Dual Loop Control

Application Note

agito

www.agito-akribis.com

Member of Akribis Systems group

Revision History

Version	Description	Date
1.0	Translated CN v1.5.	10 July 2026

Contact Information

Manufacturer	Agito Akribis Systems Ltd., Member of Akribis Systems Group
Address	56 Serangoon North Avenue 4 Singapore 555851
Telephone	+65 6484 3357
Website	www.agito-akribis.com

Copyright Notice

Products Rights

AGDx, AGCx, AGMx, AGAx, AGIx, and AGLx are products designed by Agito Akribis Systems Ltd. in Israel. Sales of the products are licensed to Akribis Systems Pte Ltd. under intercompany license agreement.

Agito Akribis Systems Ltd. has full rights to distribute above products worldwide.

Disclaimer

This product documentation was accurate and reliable at the time of its release.

Agito Akribis Systems Ltd. reserves the right to change the specifications of the product described in this manual without notice at any time.

Trademarks

Agito PCSuite is a trademark of Agito Akribis Systems Ltd..

Warranty

This product is warranted to be free of defects in material and workmanship and conforms to the specifications listed in this manual, for a period of 12 months from the shipment date from factory.

Contents

1 Introduction

1.1 About This Manual

This application note describes the implementation of dual loop control in Agito motion control systems and how to configure a dual loop control system using the Agito PCSuite software.

It also includes the parameters (keywords) related to the dual loop feature.

Note that the AGA series must be used together with the AGM800-CI motion controller.

1.2 Overview of Dual Loop System

In the indirect stage with servo control systems, transmission elements such as ball screws, planetary gearboxes, or belt drives are commonly used between the motor and the load. Due to backlash, insufficient stiffness, and other mechanical imperfections in these transmission components, the motor position often differs from the actual load position, especially while the system is in motion.

Using feedback only from the motor axis generally provides good control performance, but it cannot accurately control the actual load position. On the other hand, using only the load position feedback may result in poor control performance, and the load position is also unsuitable for commutation feedback.

A feasible solution is to adopt dual loop control architecture, which uses two position feedback signals as inputs. The motor's own position feedback is used for commutation and velocity-loop feedback, delivering the required dynamic control performance. The load feedback is used as the position feedback for the position loop, enabling precise positioning of the load. Although this design typically requires a longer tuning and optimization process, it can ensure high-performance and accurate load positioning.

This document uses a rotary brushless servo motor + ball screw + linear encoder feedback system as an example to introduce the application of the Agito dual loop control. Other types of transmission mechanisms can be treated in a similar manner.

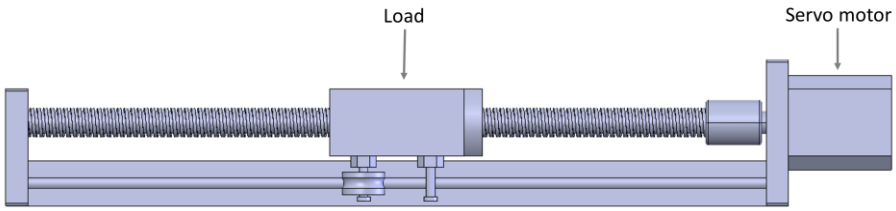


Figure 1. Indirect Stage with Servo Motor

1.3 Agito Dual Loop Control

The figure below shows how the Agito motion controller implements Dual Loop control.

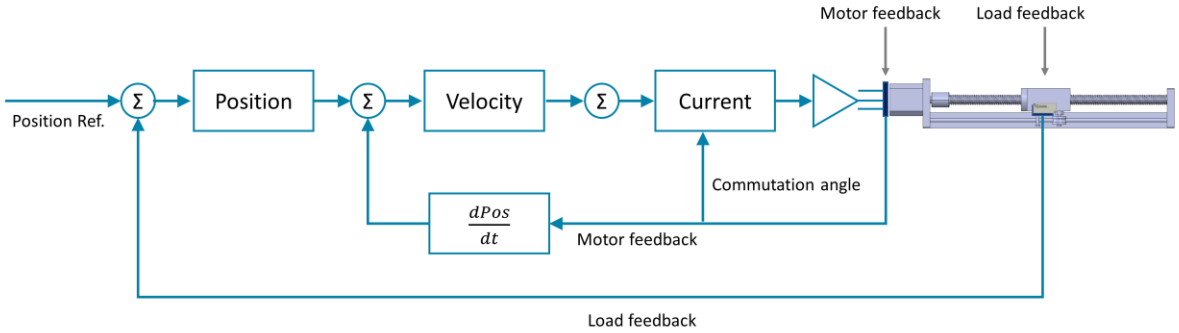


Figure 2. Dual Loop Control in Agito Amplifiers and Drives

The encoder on the load is connected to the main encoder interface (Main), which is used for position loop control.

The encoder on the motor is connected to the auxiliary encoder interface (Aux), whose feedback is used for commutation and velocity loop control. (Some Agito products do not have a dedicated AUX encoder interface; in that case, auxiliary encoder position feedback is provided via a differential input port.)

Encoder types supported by Agito products:

- Main encoder port (used as load feedback): AqB digital incremental, EnDat 2.2 absolute, Biss-C absolute
- Auxiliary encoder port (used as motor feedback): AqB digital incremental, tachometer feedback

When Dual Loop control is configured, the main encoder port (load feedback) is also used for the following functions:

- Feedback-based functions, such as Error Mapping, PEG and Position Lock.
- Homing

Use Agito PCSuite to configure encoder feedback, Dual Loop control, and tuning.

2 Operating Procedure

Before configuring Dual Loop control, it is recommended that you first perform standard control (single-loop) configuration, using only motor feedback connected to the main encoder port (Main Enc.). Single-loop control allows the use of all tuning functions, such as system frequency-domain analysis and auto-tuning, which are currently unavailable in Dual Loop control.

Once all tuning steps for single-loop control are complete and the motor runs normally, you can proceed to Dual Loop control. After completing single-loop tuning, the Dual Loop system only requires adjustment of the position loop gain.

The wiring differs between controller series when using Dual Loop control, but the debugging process is the same for all.

2.1 AGD155 Series

2.1.1 System Diagram

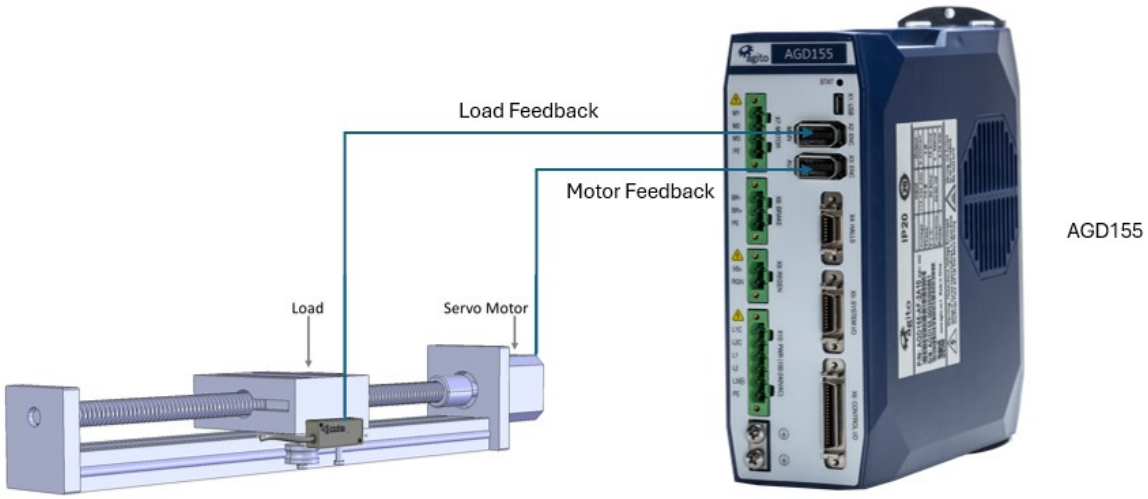


Figure 3. AGD155 Dual Loop Control System Diagram

The load encoder feedback connects to the main encoder port (X2: Main Enc.); the motor encoder feedback connects to the auxiliary encoder port (X3: Aux Enc.).

Note

Single-loop control must be performed before Dual Loop control.

2.1.2 Parameter Configuration

Step 1.

Open Agito PCSuite and set the relevant parameters (Note: if using the Agito product for the first time, please refer to the “Agito Quick Start Manual” for the relevant parameter settings).

First perform single-loop control, i.e., connect only the motor encoder feedback to the AGD155 controller's main encoder port (X2: Main Enc.), set the relevant motor parameters, and perform parameter tuning to ensure the motor encoder feedback is normal and the motor can commutate and move correctly.

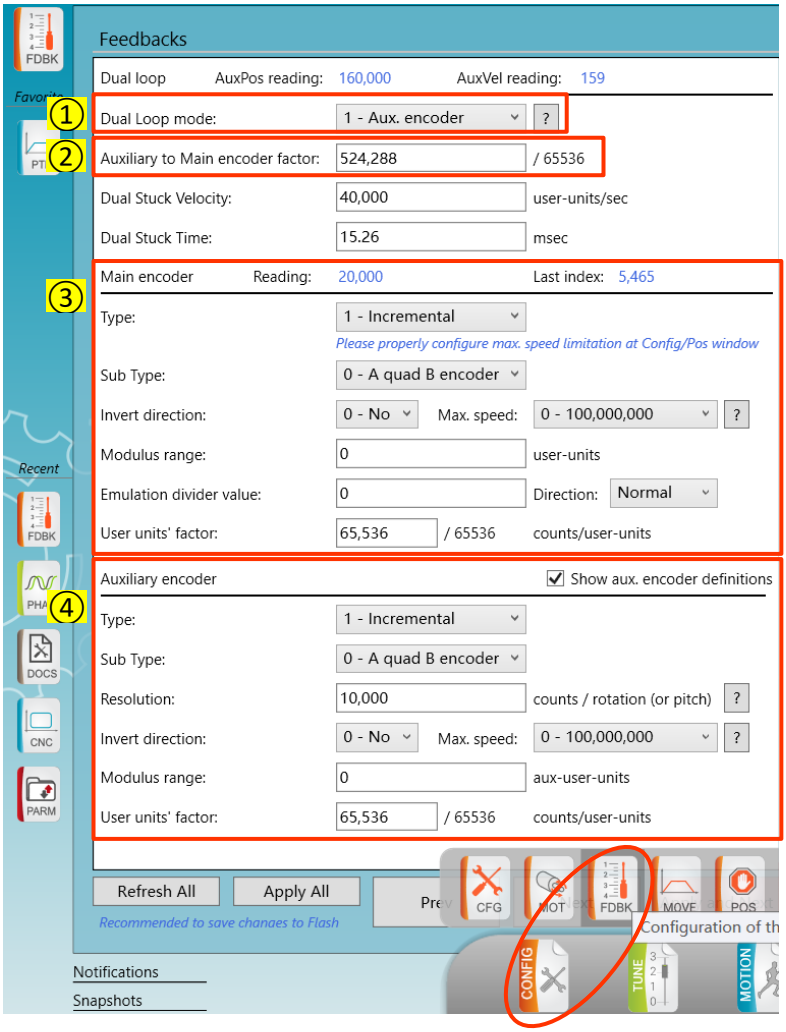
In addition, if you need to use functions such as system frequency-domain analysis and auto-tuning, they must be completed during single-loop control.

Note

During single-loop control, the motor encoder feedback connects to the main encoder port (X2: Main Enc.), while during Dual Loop control, the motor feedback connects to the auxiliary encoder port (X3: AUX Enc.).

Step 2.

After the motor runs normally, save the controller parameters (requires disabling motor first), power off, change the system wiring to Dual Loop mode, power the controller back on, and follow the steps below to change the feedback settings.



① Set Dual Loop Mode to "1 - Aux. encoder" to enable Dual Loop control mode.

② Set the auxiliary-to-main encoder scale factor (Auxiliary to Main encoder factor):

$$\text{Auxiliary to Main Encoder Factor} = \frac{Main\ Encoder\ Resolution}{Auxiliary\ Encoder\ Resolution} * 65536$$

Taking a ball screw drive as an example:

$$\text{Auxiliary to Main Encoder Factor} = \frac{Screw\ Lead\ Stroke}{Motor\ Encoder\ Resolution * Pulses/Cycle} * 65536$$

③ Set the main encoder feedback type (load feedback): supports AqB digital incremental, EnDat 2.2 absolute, and Biss-C absolute encoder feedback types.

④ Set the auxiliary encoder type (motor feedback): only AqB digital incremental encoders are supported.

The Resolution parameter should be filled in according to the motor type (Note: refer to the "Agito Quick Start Manual" for details).

After setting up the feedback, if the mechanical structure allows, you can slowly push the motor by hand and observe whether the main encoder and auxiliary encoder feedback directions are consistent. If not, you can set "Invert direction" under Main Encoder (step ③) to "1-Yes" to invert the feedback.

If the mechanical structure does not allow manual pushing (e.g., a static brake is used), after setting the parameters, you can first switch to single loop (Feedback: 0-No dual loop), move the motor briefly on the motion –> PTP/JOG page, and use an oscilloscope to monitor the APos and AAuxPos parameters to observe whether they move in the same direction (**both increasing or both decreasing**), as shown below.

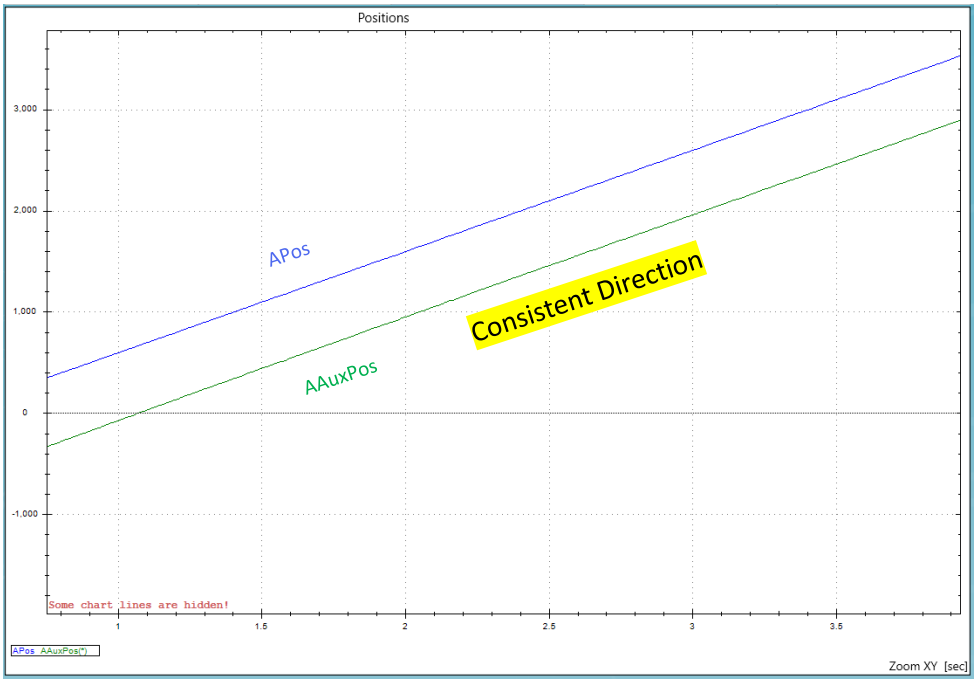
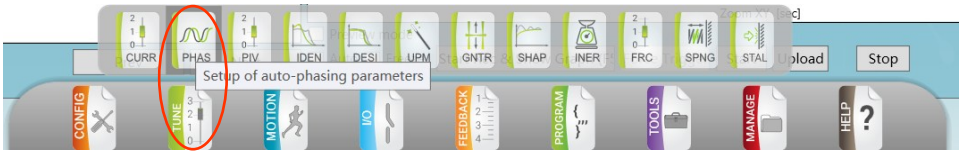
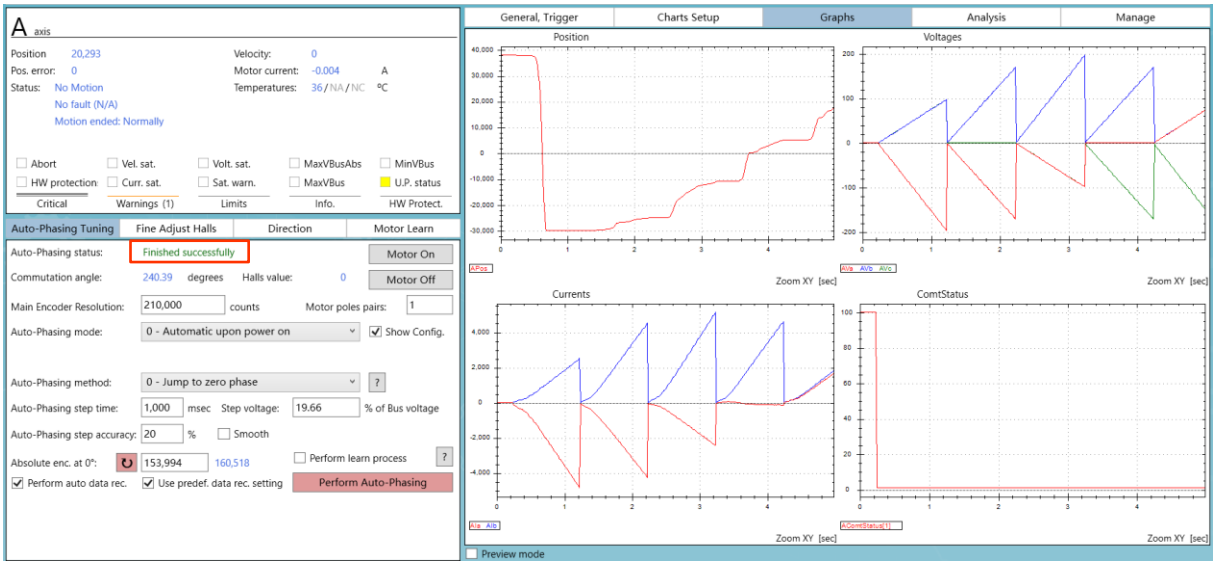


Figure 4. Pos & AuxPos Encoder Feedback

Step 3.

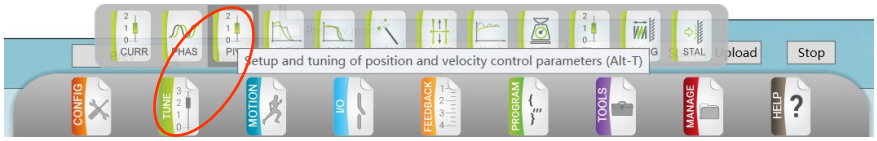
Switch to **TUNE –> PHAS** and perform commutation. A successful commutation shows: **Finished Successfully**.

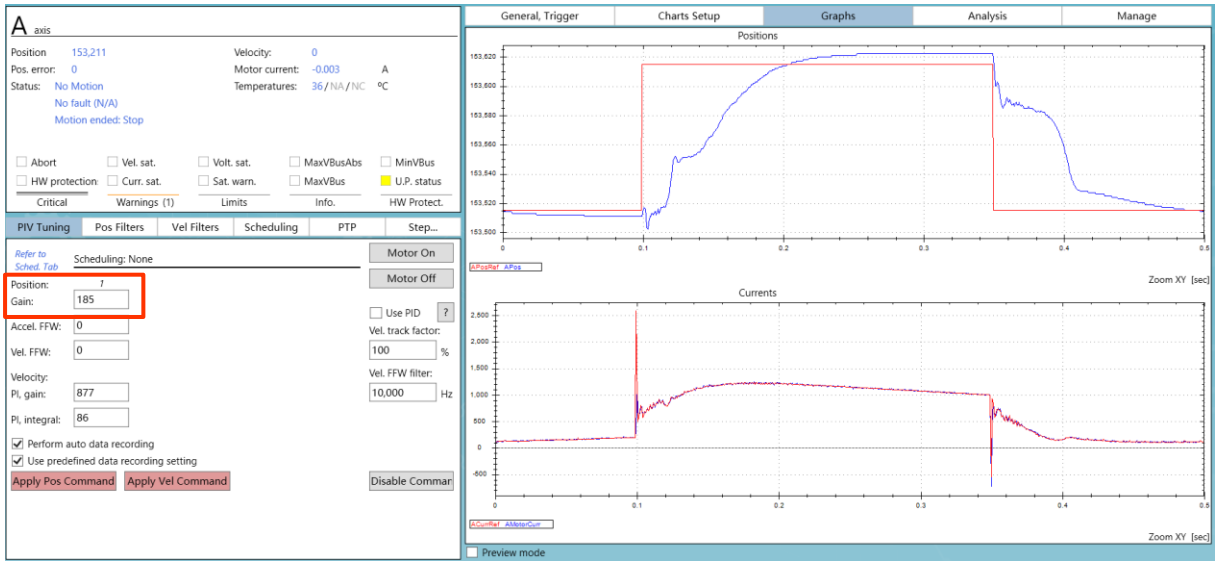




Step 4.

After commutation succeeds, switch to **TUNE** –> **PIV** –> **PIV Tuning** and perform **position loop** tuning so that the commanded position and the feedback position overlap as closely as possible.





Step 5.

Once the above steps are complete, motion control can proceed.

2.2 AGD301 Series

The AGD301 series has no dedicated Aux Enc. interface; it uses the differential input port (X4: GENERAL I/O) as the auxiliary encoder signal input port.

The AGD301 Dual Loop function requires a **customized firmware version**; please contact Agito-Akribis for support to use it.

2.2.1 System Diagram

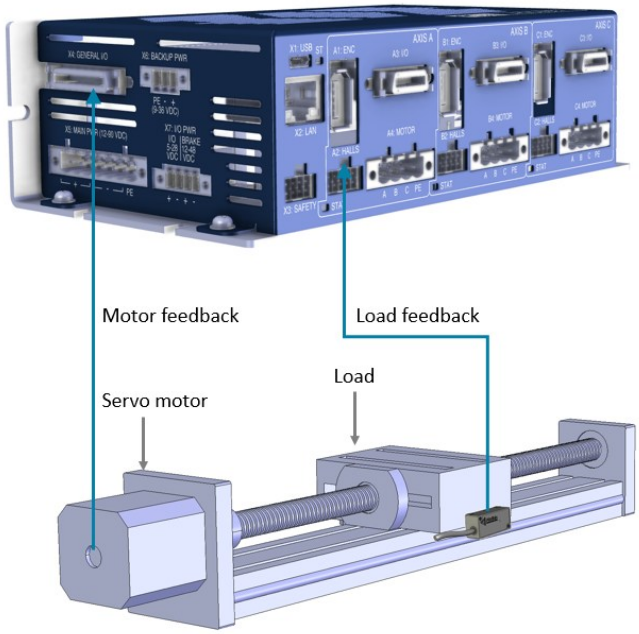


Figure 5. AGD301 Dual Loop Control System Diagram

The load encoder feedback connects to the main encoder port (X2: Main Enc.); the motor encoder feedback connects to the differential input port (X4: GENERAL I/O).

Note

Single-loop control must be performed before Dual Loop control.

Auxiliary encoder wiring definitions:

Auxiliary Encoder Pin Definition	AGD301 (X4: GENERAL I/O) Pin Definition	
A-axis Enc. A+	Bi-Dir_Diff_IO_1+	1
A-axis Enc. A-	Bi-Dir_Diff_IO_1-	2
A-axis Enc. B+	Bi-Dir_Diff_IO_2+	19
A-axis Enc. B-	Bi-Dir_Diff_IO_2-	20
B-axis Enc. A+	Bi-Dir_Diff_IO_3+	3
B-axis Enc. A-	Bi-Dir_Diff_IO_3-	4

Auxiliary Encoder Pin Definition	AGD301 (X4: GENERAL I/O) Pin Definition	
B-axis Enc. B+	Bi-Dir_Diff_IO_4+	21
B-axis Enc. B-	Bi-Dir_Diff_IO_4-	22
C-axis Enc. A+	Bi-Dir_Diff_IO_5+	25
C-axis Enc. A-	Bi-Dir_Diff_IO_5-	26
C-axis Enc. B+	Bi-Dir_Diff_IO_6+	9
C-axis Enc. B-	Bi-Dir_Diff_IO_6-	10
5V	5V	7
GND	GND	8

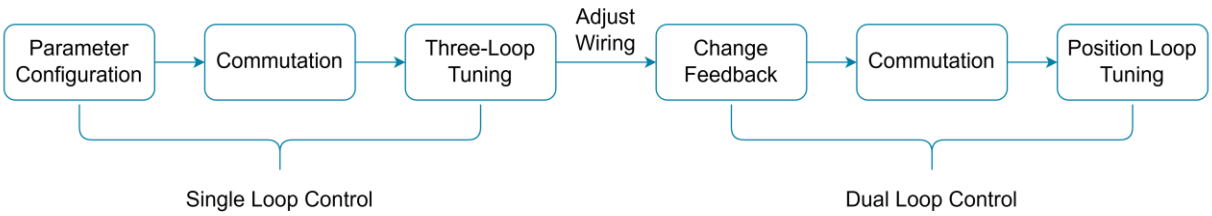
2.2.2 Parameter Configuration

Step 1.

Open Agito PCSuite and set the relevant parameters (Note: if using the Agito product for the first time, please refer to the "Agito Quick Start Manual" for the relevant parameter settings).

First perform single-loop control, i.e., connect only the motor encoder feedback to the AGD301 controller's main encoder port (A1: ENC), set the relevant motor parameters, and perform three-loop tuning to ensure the motor encoder feedback is normal and the motor can commutate and move correctly.

In addition, if you need to use functions such as system frequency-domain analysis and auto-tuning, they must be completed during single-loop control.

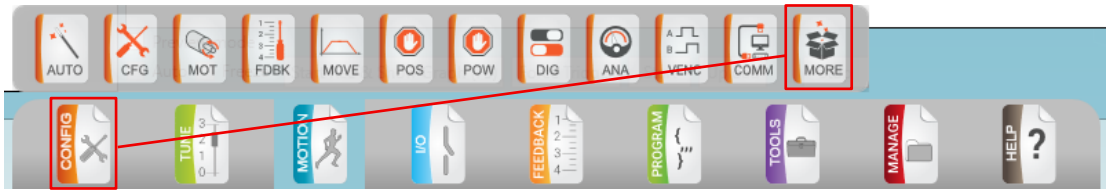


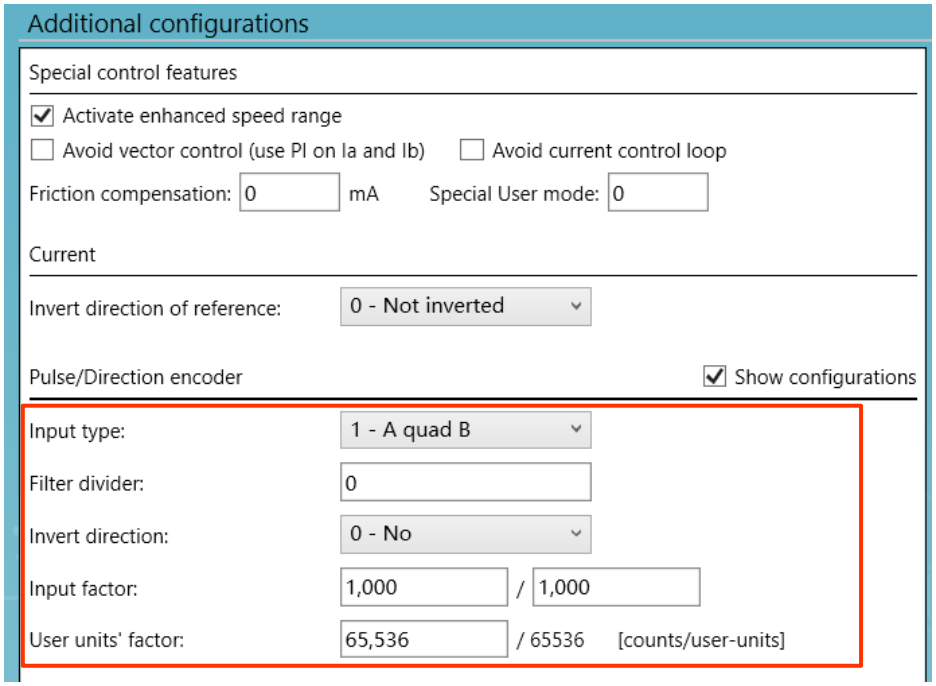
Note

During single-loop control, the motor encoder feedback connects to the main encoder port (A1: ENC), while during Dual Loop control, the motor feedback connects to the auxiliary encoder port (X4: GENERAL I/O).

Step 2.

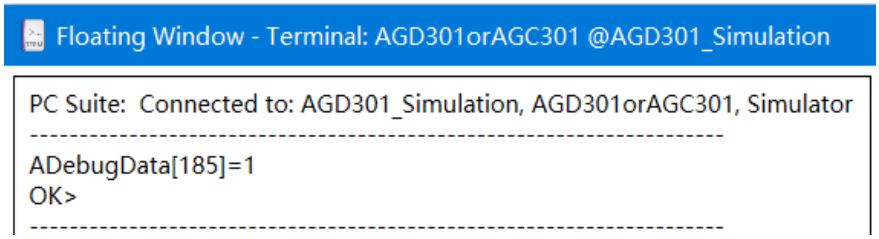
Switch to the **CONFIG** –> **MORE -> Additional Configurations** screen and set the input type to "1-A quad B".





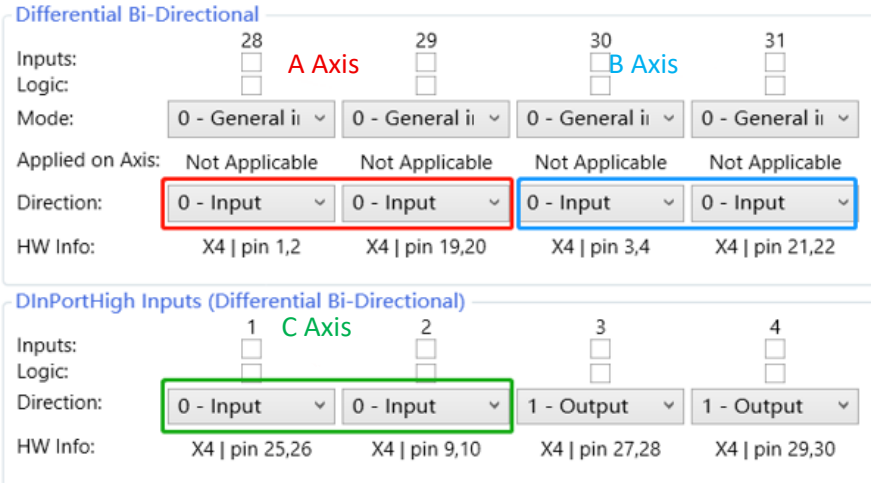
Step 3.

In the Terminal, enter "ADebugData[185]=1" to activate the Dual Loop function.



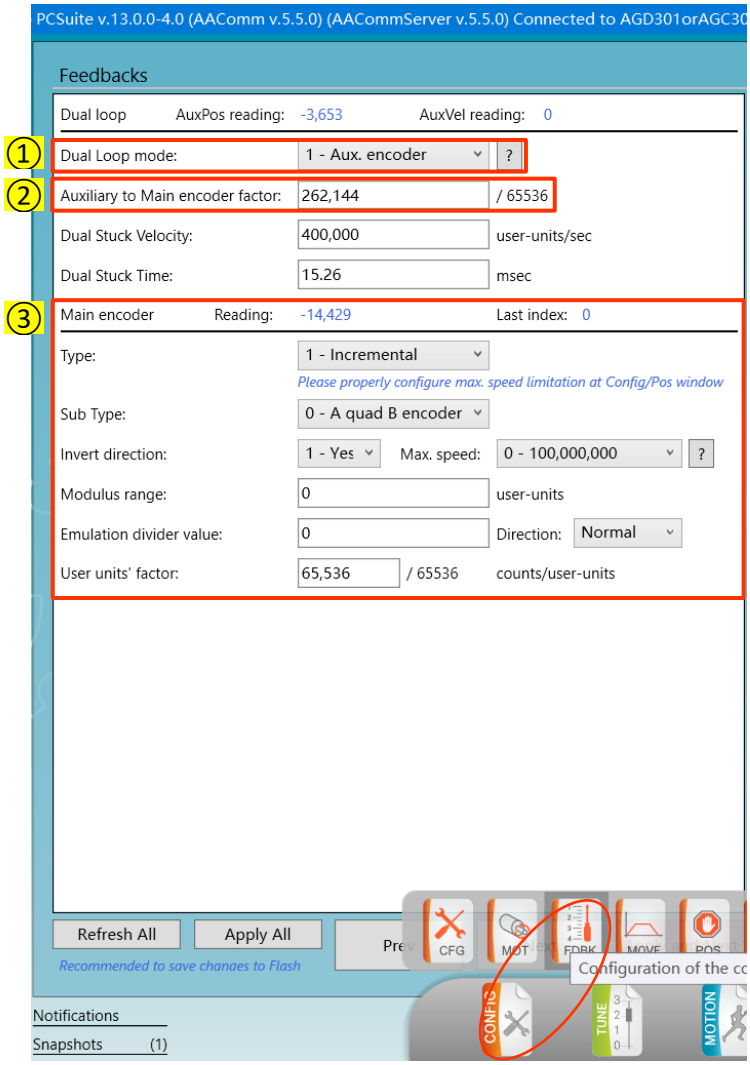
Step 4.

Switch to **I/O** –> **Discrete Inputs** screen and set the bidirectional differential setting for the corresponding axis to "**0-Input**" direction.



Step 5.

Save the controller parameters (requires disabling the motor first), power off, change the system wiring to Dual Loop control mode, power the controller back on, and follow the steps below to change the feedback settings.



① Set Dual Loop Mode to "1 - Aux. encoder" to enable Dual Loop control mode.

② Set the auxiliary-to-main encoder scale factor (Auxiliary to Main encoder factor):

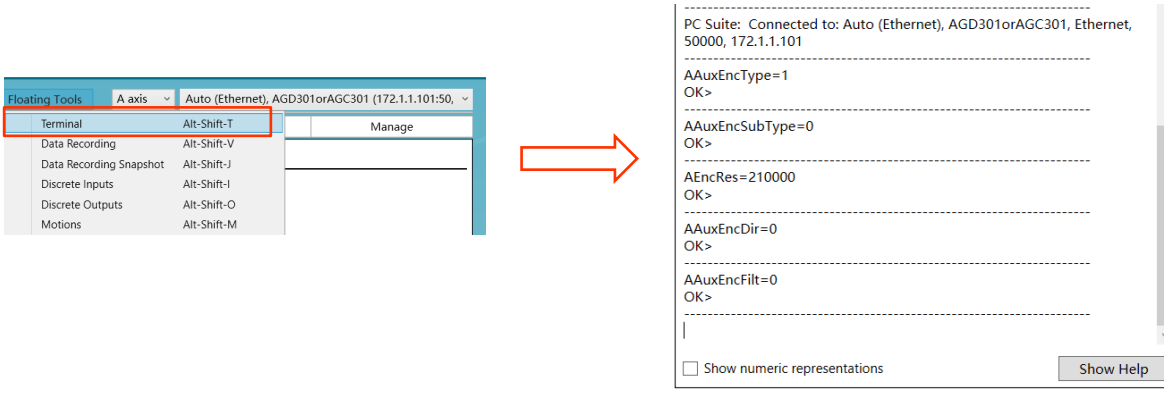
$$\text{Auxiliary to Main Encoder Factor} = \frac{Main\ Encoder\ Resolution}{Auxiliary\ Encoder\ Resolution} * 65536$$

Taking a ball screw drive as an example:

$$\text{Auxiliary to Main Encoder Factor} = \frac{Screw\ Lead\ Stroke}{Motor\ Encoder\ Resolution * Pulses/Cycle} * 65536$$

③ Set the main encoder feedback type (load feedback): supports AqB digital incremental, EnDat 2.2 absolute, and Biss-C absolute encoder feedback types.

④ Set the **auxiliary encoder type (motor feedback)**: only AqB digital incremental encoders and tachometer feedback are supported. The auxiliary encoder parameter settings have no UI in the AGD301 and can only be configured via Terminal command input:



Command Keyword	Default Value	Set Value	Meaning
AAuxEncType	1	1	1, indicates incremental encoder
AAuxEncSubType	0	0	0, indicates A quad B type
AEncRes	1	1~(2^31)-1	Pulses per motor revolution
AAuxEncDir	0	0/1	0, no inversion; 1, invert
AAuxEncFilt	0	0~255 (typical value 0)	Auxiliary encoder digital filter. When AuxEncFilt=0, Max Input Frequency = 300 MHz/(2*6)MHz; When AuxEncFilt≠0, Max Input Frequency = 300 MHz/(AuxEncFilt*2)/(2 * 6)
DebugData[185]	0	1	1, activates the Dual Loop function; 0, disables the Dual Loop function.

The Resolution (AEncRes) parameter should be filled in according to the motor type (Note: refer to the “Agito Quick Start Manual” for details). After setting up the feedback, if the mechanical structure allows, you can push the motor by hand and observe whether the main encoder and auxiliary encoder feedback directions are consistent. If not, you can set “Invert direction” under Main Encoder (step ③) to “1-Yes” to invert the feedback. If the mechanical structure does not allow manual pushing (e.g., a static brake is used), you can, after setting the parameters, first switch to single loop (Feedback: 0-No dual loop), move the motor briefly on the motion –> PTP/JOG screen, and use an oscilloscope to monitor the APos and AAuxPos parameters to observe whether they move in the same direction (**both increasing or both decreasing**), as shown below.

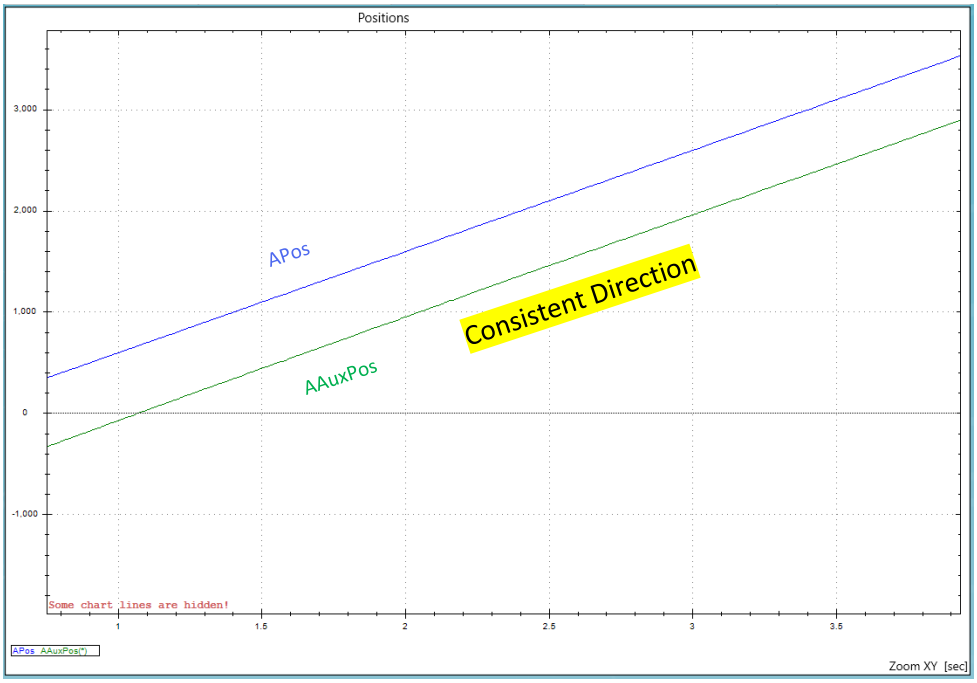
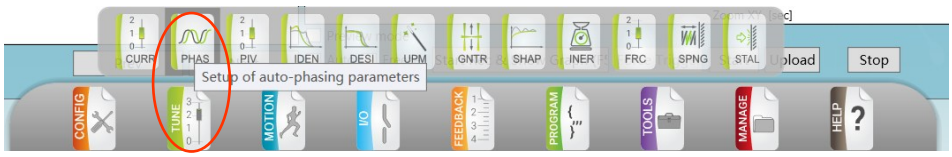
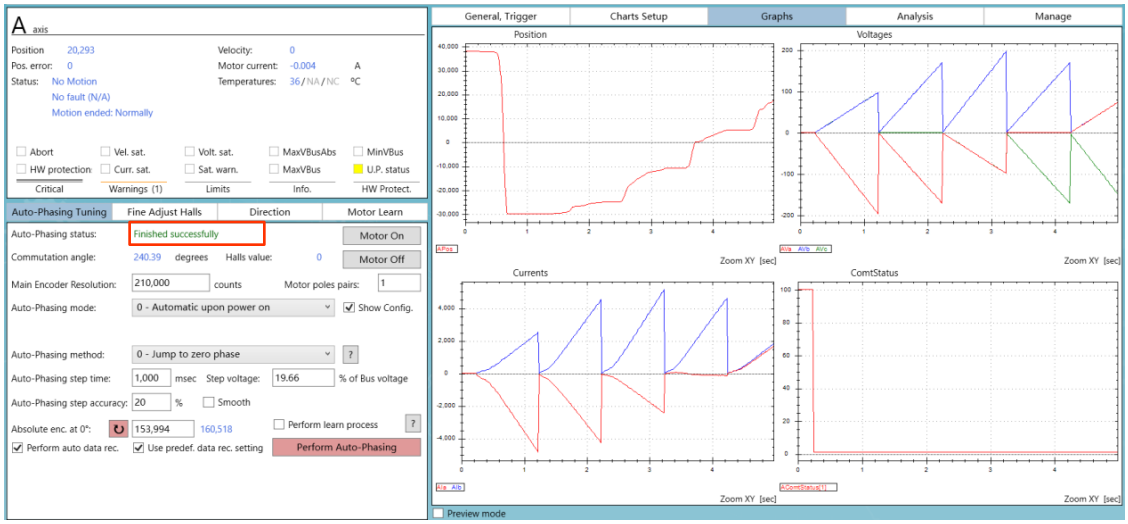


Figure 6. Pos & AuxPos Encoder Feedback

Step 6.

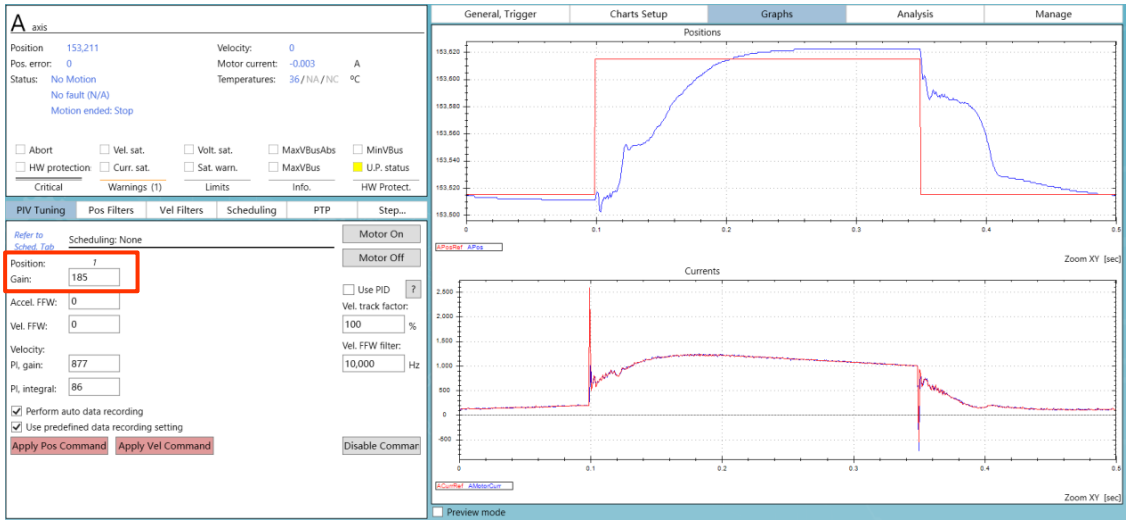
Switch to **TUNE –> PHAS** and perform commutation. A successful commutation shows: **Finished successfully.**





Step 7.

After commutation succeeds, switch to **TUNE** –> **PIV**–> **PIV Tuning** and perform **position loop** tuning so that the commanded position and the feedback position overlap as much as possible.



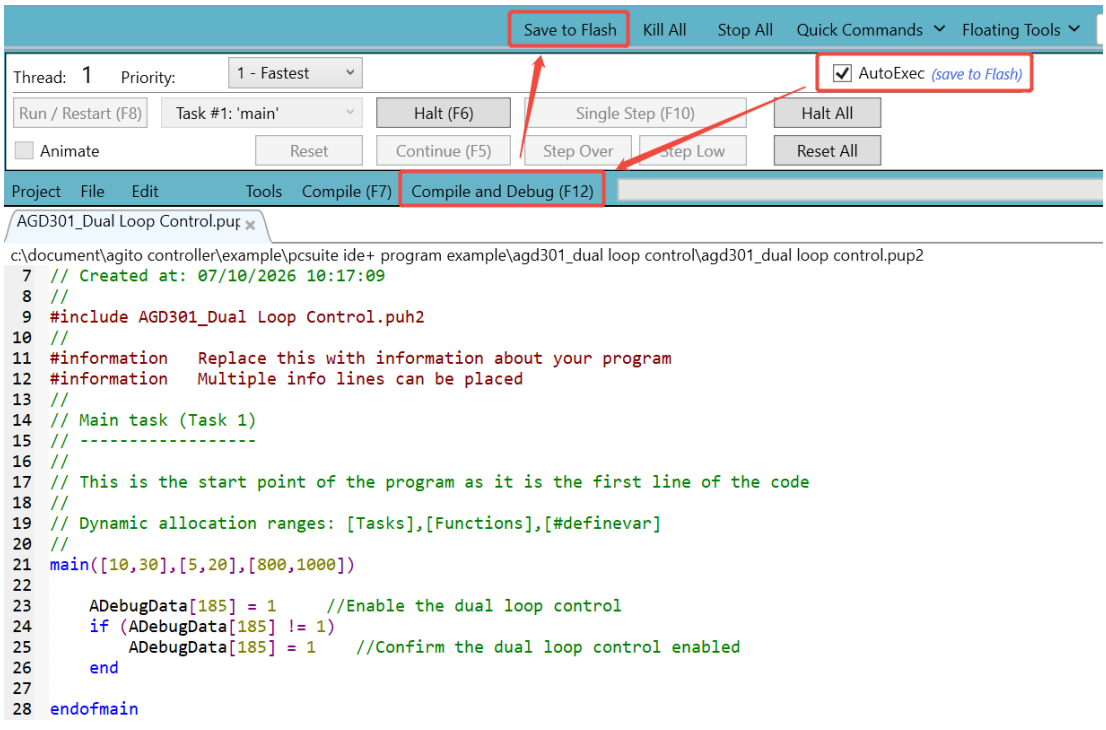
Step 8.

Once the above steps are complete, motion control can proceed.

Note

To automatically enable Dual Loop function at power-on in the IDE user-program:

Check “AutoExec” (execute automatically at power-on) –> “Compile and Debug” –> “Save to flash”



2.3 AGA101 and AGA155 Series

AGA101 and AGA155 are Central-I series amplifiers and must be used together with the AGM800 motion controller.

2.3.1 System Diagram

AGA101 system diagram:

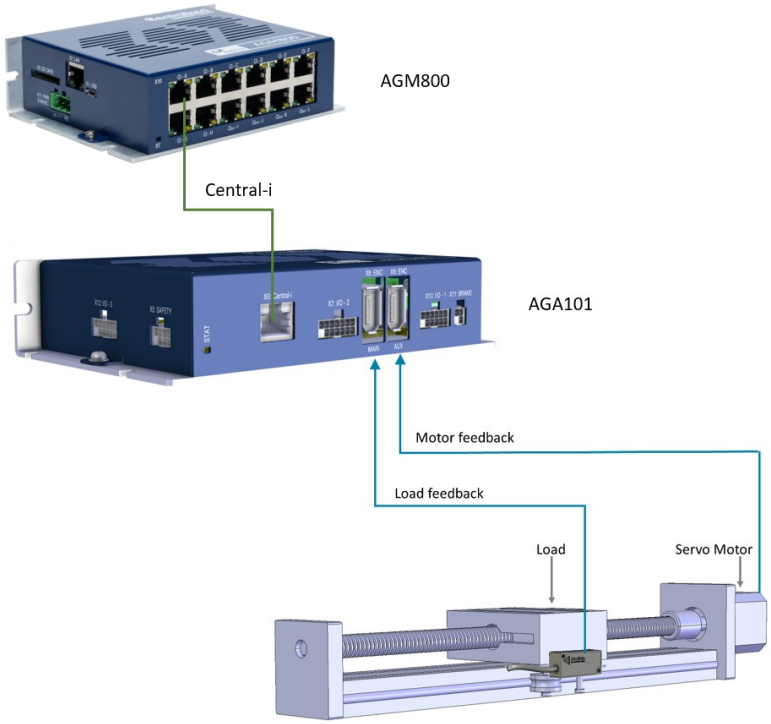


Figure 7. AGA101 Dual Loop Control System Diagram

The load encoder feedback connects to the main encoder port (X8: Main Enc.); the motor encoder feedback connects to the auxiliary encoder port (X9: Aux Enc.).

AGA155 system diagram:

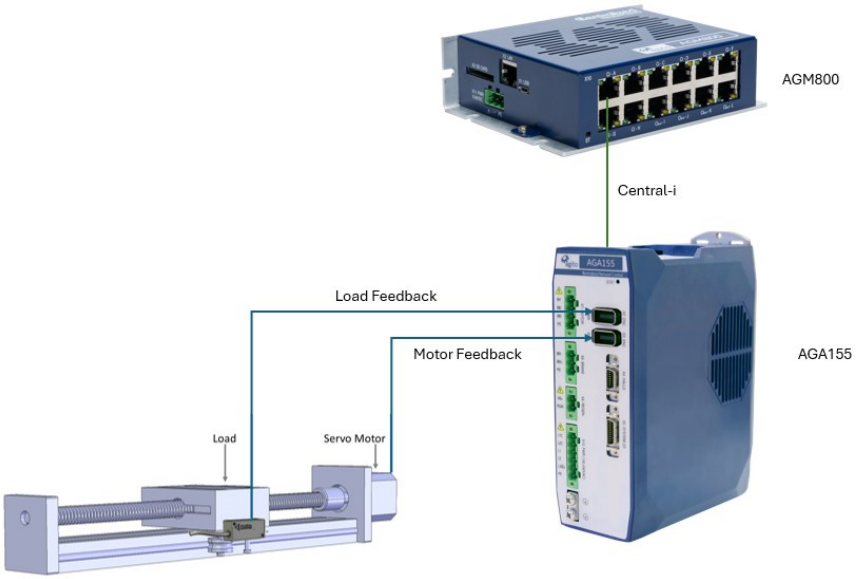


Figure 8. AGA155 Dual Loop Control System Diagram

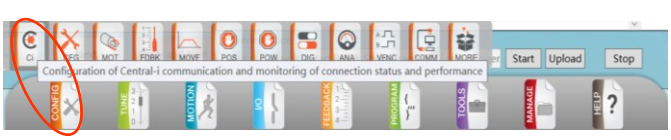
The load encoder feedback connects to the main encoder port (X2: Main Enc.); the motor encoder feedback connects to the auxiliary encoder port (X3: Aux Enc.).

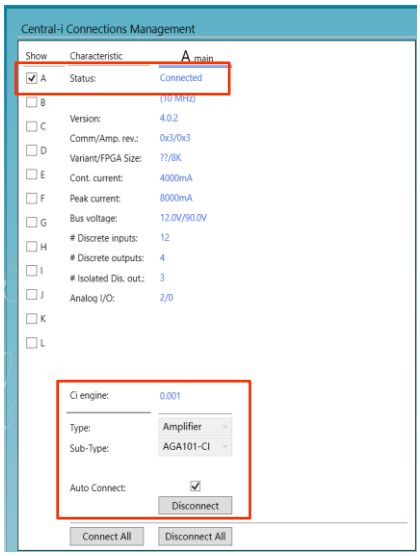
Note

Single-loop control must be performed before Dual Loop control.

2.3.2 Parameter Configuration

The Dual Loop parameter settings for the AGA101 and AGA155 series are exactly the same as for the AGD155. The only difference is that the connection must go through the AGM800, selecting the corresponding motion axis. For other steps, please refer to the AGD155 series parameter configuration procedure earlier in this document (chapter 2.1); it is not repeated here.





3 Keywords

Keyword	Description
DualLoopOn	Used to enable or disable Dual Loop mode. DualLoopOn=1, enables Dual Loop mode; the main encoder value (Pos) is used for position loop feedback. DualLoopOn=0, disables Dual Loop mode, using the default control mode; the main encoder feedback is used for both the position loop (Pos) and velocity loop feedback (Vel).
DualLoopFact	Scale factor. AuxVel multiplied by [DualLoopFact/65536] is used for velocity loop feedback.
DualStuckVel	Maximum allowed velocity error between the two encoders.
DualStuckTime	The controller reports an error when the velocity error exceeds the maximum error for longer than the set time.
Pos	Main encoder position feedback.
Vel	Vel[ ] is an array showing the main encoder feedback in three different ways: Vel[1] is the filtered velocity; Vel[2] is the raw velocity value; Vel[3] is the average velocity over 16 sampling periods.
AuxPos	Reads the auxiliary encoder position. The user can reset AuxPos to 0 while the motor is disabled.
AuxVel	Used to convey the auxiliary encoder velocity.
UsrUnits	Sets the display of the main encoder position in units other than Counts.
AuxUsrUnits	Sets the display of the auxiliary encoder position in units other than Counts.