Anomaly Detection

Application Note

www.agito-akribis.com
Member of Akribis Systems group

Revision History

Version	Description	Date
1.1	Updated the keyword description and UI image.	31 March 2026
1.0	Initial release	5 May 2025

Contact Information

Manufacturer	Agito Akribis Systems Ltd., Member of Akribis Systems Group
Address	56 Serangoon North Avenue 4, Singapore 555851
Email	info@akribis-sys.com
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.

Contents

1 About this Application Note

This application note explains how to configure the Anomaly Detection feature over PCSuite.

The following topics will be covered in this document, PCSuite configuration steps, settings related Anomaly Detection and relevant keywords summary.

1.1 Introduction to Anomaly Detection

The Anomaly Detection is a safety feature that allows the controller to detect collisions and bring the motor to a controlled stop.

The controller is first taught the time-based profile of a user defined signal under normal circumstances. Any controller parameter can be assigned as the signal; typically, the motor current, position error/velocity error is used.

Then, the user sets an allowable boundary region. During operation, the controller checks if the signal goes out of the boundary. If it does, the controller brings the motor to a controlled stop.

1.2 Pre-requisites

Before setting up the Anomaly Detection feature, it is necessary to install the payload and tune the motor. The motion profile and stroke will also have to be fixed.

Should the payload, tuning, or motion profile be changed, the anomaly setup will have to be done again.

This feature is only available on AGM800-CI and AGD155-EC products due to hardware limitations.

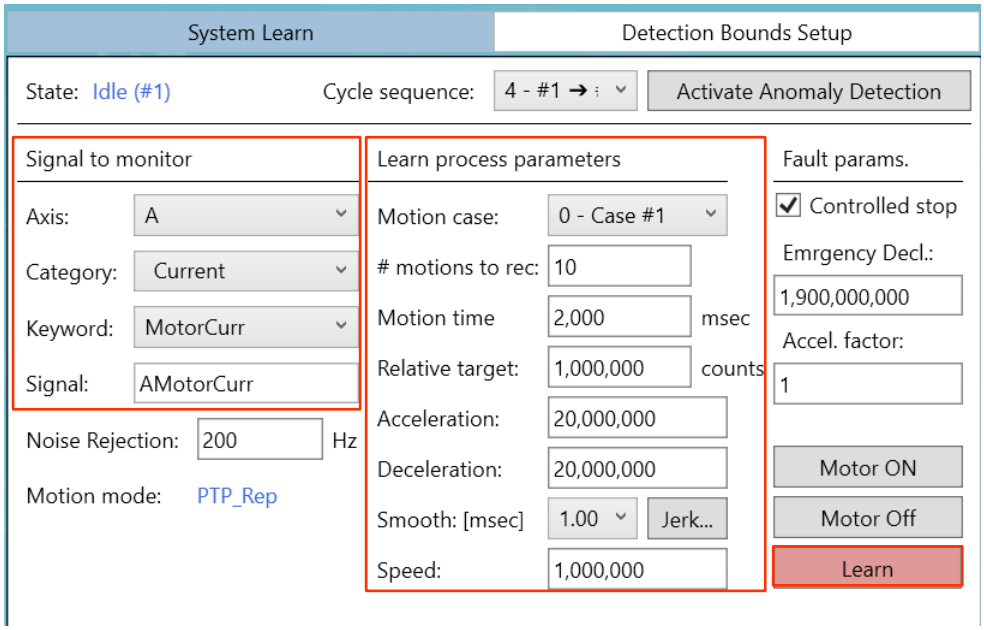
2 Configuration Setup (via PCSuite)

2.1 System Learn

Under 

, navigate to the 

page, [System Learn] tab.

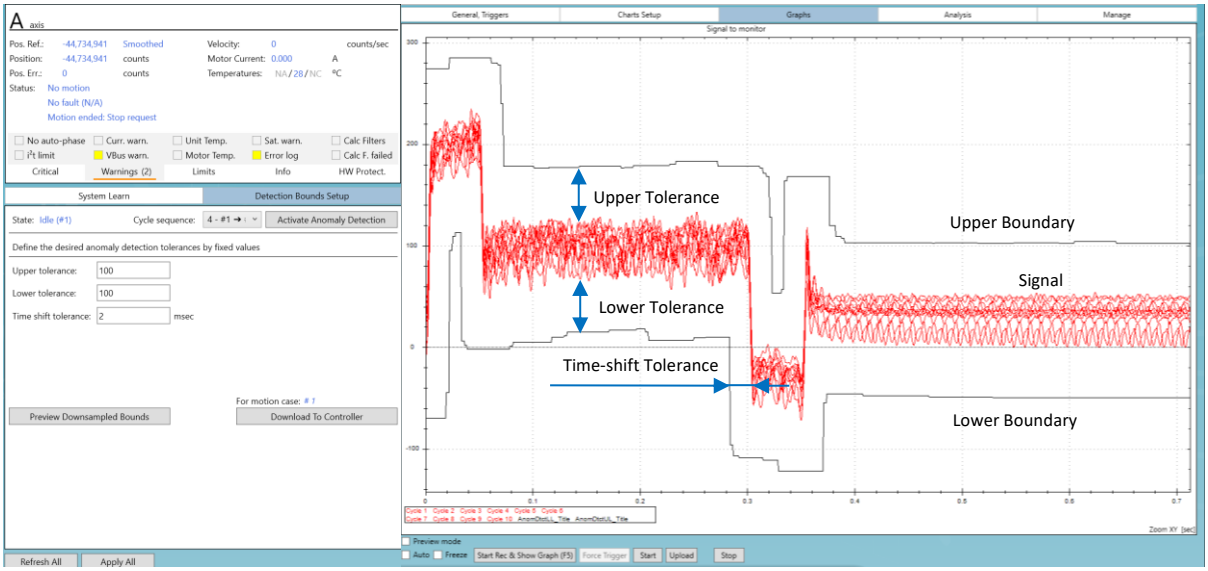


- Set the “Signal to monitor” to a parameter of your choice. For this example, we shall use MotorCurr. MotorCurr is corelated to the force, so it is a good indication to understand if the driver is applying more force than usual.
- Motion Case: Select the desired motion case to learn and download to controller (maximum quantity is 4 cases).
- #Motions to rec is the number of cycles to repeat and record the motion. Increase the number of cycles to account for variances such as friction, temperature.
- Motion time is the time to record each motion. This time needs to be longer than the motion time.
- Relative target is the displacement for each step.
- Acceleration/Deceleration/Speed/Smooth is the motion profile parameters.

Finally, click “Learn” and wait for the learn process to finish. Once done, PCSuite will display an overlay with your signal.

2.2 Detection Bounds Setup

Navigate to the [Detection Bounds Setup] tab. Click on the [Preview Downsampled Bounds] to generate a boundary.



- Upper tolerance is the spacing between the upper boundary and the maximum value of the various recorded cycles.
- Lower tolerance is the spacing between the lower boundary and the minimum value of the various recorded cycles.
- Time shift tolerance is the spacing between the upper/lower boundary in the time-axis.
- Scaling factor scales the upper limit by the factor and lower limits by the inverse of the factor.

Adjust the tolerances and click on [Preview Downsampled Bounds] to re-generate the boundaries.

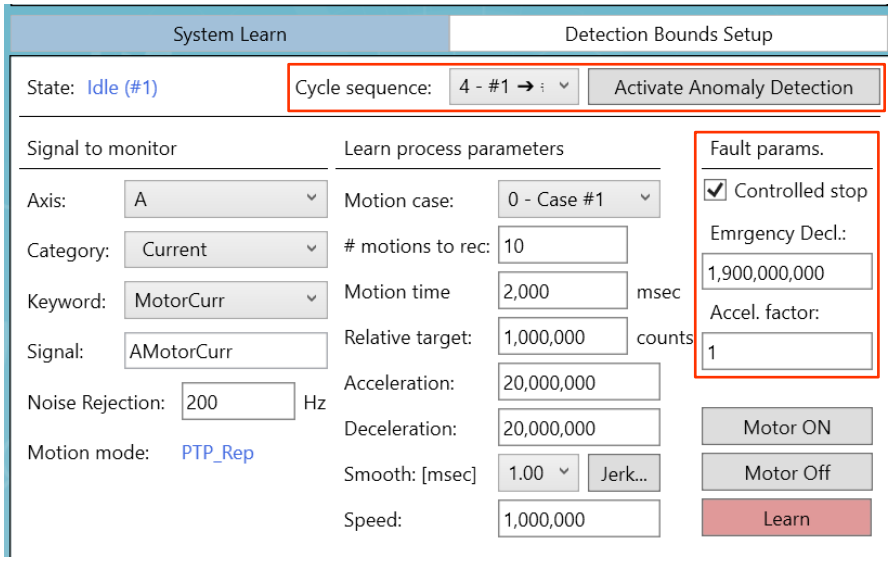
When satisfied, click on [Download To Controller] to load the boundary to the controller.

2.3 Enabling the feature

Navigate back to the [System Learn] tab.

Upon detection of an anomaly, the controller can either servo off immediately, or decelerate to a controlled stop before doing servo off. Check the box if a controlled stop is required.

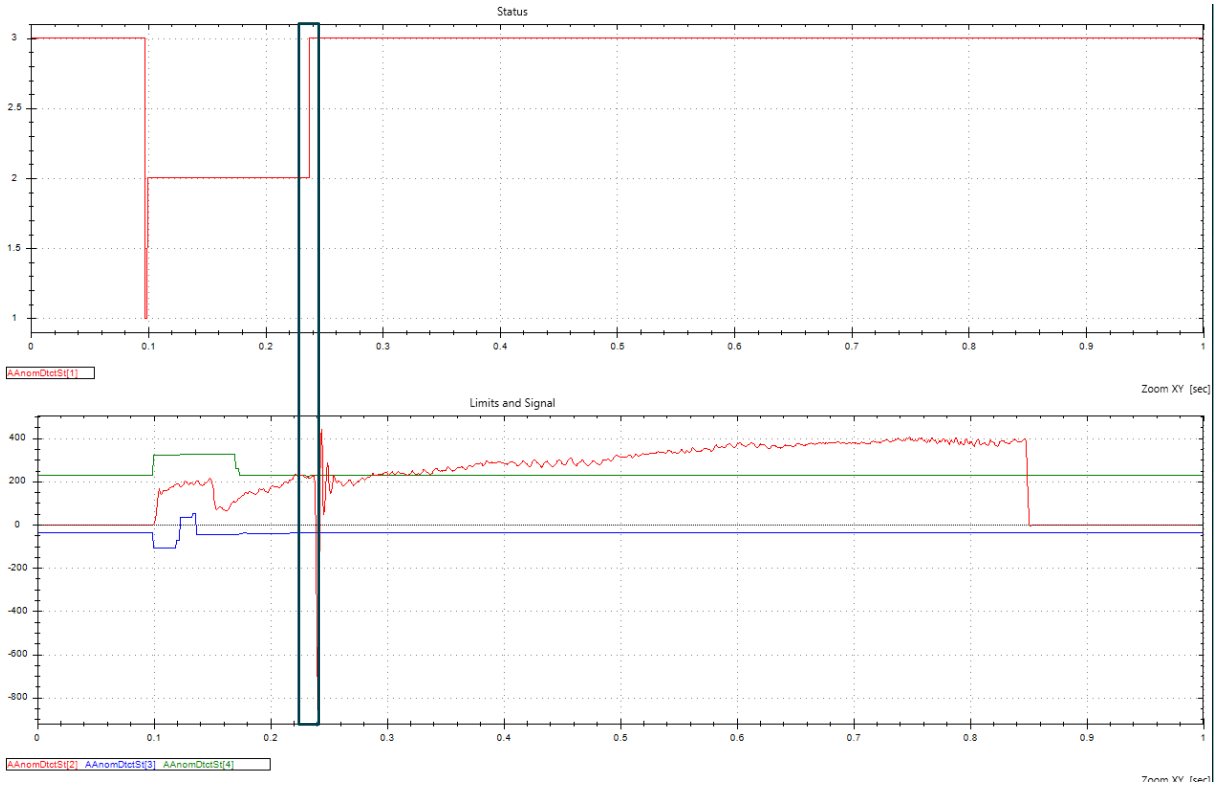
Choose the cycle sequence and click on the [Activate Anomaly Detection] to activate the feature.



2.4 Testing

Setup the data recorder to capture AnomDtctSt[1-4].

Make a motion while safely applying some disturbance force. Check that the motor does indeed report that an anomaly is detected and comes to a quick stop.



3 Anomaly Detection Relevant Keywords

Keyword	Description
AnomDtctCnfg[]	**AnomDtctCnfg[1]** selects the source signal to monitor. The source signal can be any signal that has a keyword and **AnomDtctCnfg[1]** should contain the CCC of that signal. **AnomDtctCnfg[2]** selects the cutoff frequency of the anomaly detection filter in Hz. The filter is a second order LPF, with fixed damping of 0.8. The source signal is filtered on-the-fly, and the filtered version is the one that is compared to the limits **AnomDtctUL** and **AnomDtctLL**. **AnomDtctCnfg[3]** selects the down-sampling gap used in converting the PCSuite UL and LL arrays to **AnomDtctUL** and **AnomDtctLL**. **AnomDtctCnfg[4]** selects the actual size used in **AnomDtctUL** and **AnomDtctLL**. **AnomDtctCnfg[5]** stores the expected **Speed**. **AnomDtctCnfg[6]** stores the expected **Accel**. **AnomDtctCnfg[7]** stores the expected **Decel**. **AnomDtctCnfg[8]** stores the expected **Jerk**. **AnomDtctCnfg[9]** stores the expected **RelTrgt**. **AnomDtctCnfg[10] = 1** sets the mode to **controlled stop first** upon fault. **AnomDtctCnfg[10] = 0** sets the mode to **motor off** upon fault. **AnomDtctCnfg[11]** defines the desired motion case to learn (maximum: 4 cases). **AnomDtctCnfg[12]** specifies the cycles execution order.
AnomDtctSt[]	**AnomDtctSt[1]** reflects the current state of Anomaly Detection feature. ▪ **0 = Idle (disabled by user)** ▪ **1 = Enabled the Anomaly Detection and waiting for motion** ▪ **2 = Active** ▪ **3 = Anomaly detected** because current motion parameters don't match the expected cycle. Anomaly detection will be switched back to "Active" when the machine returns to perform the expected cycle. **AnomDtctSt[2]** reflects the current value of the filtered signal. **AnomDtctSt[3]** reflects the current value of AnomDtctLL. **AnomDtctSt[4]** reflects the current value of AnomDtctUL.
AnomDtctUL[]	**AnomDtctUL[]** stores the downsampled upper limit.
AnomDtctLL[]	**AnomDtctLL[]** stores the downsampled lower limit.
AnomDtctOn	**AnomDtctOn = 1** enables the feature. **AnomDtctOn = 0** disables the feature.