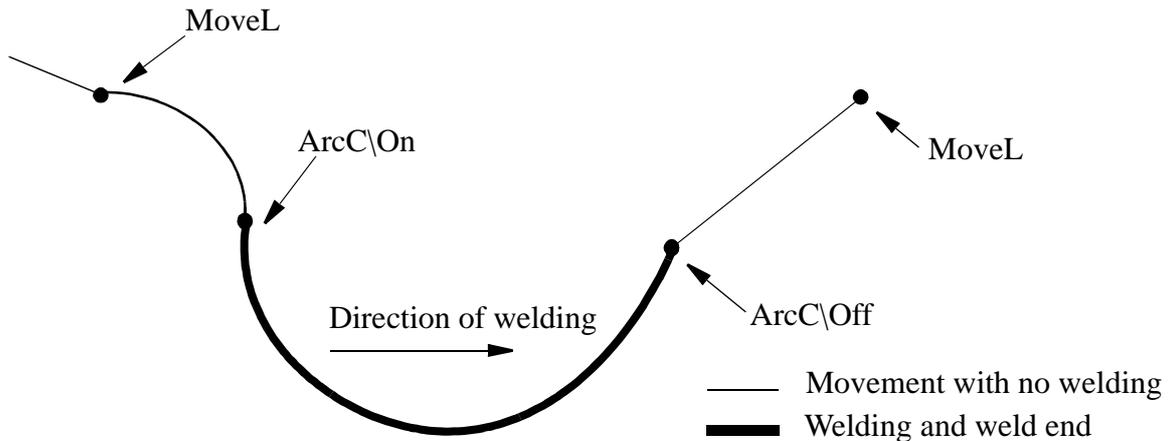


Figure 32 If welding is started without the argument `\On`, the weld is begun at the previous position.

**[\Off]**

Data type: *switch*

If the argument `\Off` is used, welding ends when the robot reaches the destination position. Regardless of what is specified in the *Zone* argument, the destination position will be a stop point.

If an *ArcC* instruction without the argument `\Off` is followed by *MoveJ*, for example, welding will end, but in an uncontrolled fashion. Logical instructions, such as *Set doI*, however, can be used between two arc welding instructions without ending the welding process.

**CirPoint**Data type: *robtarget*

The circle point of the robot. The circle point is a position on the circle between the start point and the destination point. To obtain the best accuracy, it should be placed about halfway between the start and destination points. If it is placed too close to the start or destination point, the robot may give a warning. The circle point is defined as a named position or stored directly in the instruction (marked with an \* in the instruction).

**ToPoint**Data type: *robtarget*

The destination position of the robot and external axes. This is either defined as a named position or stored directly in the instruction (indicated by an \* in the instruction).

**Speed**Data type: *speeddata*

The speed of the TCP is controlled by the argument *Speed* in the following cases:

- When the argument *\On* is used (weld start preparations at a flying start).
- When the program is run instruction-by-instruction (no welding).

The speed of the TCP *during welding* is the same as for the arguments *Seam* and *Weld* (see Figure 33).

Speed data also describes the speed of the tool's reorientation and the speed of any uncoordinated external axes.

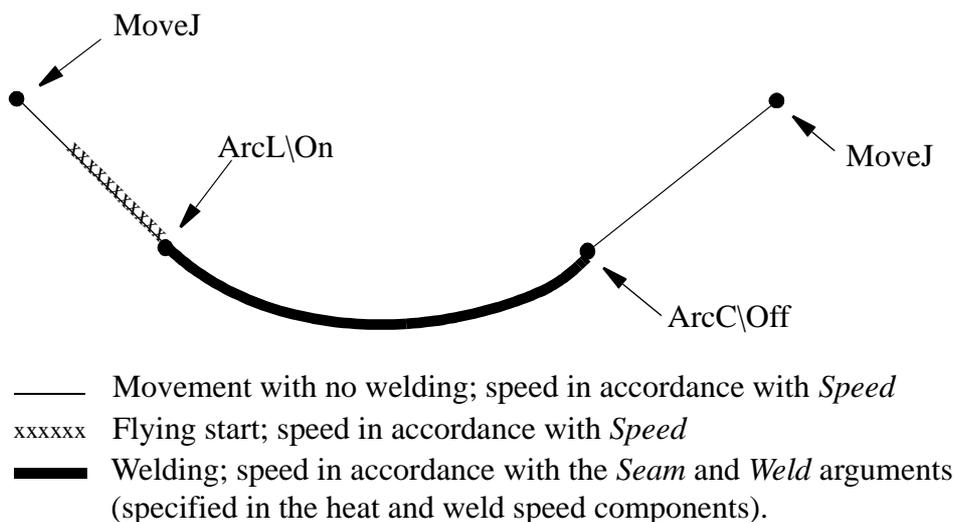


Figure 33 The speed at different phases of the welding process.

**[ \T ]**

(Time)

Data type: *num*

The argument *\T* is used to specify the total time of movement in seconds, directly in the instruction. Time is thus substituted for the speed specified in the arguments *Speed*, *Seam* and *Weld*.

This argument can be used when, for example, one or more uncoordinated external axes participate in the movement. Uncoordinated external axes should be avoided because when they are used, the program becomes more difficult to adjust. Use coordinated external axes instead. Weaving is deactivated during execution of ArcX instructions with \T arguments.

**Seam** Data type: *seamdata*

Seam data describes the start and end phases of a welding process.

The argument *Seam* is included in all arc welding instructions so that, regardless of the position of the robot when the process is interrupted, a proper weld end and restart are achieved.

Normally the same seam data is used in all instructions of a seam.

**Weld** Data type: *welddata*

Weld data describes the weld phase of the welding process.

Weld data is often changed from one instruction to the next, along a seam.

**Weave** Data type: *weavedata*

Weave data describes the weaving that is to take place during the heat and weld phases. Welding without weaving is obtained by specifying, for example, the weave data *noweave*. (No weaving if the *weave\_shape* component value is zero.)

**Zone** Data type: *zonedata*

Zone data defines how close the axes must be to the programmed position before they can start moving towards the next position.

In the case of a fly-by point, a corner path is generated past that position. In the case of a stop point (*fine*), the movement is interrupted until all axes have reached the programmed point.

A stop point is always generated automatically at the start position of a weld (even in the case of a *flying start*) and at a *controlled* weld end position. Fly-by points, such as *z10*, should be used for all other weld positions.

Weld data changes over to the next arc welding instruction at the centre point of the corner path (if not delayed by the *delay\_distance* component in the *Weld* argument).

[ \Z ] (Zone) Data type: *num*

This argument is used to specify the positional accuracy of the robot's TCP directly in the instruction. The size of the zone is specified in mm and is thus substituted in the corresponding zone specified in the zone data. The \Z argument is also useful when trimming individual corner paths.

**Tool**Data type: *tooldata*

The tool used in the movement. The TCP of the tool is the point moved to the specified destination position. The z-axis of the tool should be parallel with the torch.

**[ \WObj ]***(Work Object)*Data type: *wobjdata*

The work object (coordinate system) to which the instruction's robot position is referenced.

When this argument is omitted, the robot position is referenced to the world coordinate system. It must, however, be specified if a stationary TCP or coordinated external axes are used.

*\WObj* can be used if a coordinate system is defined for either the object in question or the weld seam.

**[ \SeamName ]***(Seam Name)*Data type: *string*

The seam name is a string which will be added to error logs if an error occurs during the welding sequence.

*\SeamName* is only applicable in the first instruction of a sequence of weld instructions i.e together with the *\On argument*.

**Program execution****Controlling process equipment**

The process equipment is controlled by the robot in such a way that the entire process and each of its phases are coordinated with the robot's movements.

**Motion**

Robot and external axes are moved to the destination position as follows:

- The TCP of the tool is moved circularly at a constant programmed speed. When coordinated axes are used, they are moved circularly at constant programmed speed relative to the work object.
- The tool is reorientated at even intervals throughout the entire course.
- Uncoordinated external axes are executed at a constant speed which means that they reach their destination at the same time as the robot axes.

If the programmed speed of reorientation or of the external axes is exceeded, these speeds will be limited, thereby reducing the speed of the TCP.

The destination position is referenced to:

- the specified object coordinate system if the argument `\WObj` is used;
- the world coordinate system if the argument `\WObj` is not used.

## Limitations

When weaving, the distance between the programmed positions should be longer than the periodic time of weaving. If the distance is shorter and if there is a significant change of angle in the path, the weaving pattern will be distorted.

The instruction *ArcC* should never be restarted after the circle point has been passed. Otherwise the robot will not take the programmed path (positioning around the circular path in another direction compared with that programmed).

## Error management

The process is supervised by a number of signal inputs. If anything abnormal is detected, program execution will stop. If, however, an error handler is programmed, the errors defined below can be remedied without stopping production. See the example in the *RestoPath* instruction.

<u>Error constant (ERRNO value)</u>	<u>Description</u>
AW_START_ERR	Start condition error; torch, gas or water supervision
AW_IGNI_ERR	Ignition error; arc supervision
AW_WELD_ERR	Weld error; arc supervision
AW_EQIP_ERR	Weld equipment error; voltage, current, water or gas supervision during welding
AW_WIRE_ERR	Wire stick error; wire stick supervision
AW_STOP_ERR	Welding interrupted with the stop process input

The process supervision is determined as a part of the process equipment configuration.

At the *start* of the process the robot checks that the following *preconditions* have been met:

- stop\_process
- water\_OK
- gas\_OK
- torch\_OK

If, after the start command is given, no approved start profile is indicated on the digital input, *arc\_OK*, within a predetermined time period, the process start will be interrupted.

When the process is started, all supervision inputs selected are monitored continuously:

- stop\_process, water\_OK, gas\_OK, arc\_OK, volt\_OK, curr\_OK, feed\_OK.

The wirestick\_err supervision is checked at the end of the weld.

## Example

```

MoveL ...
ArcL \On, *, v100, seam1, weld5, weave1, fine, gun1\Wobj:=wobj1;
ArcC *, *, v100, seam1, weld5, weave1, z10, gun1\Wobj:=wobj1;
ArcL *, v100, seam1, weld5, weave1, z10, gun1\Wobj:=wobj1;
ArcC \Off, *, *, v100, seam1, weld3, weave3, fine, gun1\Wobj:=wobj1;
MoveL...

```

In this example, a weld is performed in which weld data and weave data are changed in the final part of the weld, which is illustrated in Figure 34. Note that an arc welding instruction must be used to change the direction of the path despite the fact that no weld data is changed.

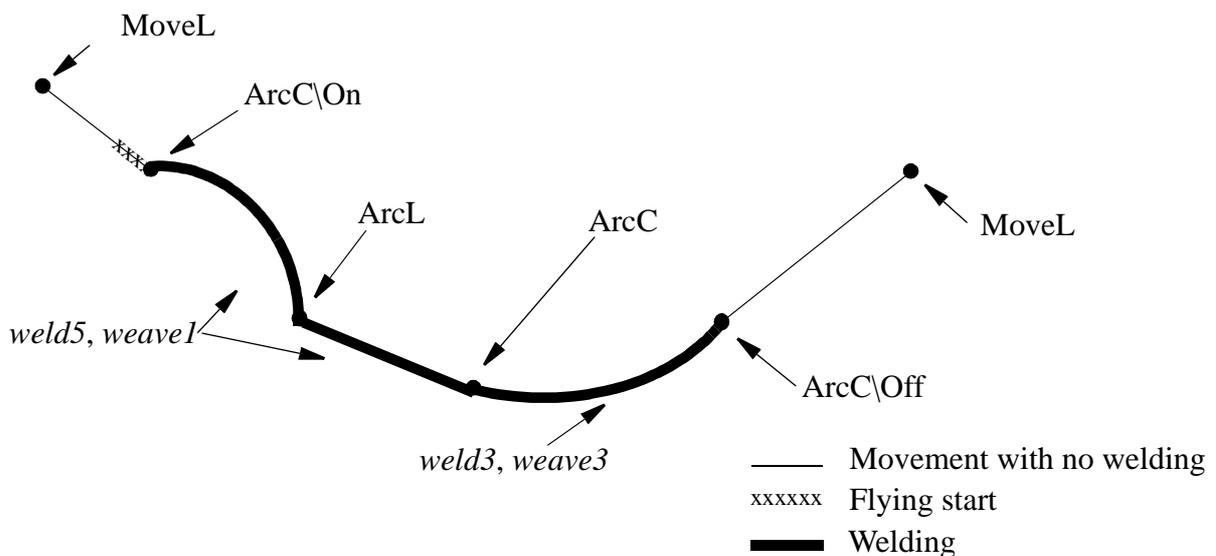


Figure 34 The direction and weld data can be changed by programming several arc welding instructions.

It is assumed, in this example, that a coordinated external axis is used in the movement. In this case, the *wobj1* work object must be specified in the instruction.

---

## Syntax

ArcC

```
[ '\On', ' ] | [ '\Off', ' ]
[ CirPoint :=' ] < expression (IN) of robtarg > ', '
[ ToPoint :=' ] < expression (IN) of robtarg > ', '
[ Speed :=' ] < expression (IN) of speeddata >
  [ ( '\ T :=' < expression (IN) of num > ) ] ', '
[ Seam :=' ] < persistent (PERS) of seamdata > ', '
[ Weld :=' ] < persistent (PERS) of welldata > ', '
[ Weave :=' ] < persistent (PERS) of weavedata > ', '
[ Zone :=' ] < expression (IN) of zonedata >
  [ '\ Z :=' < expression (IN) of num > ] ', '
[ Tool :=' ] < persistent (PERS) of tooldata >
[ '\ SeamName :=' < expression (IN) of string > ]
[ '\ WObj :=' < persistent (PERS) of wobjdata > ] ', '
```

---

## Related information

	<u>Described in:</u>
Performing a linear weld	Instructions - <i>ArcL</i>
Other positioning instructions	RAPID Summary - <i>Motion</i>
Definition of speed	Data Types - <i>speeddata</i>
Definition of zone data	Data Types - <i>zonedata</i>
Definition of tools	Data Types - <i>tooldata</i>
Definition of work objects	Data Types - <i>wobjdata</i>
Definition of seam data	Data Types - <i>seamdata</i>
Definition of weld data	Data Types - <i>welldata</i>
Definition of weave data	Data Types - <i>weavedata</i>
Installation parameters for welding equipment and welding functions	System Parameters - <i>Arc Welding</i>
Movements in general	Motion Principles
Coordinate systems	Motion Principles - <i>Coordinate Systems</i>
Process phases and sub-activities	RAPID Summary - <i>Arc Welding</i>

---

---

**seamdata****Seam data**

*Seamdata* is used to control the start and end of the weld. *Seamdata* is also used if the process is restarted after a welding operation has been interrupted.

The actual weld phase is controlled using *welddata*.

---

**Description**

Seam data describes data, the values of which, as a rule, can be maintained unaltered when welding a complete run and often also when welding several seams. Seam data is used when preparing for the welding operation, when igniting the arc, when heating after the ignition and also when ending the weld.

Seam data is included in all arc welding instructions to facilitate controlled end and start phases irrespective of where the interrupts or restarts occur.

**Note.** Some of the components of seam data depend on the configuration of the robot. If a given feature is omitted, the corresponding component is left out of the seam data. The conditions that must be met for components to exist are described in the system parameters.

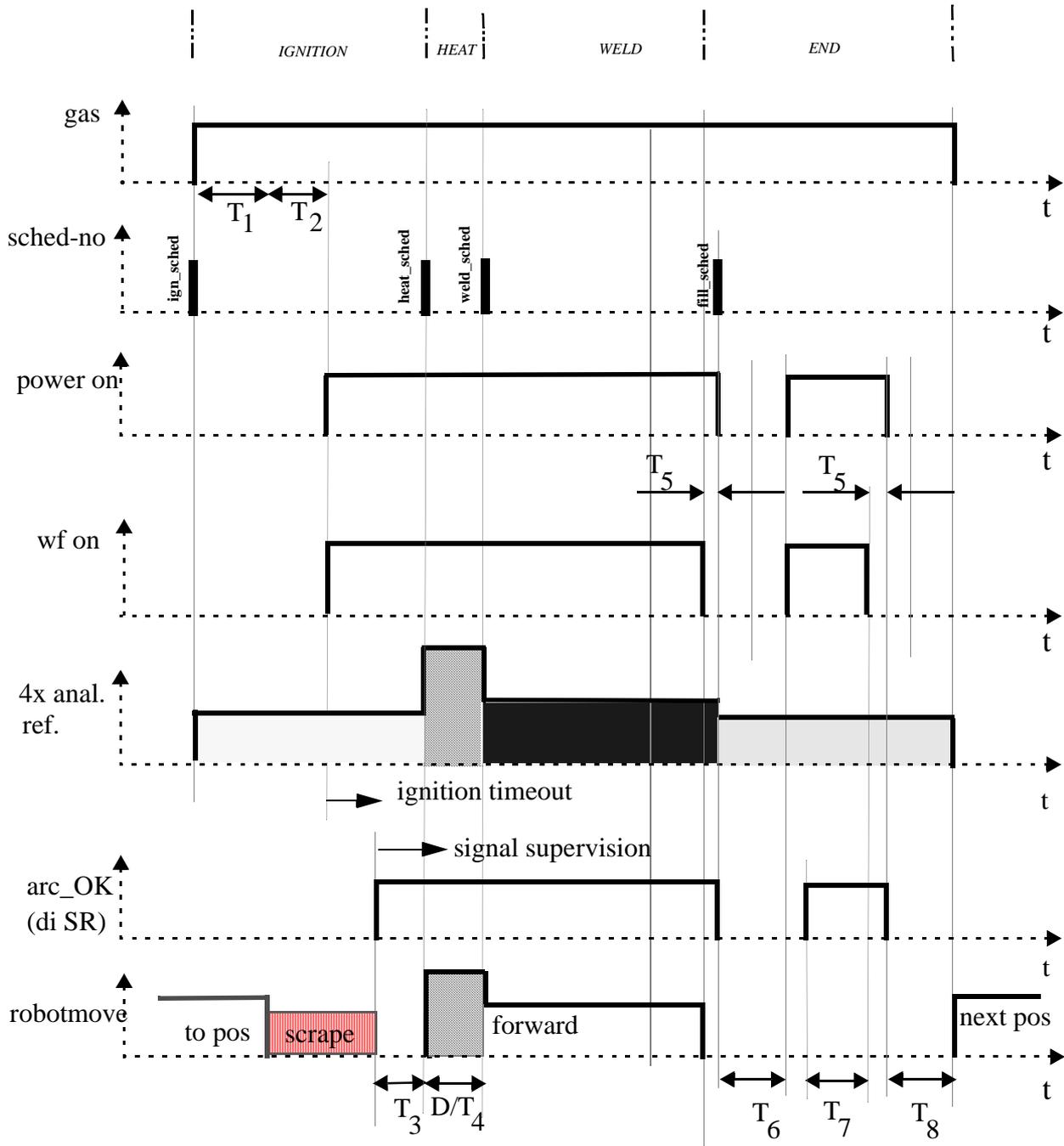
All voltages can be expressed in two ways (determined by the welding equipment):

- As absolute values (only positive values are used in this case).
- As corrections of values set in the process equipment (both positive and negative values are used in this case).

*Feeding the weld electrode* in this section refers to MIG/MAG welding. In the case of TIG welding:

- A *cold wire* is supplied to the *wire feed*.
- The necessary *welding current* reference value can be connected to any of the three analog outputs that are not used. (The *Welding voltage* reference is not used.)

The welding sequence



T1: max. gas\_purge/arc\_preset time  
 T2: gas\_preflow time  
 T3: ignition\_movement\_delay time  
 D/T4: heating distance/time

T5: burnback time  
 T6: max cooling/arc\_preset time  
 T7: filling time  
 T8: max cooling/gas\_postflow time

---

## Components

### Component group: Ignition

#### **purge\_time**

Data type: *num*

The time (in seconds) it takes to fill gas lines and the welding gun with protective gas, so-called “gas purging”.

If the first weld instruction contains the argument *\On* (flying start), the gas flow is activated at the specified gas purge time before the programmed position is reached.

If the positioning time to the start position of the weld is shorter than the gas purge time, or if the *\On* argument is not used, the robot waits in the weld start position until the gas purge time has expired.

#### **preflow\_time**

Data type: *num*

The time (in seconds) it takes to preflow the weld object with protective gas, so-called “gas preflowing”.

The robot is stationary in position during this time before the arc is ignited.

#### **ign\_sched**

(*ignition schedule*)

Data type: *num*

The identity (expressed as a number) of a weld program in connected welding equipment. It is sent to the welding equipment to be used during ignition of the arc.

See System Parameter *Arc Welding - Equipment - schedport\_type*.

#### **ign\_voltage**

Data type: *num*

The welding voltage (in volts) during ignition of the arc.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

#### **ign\_wirefeed**

Data type: *num*

The feed speed of the weld electrode during ignition of the arc.

The unit is defined in the system parameter *Arc Welding - Units - unit\_feed* and, as a rule, is m/minute or inches per minute.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**ign\_current** Data type: *num*

The welding current during ignition of the arc.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**ign\_volt\_adj** (*ignition voltage adjustment*) Data type: *num*

The welding voltage adjustment during ignition of the arc.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

This signal can be used for arbitrary purposes when something needs to be controlled using an analog output signal.

**ign\_curr\_adj** (*ignition current adjustment*) Data type: *num*

The current adjustment during ignition of the arc.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

This signal can be used for arbitrary purposes when something needs to be controlled using an analog output signal.

**ign\_move\_delay** (*ignition movement delay*) Data type: *num*

The delay (in seconds) from the time the arc is considered stable at ignition until the heating phase is started. The ignition references remain valid during the ignition movement delay.

**scrape\_start** (*scrape start type*) Data type: *num*

Type of scrape at weld start. Scrape type at restart will not be affected (it will always be Weaving scrape).

Scrape types:

0 No scrape. No scrape will occur at weld start.

1 Weaving scrape.

2 Fast scrape. The robot does not wait for the arc OK signal at the start point. However, the ignition is considered incorrect if the ignition timeout has been exceeded.

### Component group: Heat

**heat\_speed** Data type: *num*

The welding speed during heating at the start of the weld phase.

The unit is defined in the system parameter *Arc Welding - Units- velocity\_unit* and, as a rule, is mm/s or inches per minute.

**heat\_time** Data type: *num*

The heating time (in seconds) at the start of the weld phase.

*Heat\_time* is only used during timed positioning and when *heat\_distance* or *heat\_speed* equal zero.

**heat\_distance** Data type: *num*

The distance along which heat data must be active at the start of the weld.

The unit is defined in *System Parameters - Arc Welding - Units - length\_unit* and as a rule, is mm or inches.

**heat\_sched** (*heating schedule*) Data type: *num*

The identity (expressed as a number) of a weld program in connected welding equipment. It is sent to the welding equipment when the arc has been ignited and is used during heating.

See System Parameter *Arc Welding - Equipment - schedport\_type*.

**heat\_voltage** Data type: *num*

The welding voltage (in volts) during heating.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**heat\_wirefeed** Data type: *num*

The feed speed of the weld electrode during heating.

The unit is defined in the system parameter *Arc Welding - Units - unit\_feed* and, as a rule, is m/minute or inches per minute.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**heat\_current** Data type: *num*

The welding current during heating.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**heat\_volt\_adj** (*heating voltage adjustment*) Data type: *num*

The voltage adjustment for heating.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.



**postflow\_time** Data type: *num*

The time (in seconds) required for purging with protective gas after the end of a process. The purpose of gas postflow is to prevent the weld electrode and the seam from oxidizing during cooling.

**fill\_sched** (*finish schedule*) Data type: *num*

The identity (expressed as a number) of a weld program in connected welding equipment. It is sent to the welding equipment when the weld phase is completed and is used when crater-filling.

See System Parameter *Arc Welding - Equipment - schedport\_type*.

**fill\_voltage** (*crater-filling voltage*) Data type: *num*

The welding voltage (in volts) during crater-filling at the end phase of a process.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**fill\_wirefeed** (*crater-filling wirefeed*) Data type: *num*

The feed speed of the weld electrode when crater-filling.

The unit is defined in the system parameter *Arc Welding - Units - unit\_feed* and, as a rule, is m/minute or inches per minute.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**fill\_current** (*crater-filling current*) Data type: *num*

The welding current during crater-filling at the end phase of a process.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

**fill\_volt\_adj** (*filling voltage adjustment*) Data type: *num*

The voltage adjustment during crater-filling.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

This signal can be used for arbitrary purposes when something needs to be controlled using an analog output signal.

**fill\_curr\_adj** (*filling current adjustment*)Data type: *num*

The current adjustment during crater-filling.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

This signal can be used for arbitrary purposes when something needs to be controlled using an analog output signal.

---

## Structure

```

<data object of seamdata>
  <purge_time of num>
  <preflow_time of num>
  <ign_sched of num>
  <ign_voltage of num>
  <ign_wirefeed of num>
  <ign_current of num>
  <ign_volt_adj of num>
  <ign_curr_adj of num>
  <ign_move_delay of num>
  <scrape_start of num>
  <heat_speed of num>
  <heat_time of num>
  <heat_distance of num>
  <heat_sched of num>
  <heat_voltage of num>
  <heat_wirefeed of num>
  <heat_current of num>
  <heat_volt_adj of num>
  <heat_curr_adj of num>
  <cool_time of num>
  <fill_time of num>
  <bback_time of num>
  <rback_time of num>
  <postflow_time of num>
  <fill_sched of num>
  <fill_voltage of num>
  <fill_wirefeed of num>
  <fill_current of num>
  <fill_volt_adj of num>
  <fill_curr_adj of num>

```

Note that the structure changes depending on the configuration of the robot.

---

**Related information**

	<u>Described in:</u>
Weld data	Data Types - <i>welddata</i>
Installation parameters for welding equipment and functions	System Parameters - <i>Arc Welding</i>
Process phases and time diagrams	RAPID Summary- <i>Arc Welding</i>
Arc welding instructions	Instructions - <i>ArcL, ArcC</i>



---

---

**welddata****Weld data**

*Welddata* is used to control the weld during the weld phase, that is, from when the arc is established until the weld is completed.

Other phases, such as the start and end phases, are controlled using *seamdata*.

---

**Description**

Weld data describes data that is often changed along a seam. Weld data used in a given instruction along a path affects the weld until the specified position is reached. Using instructions with different weld data, it is thus possible to achieve optimum control of the welding equipment along an entire seam.

Weld data affects the weld when fusion has been established (after heating) at the start of a process.

In the case of a *flying start*, the arc is not ignited until the destination position of the arc welding instruction with the *\On* argument is reached, which means that weld data does not have any effect on the weld in this instruction.

If one arc welding instruction is exchanged for another during a weld, new weld data will occur in the middle of the corner path.

**Note.** Some of the components of weld data depend on the configuration of the robot. If a given feature is omitted, the corresponding component is left out of the weld data. The conditions that must be met for components to exist are described in the system parameters.

All voltages can be expressed in two ways (determined by the welding equipment):

- As absolute values (only positive values are used in this case).
- As corrections of values set in the process equipment (both positive and negative values are used in this case).

*Feeding the weld electrode* in this section refers to MIG/MAG welding. In the case of TIG welding:

- A *cold* wire is supplied to the *wire feed*.
- The necessary *welding current* reference value can be connected to any of the three analog outputs that are not used. (The *Welding voltage* reference is not used.)

Example

```

MoveJ p1, v100, z10, gun1;
MoveJ p2, v100, fine, gun1;
ArcL \On, p3, v100, seam1, weld1, weave1, fine, gun1;
ArcL p4, v100, seam1, weld2, weave1, z10, gun1;
ArcL \Off, p5, v100, seam1, weld3, weave3, fine, gun1;
MoveJ p6, v100, z10, gun1;

```

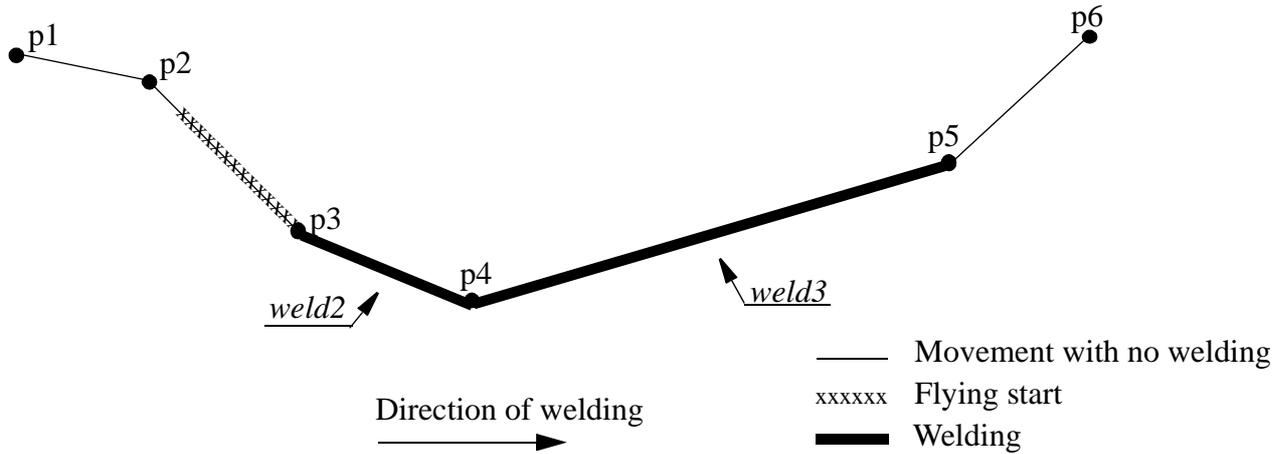
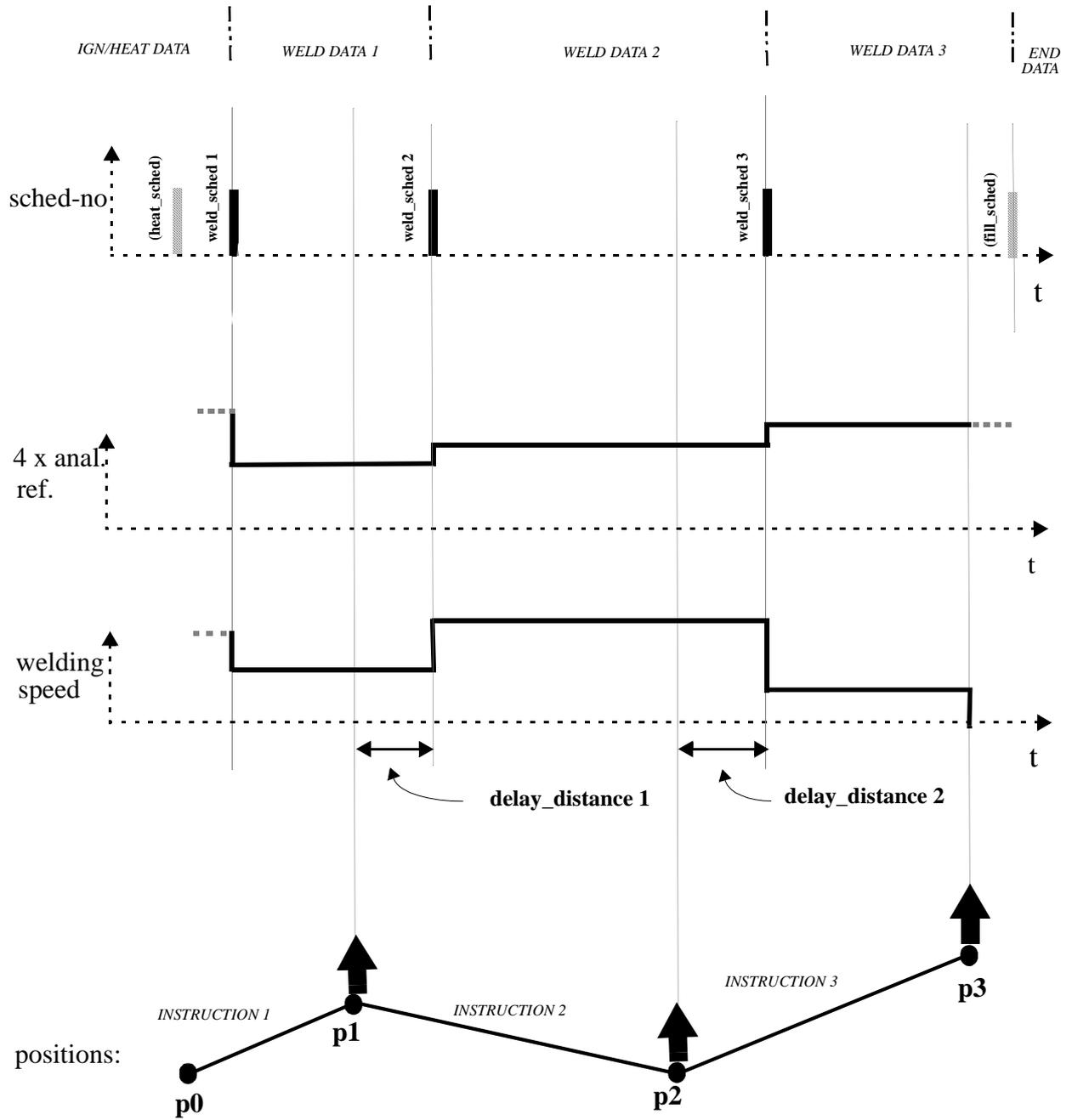


Figure 35 Weld data, such as speed and welding voltage, can be changed at each position.

Weld data is changed in every instruction. As the argument \On is used in the first instruction, the first weld data is never used.

### The welding sequence



---

**Components**

<b>weld_sched</b>	<i>(weld schedule)</i>	Data type: <i>num</i>
<p>The identity (expressed as a number) of weld programs to send to the welding equipment.</p> <p>See System Parameter <i>Arc Welding - Equipment- schedport_type</i>.</p>		
<b>weld_speed</b>		Data type: <i>num</i>
<p>The desired welding speed.</p> <p>The unit is defined in the system parameter <i>Arc Welding - Units- velocity_unit</i> and, as a rule, is mm/s or inches per minute.</p> <p>If the movements of external axes are coordinated, the welding speed is the relative speed between the tool and the object.</p> <p>If the movements of external axes are not coordinated, the welding speed is the TCP speed. The speed of the external axes is then described in the instruction's speed data. The slowest axis determines the speed to enable all axes to reach the destination position at the same time.</p>		
<b>weld_voltage</b>		Data type: <i>num</i>
<p>The welding voltage (in volts) during the weld phase.</p> <p>The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.</p>		
<b>weld_wirefeed</b>		Data type: <i>num</i>
<p>The feed speed of the weld electrode during the weld phase.</p> <p>The unit is defined in the system parameter <i>Arc Welding - Units - unit_feed</i> and, as a rule, is metres per minute or inches per minute.</p> <p>The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.</p>		
<b>weld_current</b>		Data type: <i>num</i>
<p>The welding current during the weld phase.</p> <p>The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.</p>		
<b>delay_distance</b>		Data type: <i>num</i>
<p>The delay distance (after the <i>destination</i> position) for a changeover to new weld data in the next arc welding instruction.</p>		

The unit is defined in *System Parameters - Arc Welding - Units - length\_unit* and as a rule, is mm or inches.

Usually, when changing from one arc welding instruction to another, a fly-by point is used. This results in a changeover point in the middle of the corner path. By using delay distance, the new weld data starts to take effect somewhat later (see Figure 36).

In a weld *end* instruction the delay distance will have no effect.

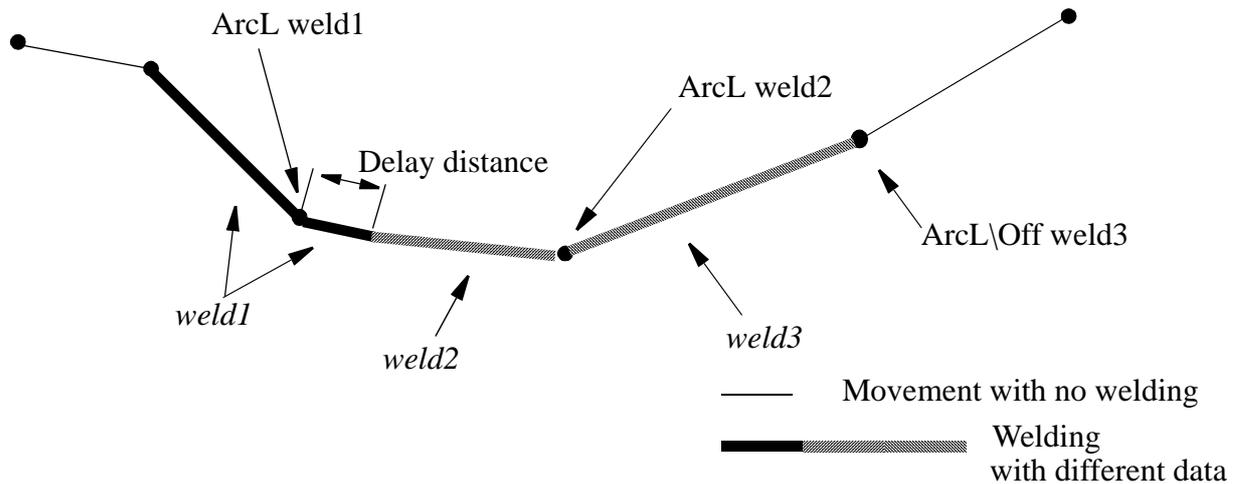


Figure 36 In the above example, the weld data changeover from weld1 to weld2 is delayed and weld2 has a *delay\_distance*=0. The *delay\_distance* in weld3 will thus have no effect.

*Delay\_distance* can, for example, be used in *ArcC* instructions to move the changeover of weld data without reprogramming the circle positions.

**weld\_volt\_adj** (*welding voltage adjustment*) Data type: *num*

The voltage adjustment during the weld phase.

The value specified is scaled and sent to the corresponding analog output, in accordance with the setting in the system parameters for analog signals.

This signal can be used for arbitrary purposes when something needs to be controlled using an analog output signal.

**weld\_curr\_adj** (*weld current adjustment*) Data type: *num*

The current adjustment during the weld phase.

The specified value is scaled and sent in accordance with the setting in the system parameters for analog signals.

This signal can be used for arbitrary purposes when something needs to be controlled using an analog output signal.

**org\_weld\_speed** (original weld speed) Data type: num

The original weld speed during the weld phase.

**Note!** Used internally by tuning functions.

**org\_weld\_voltage** (original weld voltage) Data type: num

The original weld voltage during the weld phase.

**Note!** Used internally by tuning functions.

**org\_weld\_wfeed** (original weld wirefeed speed) Data type: num

The original weld wirefeed speed during the weld phase.

**Note!** Used internally by tuning functions.

### Examples

The type of weld shown in Figure 37 is desired, with a welding voltage of 30 V and a wire feed speed of 15 m/min. The welding speed is 20 mm/s.

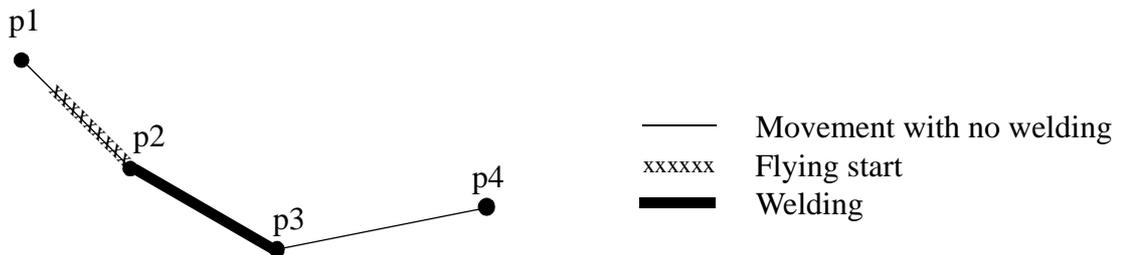


Figure 37 Welding between two points.

```
PERS welldata weld1 := [20,30,15,0];
```

```
MoveJ p1, v100, z20, gun1;
ArcL \On, p2, v100, seam1, weld1, nowave, fine, gun1;
ArcL \Off, p3, v100, seam1, weld1, nowave, fine, gun1;
MoveJ p4, v100, z20, gun1;
```

The weld data values for a weld such as the one in Figure 37 are as follows:

<u>Component</u>	<u>weld1</u>	
weld_speed	20 mm/s	Speed in relation to the seam
weld_voltage	30 V	Sent to an analog output signal
weld_wirefeed	15 m/min.	Sent to an analog output signal
delay_distance	0 mm	No delay

The weld schedule identity, weld voltage adjustment and weld current adjustment components are not active in this example.

The weld data argument does not have any effect in the ArcL \On instruction.

The type of weld shown in Figure 38 is required. The first section is to be welded using a voltage of 50 V and a wire feed speed of 20 m/min. After a specified distance on the circular arc, the voltage is to be increased to 55 V. The welding speed is 30 mm/s in each section.

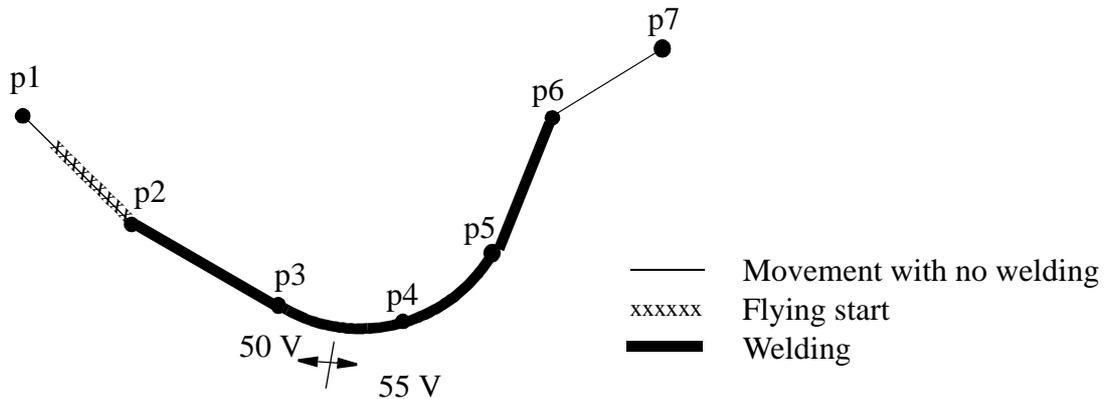


Figure 38 The weld data is changed after a specified distance on the circular path.

```
PERS welddata weld1 := [10,30,50,20,0];
PERS welddata weld2 := [10,30,55,20,17];
```

```
MoveJ p1, v100, z20, gun1;
ArcL \On, p2, v100, seam1, weld1, nowave, fine, gun1;
ArcL p3, v100, seam1, weld1, nowave, z10, gun1;
ArcC p4, p5, v100, seam1, weld2, nowave, z10, gun1;
ArcL \Off, p6, v100, seam1, weld2, nowave, fine, gun1;
MoveJ p7, v100, z20, gun1;
```

The weld data values for a weld such as the one in Figure 38 are as follows:

<u>Component</u>	<u>weld1</u>	<u>weld2</u>	
weld_sched	10	10	Identity sent to the welding equipment
weld_speed	30 mm/s	30 mm/s	
weld_voltage	50 V	55 V	
weld_wirefeed	20 m/min.	20 m/min.	
delay_distance	0 mm	17 mm	weld2 is delayed 17 mm

The weld voltage adjustment and weld current adjustment components are not active in this example.

The weld data argument does not have any effect in the ArcL \On instruction.

## Structure

<data object of *welddata*>  
<weld\_sched of *num*>  
<weld\_speed of *num*>  
<weld\_voltage of *num*>  
<weld\_wirefeed of *num*>  
<weld\_current of *num*>  
<delay\_distance of *num*>  
<weld\_volt\_adj of *num*>  
<weld\_curr\_adj of *num*>  
<org\_weld\_speed of *num*>  
<org\_weld\_voltage of *num*>  
<org\_weld\_wfeed of *num*>

Note that the structure changes depending on the configuration of the robot.

## Related information

	<u>Described in:</u>
Seam data	Data Types - <i>seamdata</i>
Installation parameters for welding equipment and functions	System Parameters - <i>Arc Welding</i>
Process phases	RAPID Summary- <i>Arc Welding</i>
Arc welding instructions	Instructions - <i>ArcL, ArcC</i>

---

---

**weavedata****Weave data**

*Weavedata* is used to define any weaving carried out during arc welding.

Weaving can be used during the heat and weld phases of a seam.

---

**Description**

Weaving is a movement, superimposed on the basic path of the process.

There are three types of weaving pattern to choose from: zigzag, V-shaped and triangular weaving. These are illustrated in Figure 39 to Figure 41.

All weave data components apply to both the heat phase and the weld phase.

The unit for weave data components, that specifies a distance and is defined in the parameter *Arc Welding - Units - length\_unit*, is as a rule expressed in mm or inches. (These components are *weave\_cycle* (for *geometric and wrist weave*), *\_width*, *\_height*, *\_bias* and *dwell* lengths.)

**Note.** Some of the components of weave data depend on the configuration of the robot. If a given feature is omitted, the corresponding component is left out of the weave data. The conditions that must be met for components to exist are described in the system parameters.

---

**Components**

**weave\_shape** (weld weave shape) Data type: *num*

The shape of the weaving pattern in the weld phase.

Specified value    Weaving pattern

0                    No weaving.

1                    Zigzag weaving as illustrated in Figure 39.

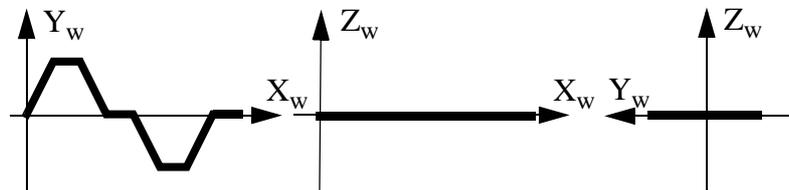


Figure 39 Zig-zag weaving results in weaving horizontal to the seam.



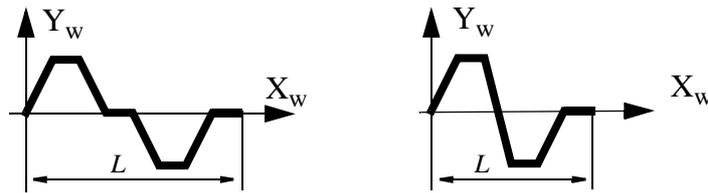


Figure 43 The length (L) of the weaving cycle for zig-zag, V-shaped and triangular weaving.

**Frequency**

The component weave\_cycle is defined as the frequency of the weaving cycle in the weld phase for weaving types 2 and 3 (see Figure 44).

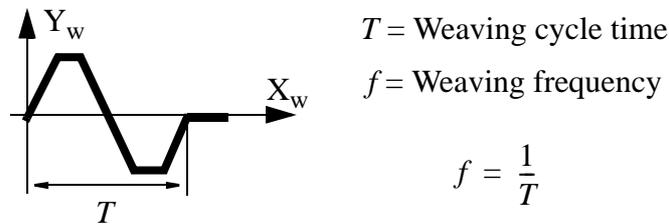


Figure 44 The frequency (f) of the weaving cycle for zig-zag weaving.

**weave\_width**

Data type: num

The width of the weaving pattern in the weld phase (see Figure 45).

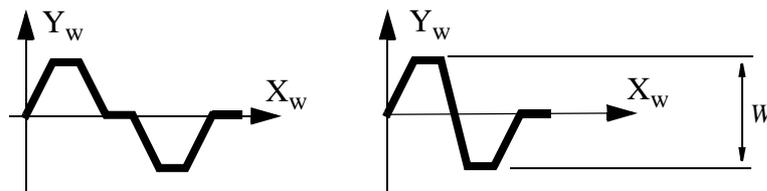


Figure 45 The width (W) of the weaving pattern for all weaving patterns.

**weave\_height**

Data type: num

The height of the weaving pattern during V-shaped and triangular weaving (see Figure 46).

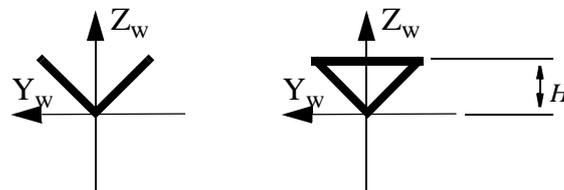


Figure 46 The height (H) of the weaving pattern for V-shaped and triangular weaving.

**dwel\_left**

Data type: num

The length of the dwell used to force the TCP to move only in the direction of the seam at the left turning point of the weave (see Figure 47).

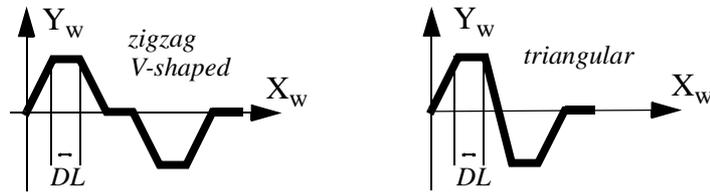


Figure 47 The length of the left dwell (DL) for different types of weaving patterns.

**dwell\_center**

Data type: num

The length of the dwell used to force the TCP to move only in the direction of the seam at the centre point of the weave (see Figure 48).

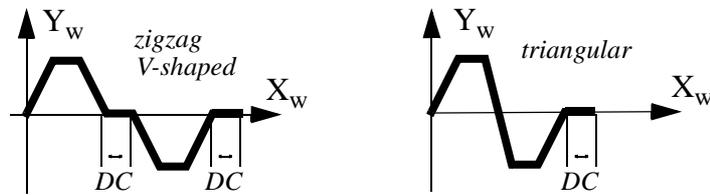


Figure 48 The length of the central dwell (DC) for different types of weaving patterns.

**dwell\_right**

Data type: num

The length of the dwell used to force the TCP to move only in the direction of the seam at the right turning point of the weave (see Figure 49).

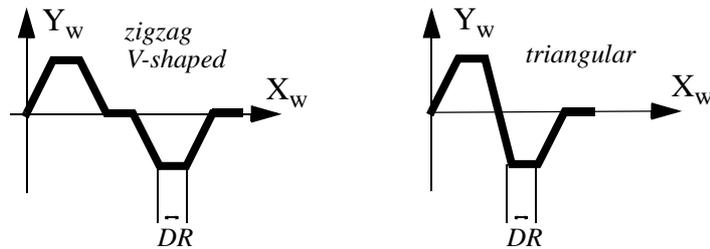


Figure 49 The length of the right dwell (DR) for different types of weaving patterns.

**weave\_dir**

(weave direction angle)

Data type: num

The weave direction angle horizontal to the seam (see Figure 50). An angle of zero degrees results in a weave that is vertical to the seam.

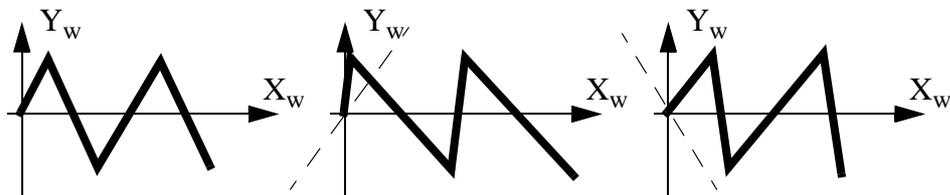


Figure 50 The shape of the weaving pattern at 0 degrees and at a positive and negative angle.

**weave\_tilt** (weave tilt angle) Data type: num

The weave tilt angle, vertical to the seam (see Figure 51). An angle of zero degrees results in a weave which is vertical to the seam.

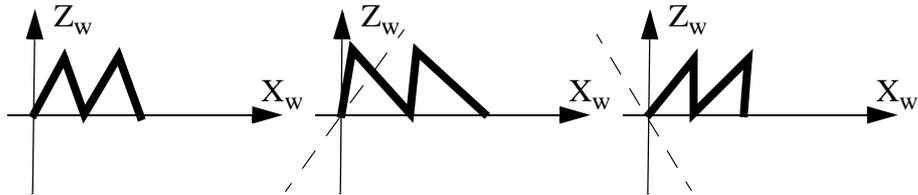


Figure 51 V-weaving at 0 degrees and at a positive and negative angle.

**weave\_ori** (weave orientation angle) Data type: num

The weave orientation angle, horizontal-vertical to the seam (see Figure 52). An angle of zero degrees results in symmetrical weaving.

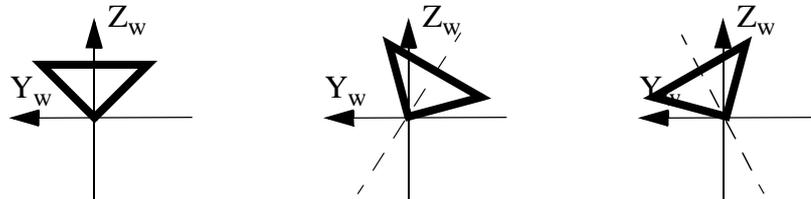


Figure 52 Triangular weaving at 0 degrees and at a positive and negative angle.

**weave\_bias** (weave centre bias) Data type: num

The bias horizontal to the weaving pattern (see Figure 53). The bias can only be specified for zig-zag weaving and may not be greater than half the width of the weave.

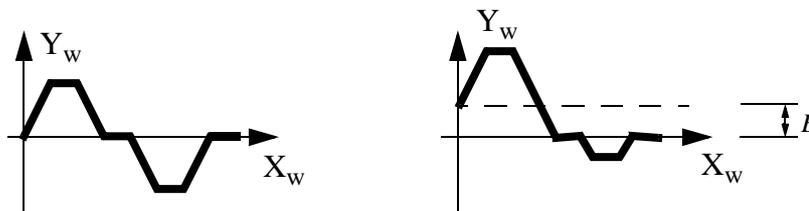


Figure 53 Zig-zag weaving with and without bias ( $B$ ).

**weave\_sync\_left** Data type: num

The coordination position to the left of the weaving pattern. It is specified as a percentage of the width on the left of the weaving centre. When weaving is carried out beyond this point, a digital output signal is automatically set to one, as illustrated in Figure 54. This type of coordination is intended for seam tracking using WeldGuide.

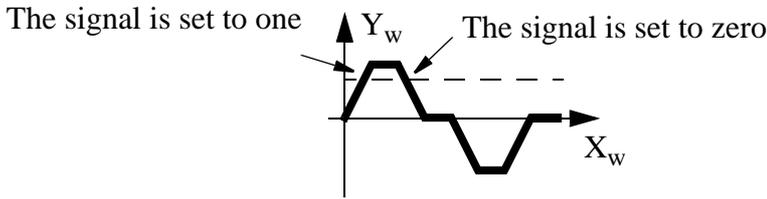


Figure 54 When WeldGuide is used, a sync. signal is required.

**weave\_sync\_right**

Data type: num

The coordination position to the right of the weaving pattern. It is specified as a percentage of the width on the right of the weaving centre. When weaving is carried out beyond this point, a digital output signal is automatically set to one, as illustrated in Figure 55. This type of coordination is intended for seam tracking using WeldGuide.

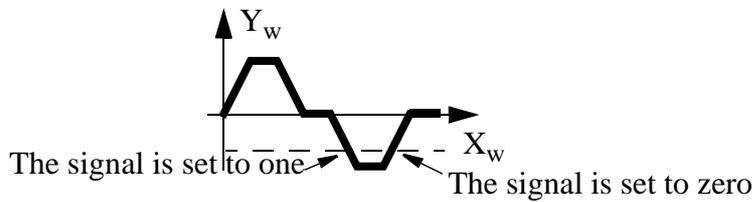


Figure 55 When WeldGuide is used, a sync. signal is required.

**wg\_track\_on**

Data type: num

Activate the weldguide seam tracker.

---

**Limitations**

The maximum weaving frequency is 2 Hz.

The inclination of the weaving pattern must not exceed the ratio 1:10 (84 degrees). (See Figure 56).

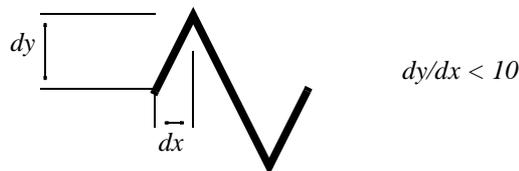


Figure 56 The weaving pattern may not be inclined more than in the ratio 1:10.

---

**Structure**

<data object of *weavedata*>  
<weave\_shape of *num*>  
<weave\_type of *num*>  
<weave\_length of *num*>  
<weave\_width of *num*>  
<weave\_height of *num*>  
<dwel\_left of *num*>  
<dwel\_center of *num*>  
<dwel\_right of *num*>  
<weave\_dir of *num*>  
<weave\_tilt of *num*>  
<weave\_ori of *num*>  
<weave\_bias of *num*>  
<weave\_sync\_left of *num*>  
<weave\_sync\_right of *num*>  
<wg\_track\_on of *num*>

---

**Related information**

	<u>Described in:</u>
Installation parameters for welding equipment and functions	System Parameters - <i>Arc Welding</i>
Process phases and timing schedules	RAPID Summary - <i>Arc Welding</i>
Arc-welding instructions	Instructions - <i>ArcL, ArcC</i>



---



---

## ArcRefresh Refresh arcweld data

*ArcRefresh* is used to tune aw process parameters during program execution.

---

### Example

```

PROC PulseWeld()

! Setup a two Hz timer interrupt
  CONNECT intno1 WITH TuneTrp;
  ITimer ,0.5 ,intno1;

! Weld the seam
  ArcL \On, p1, v100, seam1, weld5, noweave, fine, gun1;
  ArcL \Off, p2, v100, seam1, weld5, noweave, fine, gun1;

  IDelete intno1;

ENDPROC

TRAP TuneTrp

! Modify the weld_voltage component of active welddata.

  IF HighValueFlag =TRUE THEN
    weld5.weld_voltage := 10;
    HighValueFlag := FALSE;
  ELSE
    weld5.weld_voltage := 15;
    HighValueFlag := TRUE;
  ENDIF

! Order the process control to refresh process parameters
  ArcRefresh;

ENDTRAP

```

The weld voltage will be switched between 10 and 15 volts by the trap routine at a two Hz rate.

---

### Syntax

```
ArcRefresh ‘;’
```

---

**Related information**

Performing a circular weld  
Performing a linear weld  
Definition of weld data  
Definition of weave data  
Installation parameters for welding  
equipment and welding functions

Described in:

Instructions - *ArcC*  
Instructions - *ArcL*  
Data Types - *welddata*  
Data Types - *weavedata*  
System Parameters - *Arc Welding*

---



---

## ArcKill Kill the AW process

*ArcKill* is used in advanced error handlers to kill the AW process.

---

### Example

```

PROC main()

    WeldSeam1;
    WeldSeam2;

ERROR
    TPRadFk ans "The weld failed", "", "", "", "Service", "OK";

    TEST ans
    CASE 4:
        service_routine;
        TRYNEXT;

    DEFAULT:
        TRYNEXT
    ENDTEST

ENDPROC

PROC WeldSeam1()

    ! Weld the seam
    ArcL \On, p1, v100, seam1, weld5, nowave, fine, gun1;
    ArcL \Off, p2, v100, seam1, weld5, nowave, fine, gun1;

ERROR
    TEST ERRNO
    CASE AW_IGNI_ERR:
    !    Try to restart the process if this was an ignition error.
        RETRY;

    DEFAULT:
    !    Kill the aw process and raise the error to the main routine
        ArcKill;
        RAISE
    ENDTEST

ENDPROC

```

---

**Syntax**

ArcKill ‘;’

---

**Related information**

Performing a circular weld

Performing a linear weld

Described in:

Instructions - *ArcC*

Instructions - *ArcL*

---

---

**arcdata****Arc process data**

*Arcdata (Arc Process Data)* is used to define a number of arc welding process parameters that are essential for the process:

- The latest used parameters (velocity, wirefeed speed and voltage).
- Statistical data (mean and min/max values for voltage and current in the latest seam).

This data type does not normally have to be used since these values can be accessed using the system variable ARC\_DATA.

---

**Description**

The latest used parameters (stored in the system variable ARC\_DATA) are affected in any circumstances.

The following requirements are applicable for affecting the statistical data:

- The system is an ARCITEC system.
- The system is configured so it uses the analog inputs for voltage and current. In this case external measuring equipment must be used.

---

**Components**

**req\_vel** Data type: *num*

Latest requested velocity.

**req\_volt** Data type: *num*

Latest requested weld voltage.

**req\_wfd\_sp** Data type: *num*

Latest requested weld wirefeed speed.

**act\_volt** Data type: *num*

Current measured weld voltage.

**min\_volt** Data type: *num*

Lowest measured value of the weld voltage during the current seam (or latest seam).

<b>max_volt</b>	Data type: <i>num</i>
Highest measured value of the weld voltage during the current seam (or latest seam).	
<b>mean_volt</b>	Data type: <i>num</i>
Mean value of the measured weld voltage during the current seam (or latest seam).	
<b>act_curr</b>	Data type: <i>num</i>
Measured weld current at this time.	
<b>min_curr</b>	Data type: <i>num</i>
Lowest measured value of the weld current during the current seam (or latest seam).	
<b>max_curr</b>	Data type: <i>num</i>
Highest measured value of the weld current during the current seam (or latest seam).	
<b>mean_curr</b>	Data type: <i>num</i>
Mean value of the measured weld current during the current seam (or latest seam).	

---

## Example

```
ArcL\On ...  
ArcL\Off ...  
TPWrite "Voltage mean value for the latest seam = "\Num:=ARC_DATA.mean_volt;
```

When the seam is finished, the mean value of the process parameter voltage will be displayed on the TeachPendant.

**Structure**

<data object of *arcdata*>  
<req\_vel of *num*>  
<req\_volt of *num*>  
<req\_wfd\_sp of *num*>  
<act\_volt of *num*>  
<min\_volt of *num*>  
<max\_volt of *num*>  
<mean\_volt of *num*>  
<act\_curr of *num*>  
<min\_curr of *num*>  
<max\_curr of *num*>  
<mean\_curr of *num*>

---

**Related information**

Installation parameters for welding  
equipment and functions  
Process phases and timing schedules  
Arc welding instructions

Described in:

System Parameters - *Arc Welding*  
RAPID Summary - *Arc Welding*  
Instructions - *ArcL, ArcC*



---

---

**A**

arc welding 33, 41  
    blocking process 11, 12  
    configuration 21  
arc welding data 4  
arc welding instructions 3  
ArcC 3, 41  
ArcKill 77  
ArcL 3, 33  
ArcRefresh 75  
arcwelding system  
    choose 8, 10, 11, 14

**B**

Blocking 11, 12  
burnback 26

**C**

crater fill 26  
crater-filling 55

**E**

end phase 54  
Equipment  
    arc welding parameters 27

**F**

Function  
    arc welding parameters 24

**G**

Gas on/off 13

**H**

heat 52

**I**

ignition 51  
Increments  
    tuning arc welding 15  
inhibit arc welding 11, 12  
inhibit weld, weave 28

**M**

Man wiref 13  
manual gas on/off 13  
manual wire feed 13

**P**

preheating 26  
programming  
    arc welding 3  
P-start 21

**R**

RAPID Converters 21  
rollback 27

**S**

scrape start 25  
seamdata 4, 49  
Select syst. 14, 17  
supervision  
    arc welding signals 28  
System  
    arc welding parameters 22

**T**

tuning increments 15

**U**

Unit  
    arc welding parameters 23

**W**

weave  
    parameters 26  
weavedata 4, 67  
Weld Tuning 8  
welldata 4, 59  
wire feed  
    manual 13







---

ABB Robotics AB  
DPT / MT  
S-721 68 VÄSTERÅS  
SWEDEN

Telephone: +46 (0) 21 34 40 00  
Telefax: +46 (0) 21 13 25 92