



User Manual

NANOSCAN NPC-D-6xxx SERIES NANOMECHANISM CONTROLLER COMMAND SET AND CONTROL SYSTEM

Description of control system and command set
allowing configuration and control from
customer software

Safety Precautions

WARNINGS

HAZARDOUS VOLTAGES

The Product generates high voltages and relies on the provision of a protective earth (ground) conductor to prevent user accessible components developing a hazardous potential in the event of an insulation failure. This protective earth is provided by the external power supply and only an approved power supply should be used with the product. Additional protection is provided by special NanoMechanism interface connectors and cable assemblies. To maintain the integrity of the operator safety systems only approved NanoMechanisms and cables should be used with the product. The product should not be used if there are any signs of damage or if the equipment is believed to be faulty. It should be returned to the manufacturer for investigation and repair.



DO NOT remove the equipment's protect covers. There are no user serviceable parts within the equipment and removal of the cover will expose the user to potential high voltage hazards and will invalidate the Warranty.



Do read the manual before using the controller to understand how to correctly and safely operate the product. Incorrect use of the equipment may lead to personal injury or damage to property. Always turn the equipment off and remove the mains plug when not in use. Always use the equipment as specified in this manual.

CAUTIONS

ELECTROSTATIC SENSITIVE DEVICES (ESD)



The unit contains components that are susceptible to damage through electrostatic discharge at the NanoMechanism and interface connectors. Removal of protective connector covers and connection of cables should be performed in a static safe environment using approved static safety handling procedures.

ENVIRONMENT

The unit is designed for use in-doors in a dry environmentally controlled manufacturing facility, office or laboratory. The temperature and relative humidity should be kept within those specified in Table 2.1. Significant dust or acoustic/mechanical vibration may cause faulty operation or damage to components so should be avoided. Maintain adequate cooling of the controller by not restricting the air flow to and from the fan cooling vents. For prolonged periods of operation it is advisable to keep the environmental humidity to a minimum.



DIRECTIVE AND STANDARDS APPLIED:

2004/108/EC	EMC Directive BS EN 61326-1:2006 Electrical equipment for measurement, control and laboratory use EMC requirements — Part 1: General requirements FCC part 15, subpart B
2006/95/EC	Low Voltage Directive BS EN 61010-1:2010 Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements
2003/108/EC	WEEE Directive

Table of Contents

1	Introduction.....	19
2	System overview	20
2.1	Controller security	20
2.2	Controller and stage identification.....	20
2.3	Controller communications	20
2.4	Stage calibration presets	20
2.5	Control system.....	20
2.6	Resonance detection and frequency spectrum analysis.....	21
2.7	Position rebalancing.....	22
2.8	Analogue inputs, outputs and position command	22
2.9	Stepped inputs, outputs and position command	22
2.10	Function playback and waveform creation.....	22
2.11	Snapshot capture.....	23
2.12	Scope signals	23
2.13	Controller protection.....	24
2.14	Controller synchronisation	24
2.15	Controller operation with sensor-only systems	24
2.16	Diagnostics logging.....	24
3	Notation used in this document	25
3.1	Format of command descriptions.....	25
3.1.1	Command name	25
3.1.2	Description.....	25
3.1.3	Command	25
3.1.4	Parameters	25
3.1.5	Results.....	27
3.1.6	Possible error reports	27
3.1.7	Minimum security level	28
3.1.8	Supported in	28
3.2	Error reporting.....	28
3.2.1	Error result fields.....	28
3.2.2	Error code text	28
4	Controller security.....	30
4.1	Security access and security levels	30
4.1.1	controller.security.user.get/set	31
4.1.2	controller.security.lock	33
5	Controller and stage identification	34
5.1	Controller identification	34
5.1.1	identity.hardware.part.get	34

5.1.2	identity.hardware.serial.get	35
5.1.3	identity.hardware.mandate.get	36
5.1.4	identity.hardware.caldate.get	37
5.1.5	identity.hardware.bootloader-version.get	38
5.1.6	identity.hardware.platform-version.get	39
5.1.7	identity.software.version.get	40
5.1.8	identity.software.reldate.get	41
5.1.9	identity.software.part.get	42
5.1.10	controller.sampling-time.get	43
5.1.11	controller.channels.get	44
5.1.12	controller.status.get	45
5.2	Stage presence	46
5.2.1	stage.status.stage-connected.get	46
5.3	Stage identification	47
5.3.1	identity.stage.part.get	47
5.3.2	identity.stage.serial.get	48
5.3.3	identity.stage.axisid.get	49
5.3.4	identity.stage.mandate.get	50
5.3.5	identity.stage.caldate.get	51
6	Controller communications	52
6.1	USB and RS-232 communications	52
6.2	Ethernet communications	52
6.2.1	controller.tcpip-comms.ip-address.get/set	53
6.2.2	controller.tcpip-comms.tcp-port.get/set	54
7	Stage calibration presets	55
7.1	Stage presets	55
7.2	Other stage data	55
7.3	Stage calibration storage	56
7.4	Stage preset index	56
7.5	Stage calibration preset commands	58
7.5.1	stage.calibration.status.get	58
7.5.2	stage.calibration.preset.current.get	60
7.5.3	stage.calibration.preset.default.get	61
7.5.4	stage.calibration.preset.default.save	62
7.5.5	stage.calibration.preset.exists	63
7.5.6	stage.calibration.preset.is-factory.get	64
7.5.7	stage.calibration.preset.load	66
7.5.8	stage.calibration.preset.save	68
7.5.9	stage.calibration.preset.save-with-name	70
7.5.10	stage.calibration.preset.delete	72
7.5.11	stage.calibration.preset.name.get/set	74
7.5.12	stage.calibration.preset.configuration-id.get/set	76
8	Control system	78
8.1	Overview	78

8.2 Control system architecture	79
8.2.1 Control system top-level architecture.....	79
8.2.2 Command processing.....	80
8.2.3 Measured position processing	81
8.2.4 In position check.....	82
8.2.5 Control loop	83
8.2.6 Feedforward command trajectory control.....	84
8.2.7 Calculate velocity and acceleration.....	85
8.2.8 Open-loop control	86
8.2.9 Closed-loop control.....	87
8.2.10 Closed-loop position control.....	88
8.2.11 Closed-loop position and velocity control.....	88
8.2.12 Closed-loop position, velocity and acceleration control	88
8.2.13 Damping	89
8.2.14 Active damping	89
8.2.15 Actuator output	90
8.3 Command processing	91
8.3.1 stage.position.absolute-command.get/set.....	92
8.3.2 stage.range.closed-loop-command.minimum.get	94
8.3.3 stage.range.closed-loop-command.maximum.get	95
8.3.4 stage.position.command.get/set	96
8.3.5 stage.mode.analogue-command.get/set.....	98
8.3.6 stage.command.analogue.low-pass-filter.enable.get/set	100
8.3.7 stage.command.analogue.low-pass-filter.time-constant.get/set	102
8.3.8 stage.command.analogue.low-pass-filter.filter-order.get/set	104
8.3.9 stage.command.analogue.low-pass-filter.bode.gain.get.....	106
8.3.10 stage.command.analogue.low-pass-filter.bode.phase.get.....	107
8.3.11 stage.command.analogue.high-pass-filter.gain.get/set.....	108
8.3.12 stage.command.analogue.high-pass-filter.time-constant.get/set	110
8.3.13 stage.command.analogue.high-pass-filter.q.get/set	112
8.3.14 stage.command.analogue.high-pass-filter.bode.gain.get	114
8.3.15 stage.command.analogue.high-pass-filter.bode.phase.get	115
8.3.16 stage.command.digital.high-pass-filter.gain.get/set	116
8.3.17 stage.command.digital.high-pass-filter.time-constant.get/set	118
8.3.18 stage.command.digital.high-pass-filter.q.get/set.....	120
8.3.19 stage.command.digital.high-pass-filter.bode.gain.get.....	122
8.3.20 stage.command.digital.high-pass-filter.bode.phase.get.....	123
8.4 Measured position processing	124
8.4.1 stage.position.measured.get.....	124
8.4.2 stage.position.low-pass-filter.enable.get/set	125
8.4.3 stage.position.low-pass-filter.time-constant.get/set	127
8.4.4 stage.position.low-pass-filter.filter-order.get/set.....	129
8.4.5 stage.position.low-pass-filter.bode.gain.get	131
8.4.6 stage.position.low-pass-filter.bode.phase.get	132
8.4.7 stage.position.output.low-pass-filter.enable.get/set	133
8.4.8 stage.position.output.low-pass-filter.time-constant.get/set	135
8.4.9 stage.position.output.low-pass-filter.filter-order.get/set.....	137
8.4.10 stage.position.output.low-pass-filter.bode.gain.get	139
8.4.11 stage.position.output.low-pass-filter.bode.phase.get	140

8.4.12	stage.position.output.low-pass-filter.bode.combined-gain.get	141
8.4.13	stage.position.output.low-pass-filter.bode.combined-phase.get	142
8.5	In position check	143
8.5.1	stage.in-position.error-threshold.get/set	144
8.5.2	stage.in-position.lpf.time-constant.get/set.....	146
8.5.3	stage.in-position.window-filter.size.get/set.....	148
8.5.4	stage.in-position.window-filter.valid-threshold.get/set.....	150
8.5.5	stage.status.in-position.unconfirmed.get	152
8.5.6	stage.status.in-position.lpf-confirmed.get.....	153
8.5.7	stage.status.in-position.window-filter-confirmed.get	154
8.5.8	stage.mode.in-position-output-select.get/set	155
8.5.9	stage.status.stage-moving.get.....	157
8.6	Control loop	158
8.6.1	stage.mode.closed-loop.get/set	159
8.7	Feedforward command trajectory control.....	161
8.7.1	stage.command-trajectory.enable.get/set.....	162
8.7.2	stage.command-trajectory.launch-acceleration.get/set.....	164
8.7.3	stage.command-trajectory.speed.get/set.....	166
8.7.4	stage.command-trajectory.braking-deceleration.get/set	168
8.8	Calculate velocity and acceleration.....	170
8.8.1	stage.position.velocity-accel-filter.time-constant.get/set	171
8.8.2	stage.position.velocity-accel-filter.bode.gain.get.....	173
8.8.3	stage.position.velocity-accel-filter.bode.phase.get.....	174
8.8.4	stage.position.velocity-accel-filter.bode.velocity.combined-gain.get	175
8.8.5	stage.position.velocity-accel-filter.bode.velocity.combined-phase.get	176
8.8.6	stage.position.velocity-accel-filter.bode.accel.combined-gain.get.....	177
8.8.7	stage.position.velocity-accel-filter.bode.accel.combined-phase.get	178
8.9	Open-loop control	179
8.9.1	stage.open-loop.gain.get/set.....	180
8.9.2	stage.open-loop.offset.get/set.....	182
8.9.3	stage.open-loop.output-offset-gain.get	184
8.10	Closed-loop control	185
8.10.1	Closed-loop control design	185
8.10.2	Nested PIDs for velocity and acceleration control.....	185
8.10.3	stage.closed-loop.pid.control.get/set	187
8.10.4	stage.closed-loop.pid.position.proportional.gain.get/set	189
8.10.5	stage.closed-loop.pid.position.proportional.set-point-weighting.get/set	191
8.10.6	stage.closed-loop.pid.position.integral.time-constant.get/set.....	193
8.10.7	stage.closed-loop.pid.position.differential.gain.get/set.....	195
8.10.8	stage.closed-loop.pid.position.feedforward.gain.get/set	197
8.10.9	stage.closed-loop.pid.velocity.proportional.gain.get/set.....	199
8.10.10	stage.closed-loop.pid.velocity.proportional.set-point-weighting.get/set	201
8.10.11	stage.closed-loop.pid.velocity.integral.time-constant.get/set	203
8.10.12	stage.closed-loop.pid.velocity.differential.gain.get/set	205
8.10.13	stage.closed-loop.pid.velocity.feedforward.gain.get/set	207
8.10.14	stage.closed-loop.pid.accel.proportional.gain.get/set	209
8.10.15	stage.closed-loop.pid.accel.proportional.set-point-weighting.get/set	211
8.10.16	stage.closed-loop.pid.accel.integral.time-constant.get/set.....	213

8.10.17	stage.closed-loop.pid.accel.feedforward.gain.get/set	215
8.11	Damping	217
8.11.1	stage.notch-filter.filter-type.get/set.....	218
8.11.2	stage.notch-filter.time-constant.get/set	220
8.11.3	stage.notch-filter.q.get/set.....	222
8.11.4	stage.notch-filter.second-filter-time-constant.get/set	224
8.11.5	stage.notch-filter.second-filter-q.get/set.....	226
8.11.6	stage.notch-filter.bode.gain.get.....	228
8.11.7	stage.notch-filter.bode.phase.get.....	229
8.11.8	stage.active-damping.enable.get/set	230
8.11.9	stage.active-damping.velocity.gain.get/set	232
8.11.10	stage.active-damping.velocity.deadband.get/set	234
8.12	Actuator output	236
8.12.1	stage.mode.freeze-servo-output.get/set	237
8.12.2	stage.status.servo-output-at-limits.get	239
9	Closed-loop control system calibration	240
9.1	Overview.....	240
9.2	Initial position PID loop calibration	240
9.3	Set position, velocity and acceleration filters	240
9.4	Configure damping and fit load	241
9.5	Tune position PID+FF loop	242
9.6	Prepare velocity PID+FF loop	242
9.7	Tune velocity PID+FF loop.....	243
9.8	Prepare acceleration PI+FF loop	243
9.9	Tune acceleration PI+FF loop	244
9.10	Tune command trajectory control.....	244
9.11	Iterate control loop tunings.....	245
10	Resonance detection and frequency spectrum analysis.....	246
10.1	Overview.....	246
10.2	Resonance detection protective shutdown	247
10.2.1	resonance-detect.protective-shutdown.state.get	248
10.2.2	resonance-detect.protective-shutdown.clear	249
10.3	Frequency spectrum analysis	250
10.3.1	resonance-detect.frequency-range.maximum.get.....	251
10.3.2	resonance-detect.frequency-range.resolution.get.....	252
10.4	Protective shutdown frequency spectrum capture	253
10.4.1	resonance-detect.protective-shutdown.capture.status.get.....	254
10.4.2	resonance-detect.protective-shutdown.capture.peak-error.frequency.get	255
10.4.3	resonance-detect.protective-shutdown.capture.peak-error.magnitude.get	256
10.4.4	resonance-detect.protective-shutdown.capture.measured-position.magnitude.get ..	257
10.4.5	resonance-detect.protective-shutdown.capture.commanded-position.magnitude.get ..	259
10.4.6	resonance-detect.protective-shutdown.capture.error.magnitude.get ..	261
10.5	Protective shutdown configuration	263

10.5.1	resonance-detect.protective-shutdown.enable.get/set.....	264
10.5.2	resonance-detect.protective-shutdown.error-threshold.get/set.....	266
10.5.3	resonance-detect.protective-shutdown.error-time.get/set.....	268
10.5.4	resonance-detect.protective-shutdown.frequency-range.minimum.get/set.....	270
10.5.5	resonance-detect.protective-shutdown.frequency-range.maximum.get/set.....	272
10.6	Frequency spectrum snapshot capture	274
10.6.1	resonance-detect.capture.trigger.....	275
10.6.2	resonance-detect.capture.status.get.....	276
10.6.3	resonance-detect.capture.peak-error.frequency.get	277
10.6.4	resonance-detect.capture.peak-error.magnitude.get	278
10.6.5	resonance-detect.capture.measured-position.magnitude.get	279
10.6.6	resonance-detect.capture.commanded-position.magnitude.get	281
10.6.7	resonance-detect.capture.error.magnitude.get	283
11	Position rebalancing for closed-loop range.....	285
11.1	Rebalancing design	285
11.2	Commanding rebalancing	286
11.2.1	stage.range.auto-balance.select-on-startup.get/set.....	286
11.2.2	stage.range.auto-balance.trigger.....	288
11.2.3	stage.range.auto-balance.in-progress.get	290
11.3	Stage range	291
11.3.1	stage.range.closed-loop.minimum.get	291
11.3.2	stage.range.closed-loop.maximum.get	292
11.3.3	stage.range.closed-loop.maximum.get	293
11.3.4	stage.range.closed-loop.range.get	294
11.3.5	stage.range.open-loop.range.get.....	295
11.3.6	stage.range.auto-balance.offset.get	296
12	Analogue inputs, outputs and position command	297
12.1	Analogue inputs	297
12.1.1	Overview.....	297
12.1.2	Controllers with dedicated analogue inputs	297
12.1.3	Controllers with switchable analogue inputs	298
12.1.4	controller.analogue-inputs.get.....	299
12.1.5	stage.mode.analogue-command.input-invert.get/set	300
12.1.6	stage.mode.analogue-command.input-2-invert.get/set.....	302
12.1.7	stage.mode.analogue-command.input-select.get/set	304
12.1.8	stage.command.analogue.scaling.get	306
12.2	Analogue outputs	307
13	Stepped inputs, outputs and position command	308
13.1	Stepped inputs	308
13.1.1	stage.stepped.input.enable.get/set	310
13.1.2	stage.stepped.input.is-quadrature.get/set.....	312
13.1.3	stage.stepped.input.reverse-direction.get/set	314
13.1.4	stage.stepped.input.debounce-time.get/set	316
13.1.5	stage.stepped.input.step-direction.is-rising-edge.get/set	318
13.2	Stepped position command	320

13.2.1	stage.position.stepped-command.get/set	321
13.2.2	stage.position.stepped-command.increment	323
13.2.3	stage.position.stepped-command.steps.get/set.....	324
13.2.4	stage.position.stepped-command.steps.increment.....	326
13.2.5	stage.position.stepped-command.step-size.get/set	327
13.3	Stepped outputs.....	329
13.3.1	stage.stepped.output.select.get/set	331
13.3.2	stage.stepped.output.is-quadrature.get/set.....	334
13.3.3	stage.stepped.output.reverse-direction.get/set.....	336
13.3.4	stage.stepped.output.step-direction.is-rising-edge.get/set.....	338
13.3.5	stage.stepped.output.step-size.get/set	340
13.3.6	stage.stepped.output.send-full-value.get/set	342
13.3.7	stage.stepped.output.quadrature.hold-time.get/set.....	344
13.3.8	stage.stepped.output.step-direction.settle-time.get/set.....	346
13.3.9	stage.stepped.output.step-direction.hold-time.get/set	348
14	Function playback	350
14.1	Overview.....	350
14.2	Waveform configuration using sampled data points.....	352
14.2.1	Waveform entry	352
14.2.2	Ramped or stepped position command with longer sample periods.....	352
14.2.3	function.waveform.data.get/set	353
14.2.4	function.waveform.count.get/set	355
14.2.5	function.waveform.sample-period.get/set	357
14.2.6	function.waveform.command-transition.get/set.....	359
14.2.7	function.waveform.soft-stop-at-end.get/set.....	361
14.3	Waveform design with repeated sections	363
14.3.1	function.waveform.repeat-start.get/set.....	364
14.3.2	function.waveform.repeat-end.get/set.....	366
14.3.3	function.waveform.repeat-count.get/set.....	368
14.4	Function playback control from host PC	370
14.4.1	function.state.get	371
14.4.2	function.command.start	373
14.4.3	function.command.stop	375
14.4.4	function.command.soft-stop.....	377
14.4.5	function.command.pause.....	379
14.4.6	function.command.unpause.....	381
14.5	Function playback control from digital trigger inputs	383
14.5.1	function.trigger-inputs.start.get/set.....	384
14.5.2	function.trigger-inputs.stop.get/set.....	386
14.5.3	function.trigger-inputs.soft-stop.get/set.....	388
14.5.4	function.trigger-inputs.pause.get/set.....	390
14.5.5	function.trigger-inputs.unpause.get/set.....	392
14.5.6	snapshot.trigger-inputs.start.get/set.....	394
14.5.7	snapshot.trigger-inputs.stop.get/set.....	396
14.5.8	snapshot.trigger-inputs.start-delay.get/set.....	398
14.5.9	function.trigger-inputs.enabled.get.....	399
14.5.10	function.trigger-inputs.enabled.enable-inputs	401

14.5.11	function.trigger-inputs.enabled.disable-inputs.....	403
14.6	Control of external equipment with digital trigger outputs	405
14.6.1	Overview.....	405
14.6.2	Trigger output event types	405
14.6.3	Factors to consider with position-based triggering.....	406
14.6.4	Trigger events at start or end of sample period	407
14.6.5	Trigger output pulse duration	407
14.6.6	Factors to consider with trigger timings.....	407
14.6.7	Manual control of trigger output	408
14.6.8	function.waveform.trigger-out-event.get/set.....	409
14.6.9	function.waveform.trigger-at-sample-end.get/set.....	411
14.6.10	function.trigger-output.pulse-time.get/set.....	413
14.6.11	function.trigger-output.trigger.....	415
15	Waveform generator.....	416
15.1	Overview.....	416
15.2	Waveform generator segment configuration commands.....	417
15.2.1	function.waveform-generator.clear	418
15.2.2	function.waveform-generator.segment.type.get/set	419
15.2.3	function.waveform-generator.segment.parameter.get/set	421
15.2.4	function.waveform-generator.segment.continue-position-velocity.get/set.....	423
15.2.5	function.waveform-generator.count.get/set.....	426
15.2.6	function.waveform-generator.soft-stop-at-end.get/set	428
15.3	Waveform generator segment function types	430
15.3.1	Function configuration	430
15.3.2	Rounding errors with large sample periods.....	431
15.3.3	Function types supported in firmware versions.....	431
15.3.4	Basic function types.....	432
15.3.5	Sine-wave profile S-curves for acceleration and deceleration	442
15.3.6	Linear profile S-curves for acceleration and deceleration	457
15.3.7	Steps using sine-wave profile S-curves	471
15.3.8	Steps using linear profile S-curves	478
15.3.9	"Staircase" repeated steps.....	484
15.4	Waveform design example	500
15.5	Waveform design with repeating waveforms	504
15.5.1	function.waveform-generator.repeat-start.get/set	505
15.5.2	function.waveform-generator.repeat-end.get/set	507
15.5.3	function.waveform-generator.repeat-count.get/set	509
15.6	Waveforms with synchronised digital trigger outputs	511
15.6.1	Trigger points within waveform	511
15.6.2	Trigger output event types	511
15.6.3	Trigger output offset times	512
15.6.4	Multiple triggers at irregular intervals	513
15.6.5	function.waveform-generator.trigger-output.start-trigger.get/set	514
15.6.6	function.waveform-generator.trigger-output.end-trigger.get/set	516
15.6.7	function.waveform-generator.segment.trigger-output.start-trigger.get/set	518
15.6.8	function.waveform-generator.segment.trigger-output.start-trigger-offset.get/set	520
15.6.9	function.waveform-generator.segment.trigger-output.end-trigger.get/set	522

15.6.10	function.waveform-generator.segment.trigger-output.end-trigger-offset.get/set	524
15.6.11	function.waveform-generator.segment.trigger-output.during-trigger.get/set	526
15.6.12	function.waveform-generator.segment.trigger-output.during-trigger-offset.get/set	528
15.6.13	function.waveform-generator.segment.trigger-output.during-trigger-count.get/set	530
15.7	Waveform check, waveform preparation and sampled data points	532
15.7.1	Waveform generation and sample rates	532
15.7.2	Waveform generator check	532
15.7.3	Waveform generator error reporting	533
15.7.4	Waveform generator sampled data point preparation	535
15.7.5	function.waveform-generator.sample-period.get/set	536
15.7.6	function.waveform-generator.check-waveform	538
15.7.7	function.waveform-generator.prepare-waveform	539
15.7.8	function.waveform-generator.prepare-start	540
15.7.9	function.waveform-generator.prepare-waveform-status.get	542
15.7.10	function.waveform-generator.failed-at-segment-index.get	544
15.7.11	function.waveform-generator.failure-cause.get	545
15.7.12	function.waveform-generator.waveform-duration.get	546
15.7.13	function.waveform-generator.segment.duration.get	547
15.7.14	function.waveform-generator.segment.start-position.get	548
15.7.15	function.waveform-generator.segment.start-velocity.get	549
15.7.16	function.waveform-generator.segment.end-position.get	551
15.7.17	function.waveform-generator.segment.end-velocity.get	552
16	Waveform builder	554
16.1	Overview	554
16.2	Waveform builder and sample rates	554
16.3	Staircase-step waveform builder	555
16.3.1	Overview	555
16.3.2	Waveform configurations	555
16.3.3	Trigger outputs	559
16.3.4	Check and prepare waveform	559
16.3.5	function.waveform-builder.staircase.clear	561
16.3.6	function.waveform-builder.staircase.check	562
16.3.7	function.waveform-builder.staircase.prepare	563
16.3.8	function.waveform-builder.staircase.prepare-start	564
16.3.9	function.waveform-builder.staircase.is-bidirectional.get/set	565
16.3.10	function.waveform-builder.staircase.steps.get/set	567
16.3.11	function.waveform-builder.staircase.step-distance.get/set	569
16.3.12	function.waveform-builder.staircase.step-duration.get/set	571
16.3.13	function.waveform-builder.staircase.hold-duration.get/set	573
16.3.14	function.waveform-builder.staircase.start-position.get/set	575
16.3.15	function.waveform-builder.staircase.start-step-duration.get/set	577
16.3.16	function.waveform-builder.staircase.start-hold-duration.get/set	579
16.3.17	function.waveform-builder.staircase.return-step-duration.get/set	581
16.3.18	function.waveform-builder.staircase.return-hold-duration.get/set	583
16.3.19	function.waveform-builder.staircase.repeat-count.get/set	585
16.3.20	function.waveform-builder.staircase.trigger-output.trigger.get/set	587
16.3.21	function.waveform-builder.staircase.trigger-output.delay.get/set	589

17	Snapshot capture	591
17.1	Snapshot configuration and control.....	591
17.1.1	snapshot.fire	592
17.1.2	snapshot.stop	593
17.1.3	snapshot.capture.count.get/set.....	594
17.1.4	snapshot.trigger.to-target.get/set.....	595
17.1.5	snapshot.trigger.from-target.get/set.....	597
17.1.6	snapshot.trigger.step.get/set.....	599
17.2	Snapshot results	601
17.2.1	snapshot.response.data.get.....	601
17.2.2	snapshot.response.count.get.....	603
17.2.3	snapshot.response.data-select.get.....	604
18	Scope monitoring of internal data	605
18.1	Scope measurement and routing	605
18.1.1	scope.data-select.get/set.....	607
18.1.2	scope.measurement.get	609
18.1.3	scope.routing.to-snapshot.get/set.....	610
18.1.4	scope.routing.to-output.get/set	612
18.1.5	scope.routing.output-scaling.gain.get/set	614
18.1.6	scope.routing.output-scaling.offset.get/set	616
18.2	Scope data signal selection	618
19	Controller protection	622
19.1	Overview.....	622
19.1.1	Automatic fan control	622
19.1.2	Controller thermal (overtemperature) protection	622
19.1.3	Stage short-circuit (overcurrent) protection.....	623
19.1.4	Controller actions during and after protective shutdown	623
19.2	Automatic fan control	624
19.2.1	protection.fan.mode.get/set	624
19.2.2	protection.fan.state.get	626
19.2.3	protection.fan.psu.on-temperature.get/set	627
19.2.4	protection.fan.psu.off-temperature.get/set	628
19.2.5	protection.fan.heatsink.on-temperature.get/set.....	629
19.2.6	protection.fan.heatsink.off-temperature.get/set.....	630
19.3	Controller thermal (overtemperature) protection	631
19.3.1	protection.thermal.status.get.....	631
19.3.2	protection.thermal.heatsink.temperature.get.....	632
19.3.3	protection.thermal.heatsink.overtemperature-detect-threshold.get.....	633
19.3.4	protection.thermal.heatsink.overtemperature-detect-time.get.....	634
19.3.5	protection.thermal.heatsink.overtemperature-clear-threshold.get.....	635
19.3.6	protection.thermal.heatsink.overtemperature-clear-time.get.....	636
19.3.7	protection.thermal.psu.temperature.get	637
19.3.8	protection.thermal.psu.overtemperature-detect-threshold.get	638
19.3.9	protection.thermal.psu.overtemperature-detect-time.get	639
19.3.10	protection.thermal.psu.overtemperature-clear-threshold.get	640

19.3.11	protection.thermal.psu.overtemperature-clear-time.get	641
19.4	Stage short-circuit (overcurrent) protection	642
19.4.1	protection.overcurrent.status.get	642
19.4.2	protection.overcurrent.current.get	643
19.4.3	protection.overcurrent.detect-threshold.get	644
19.4.4	protection.overcurrent.detect-time.get	645
20	Controller synchronisation	646
20.1	Controller sensor synchronisation	646
20.1.1	controller.synchronisation.master.get	647
20.1.2	controller.synchronisation.slave.get	648
21	Controller operation with sensor-only systems	649
21.1	Capacitive sensor systems	649
21.2	Front-panel LEDs for sensor systems	649
21.3	Capacitive sensor startup	649
21.4	Sensor-only commands	651
21.4.1	controller.sensor-only.get	651
21.4.2	stage.mode.is-sensor-only.get	652
21.4.3	stage.position.measured.is-in-calibrated-range.get	653
21.4.4	stage.position.measured.is-readable.get	654
21.4.5	stage.position.calibrated-range.minimum.get	655
21.4.6	stage.position.calibrated-range.maximum.get	656
21.4.7	stage.position.calibrated-range.range.get	657
22	Diagnostics logging	658
22.1	Overview	658
22.2	Controller diagnostics	659
22.2.1	diagnostics-logging.controller.since-last-service.power-on.time.get	659
22.2.2	diagnostics-logging.controller.since-last-service.power-on.total.get	660
22.2.3	diagnostics-logging.controller.since-last-service.fan-on.time.get	661
22.2.4	diagnostics-logging.controller.since-last-service.fan-on.total.get	662
22.2.5	diagnostics-logging.controller.since-last-service.shutdowns.total.get	663
22.2.6	diagnostics-logging.controller.since-last-service.current.max-positive.get	664
22.2.7	diagnostics-logging.controller.since-last-service.current.max-negative.get	665
22.2.8	diagnostics-logging.controller.since-last-service.temperature.max.get	666
22.2.9	diagnostics-logging.controller.this-power-on.power-on.time.get	667
22.2.10	diagnostics-logging.controller.this-power-on.power-on.total.get	668
22.2.11	diagnostics-logging.controller.this-power-on.fan-on.time.get	669
22.2.12	diagnostics-logging.controller.this-power-on.fan-on.total.get	670
22.2.13	diagnostics-logging.controller.this-power-on.shutdowns.total.get	671
22.2.14	diagnostics-logging.controller.this-power-on.current.max-positive.get	672
22.2.15	diagnostics-logging.controller.this-power-on.current.max-negative.get	673
22.2.16	diagnostics-logging.controller.this-power-on.temperature.max.get	674
22.3	Stage diagnostics	675
22.3.1	diagnostics-logging.stage.since-last-service.power-on.time.get	675
22.3.2	diagnostics-logging.stage.since-last-service.power-on.total.get	676

22.3.3	diagnostics-logging.stage.since-last-service.shutdowns.total.get	677
22.3.4	diagnostics-logging.stage.since-last-service.current.max-positive.get.....	678
22.3.5	diagnostics-logging.stage.since-last-service.current.max-negative.get	679
23	Deprecated commands	680
23.1	Command processing.....	680
23.1.1	stage.mode.digital-command.get/set	680
23.2	Measured position processing	682
23.2.1	stage.position.low-pass-filter.filter-location.get/set.....	682
23.2.2	stage.position.low-pass-filter.q.get/set.....	684
23.4	Control loop configuration	686
23.4.1	stage.mode.get	686
23.4.2	stage.mode-only.get/set	688
23.4.3	stage.mode-mask.set	691
23.4.4	stage.mode.analogue-command.enable/disable	694
23.4.5	stage.mode.digital-command.enable/disable.....	696
23.4.6	stage.mode.closed-loop.enable/disable	698
23.4.7	stage.mode.freeze-servo-output.enable/disable	700
23.4.8	stage.mode.in-position-output-select.lpf/window-filter	702
23.5	Closed-loop control	704
23.5.1	stage.pid.proportional.gain.get/set.....	704
23.5.2	stage.pid.proportional.set-point-weighting.get/set.....	706
23.5.3	stage.pid.integral.time-constant.get/set	708
23.5.4	stage.pid.integral.error-magnitude.max.get/set.....	710
23.5.5	stage.pid.differential.gain.get/set	712
23.5.6	stage.pid.differential.time-constant.get/set.....	714
23.5.7	stage.pid.feedforward.gain.get/set.....	716
23.6	Damping	718
23.6.1	stage.notch-filter.filter-location.get/set	718
23.7	Active damping	721
23.7.1	stage.active-damping.accel.gain.get/set.....	721
23.7.2	stage.active-damping.accel.deadband.get/set.....	723
23.8	Function playback configuration	725
23.8.1	function.waveform.steps-per-sample.get/set	725
23.8.2	function.waveform.steps-per-trigger-out-pulse.get/set.....	727
23.8.3	function.waveform.waveform-start.get/set	729
23.8.4	function.waveform.waveform-end.get/set	731
23.8.5	function.waveform.iterations.get/set	733

Related documents

Document Ref	Title	Usage
EN-014217-UM	NanoScan NPC-D Series NanoMechanism Controller - Controller Interface Library	Details the usage of controller interface DLL in order to send commands to the controller
EN-014635-UM	NanoScan NPC-D-6xx0 NanoMechanism Controller - User Manual	User manual for the 6000 series controller

Revision history

Revision	Changes
1.0	First version.
2.0	<p>General: Reformatted for Prior Scientific and NanoScan rebranding.</p> <p>General: Added revision history.</p> <p>General: Added “Command” section for every command, to make it clearer how to construct command strings.</p> <p>General: Added “Supported in” section for every command, so that this document remains relevant for older versions of firmware and controller interface library software.</p> <p>General: Renamed “Security” section for every command to “Minimum security level”, to make this clear.</p> <p>2, “System overview”: Substantial rework.</p> <p>4.2, “Controller protection”: Added details of fan control and associated commands for firmware application 6.4.1 and controller interface 2.4.1.</p> <p>5.1, “Control system architecture”: Substantial rework to diagrams, leading to some commands changing locations within the document.</p> <p>5.2.1, “stage.position.absolute-command.get/set”: Added for firmware application 6.2.12 and controller interface 2.2.7.</p> <p>6, “Stepped inputs, outputs and position command”: Added for firmware application 6.2.12 and controller interface 2.2.7.</p> <p>7, “Function playback”: Substantial rework to add details of waveform generator and associated commands for firmware application 6.4.1 and controller interface 2.4.1, and to integrate this with sample-based waveform. Some changes to sample-based commands for commonality with waveform generator. Waveform generator is now considered the main method to use when setting up function playback, and is documented as such.</p>

Revision	Changes
	<p>7, “Function playback”: Added “in-position” trigger output type for firmware application 6.4.1 and controller interface 2.4.1.</p> <p>9, “Scope data signal selection”: Updated to support new signals available.</p> <p>11, “Deprecated commands”: Deprecated some function playback commands for sample-based waveform configuration. These have been superseded by equivalent waveform-generator-style commands.</p> <p>11, “Deprecated commands”: Added details of which versions of firmware and controller interface library software these commands were deprecated in, and which commands have replaced them.</p>
3.0	Internal use only
4.0	<p>General: Changed controller naming from “6xx0” to “6xxx”, since we now have the NPC-D-6111 controller.</p> <p>2, “System overview”: Moved “Stage calibration presets” section to the front and reworded. Added overviews for analogue I/O and diagnostics logging.</p> <p>4.1, “Controller system”: Added details of existing command “controller.channels.get” which seems to have been missed in previous versions of the document.</p> <p>5, “Stage calibration presets”: Moved from section 10 to here. Customers who do not want/need to configure the control loop and only need to load presets are now presented with this near the start of the document. Reworked description, including to cover preset naming. Added extra section to mention that more data is stored to the stage than just preset.</p> <p>6, “Control system”: Changes for new analogue input selection feature.</p> <p>7, “Analogue inputs, outputs and position command”: Added section to describe analogue interfacing more fully. Includes details of new features to allow analogue input selection and direction invert for microscopy customers.</p> <p>12, “Diagnostics logging”: Added section for new diagnostics logging feature.</p>
4.1	<p>Title page and back page: Minor changes for document style.</p> <p>2, “System overview”: Added overview for position rebalancing.</p> <p>7, “Position rebalancing for closed-loop range”: Added section for new position rebalancing feature.</p>
5.0	<p>General: Added new commands for firmware version 6.6.</p> <p>2, “System overview”: Reworked “Control system” section for new control loop features. Revised other sections for document layout changes covered below.</p>

Revision	Changes
	<p>3, "Notation": Added details of standard units used.</p> <p>4, "Controller security": Moved this to a separate section for clarity.</p> <p>5, "Controller and stage identification": Minor rework for clarity.</p> <p>6, "Stage calibration presets": Deleted duplicate listing for command "stage.calibration.preset.exists".</p> <p>7, "Control system": Added overview. Substantial rework to diagrams for new control loop features, leading to some commands changing locations within the document. New commands added for position monitor low-pass filters, closed-loop control of position/velocity/acceleration, notch filter, active damping. Moved "status" and "mode" commands to sub-sections relating to where they are used. Added text describing each control system block in the sub-section which describes the commands for that block. Added references from each diagram to the section with commands and description for that block.</p> <p>7.3, "Command processing": Added new filter commands.</p> <p>7.4, "Measured position processing": Updated commands to match new control loop.</p> <p>8, "Closed-loop control system calibration": Added new section to assist users with calibrating the new control loop.</p> <p>10, "Analogue inputs, outputs and position command": Slightly reworked diagrams and scope points to be more generally useful.</p> <p>12, "Function playback": Restructured this section to be focussed on sampled-based waveform and general playback features. Reworked overview/introduction. Added triggering at sample start/end when setting waveform point by point.</p> <p>13, "Waveform generator": Moved this to a separate section for clarity. Reworked overview/introduction. New waveform generator segment types to create ramps for fixed distance and duration, S-curve waveforms, steps using S-curve waveforms, and staircase steps. Sections arranged for easier navigation through related waveforms. Added reporting of failure cause for incorrectly configured waveforms. Added "supported in" section for each function type. Added "segment end" triggers, and offset times for segment triggers. Velocities now use nm/ms and accelerations use nm/ms/ms. Added "prepare and start" command.</p> <p>14, "Waveform builder": Added section for new feature.</p> <p>16.2, "Scope data signal selection": Updated for new control loop features.</p> <p>17, "Controller protection": Moved this to a separate section for clarity.</p> <p>18, "Controller synchronisation": Moved this to a separate section for clarity. Added description of synchronisation functionality.</p>

Revision	Changes
	<p>19, “Controller operation with sensor-only systems”: Added new section to cover use of controller with sensors.</p> <p>21, “Deprecated commands”: With control loop changes, deprecated some commands relating to position monitor low-pass filter, closed-loop, notch filter.</p>
5.1	<p>General: Standardised notation of enumeration types and values, to support reporting of enumeration values in controller interface version 2.6.</p> <p>6, “Controller communications”: Added new section to cover TCP/IP comms in firmware version 6.6.</p> <p>10, “Analogue inputs, outputs and position command”: Added analogue microns-per-volt scaling.</p> <p>20, “Controller operation with sensor-only systems” and generally throughout document: Added error report for sensor-only systems.</p> <p>21, “Diagnostics logging”: Minor changes to section ordering for consistency.</p>
5.2	<p>8, “Control system”: Added reporting of stage at actuator limits, with this contributing to reporting whether stage is moving. Control loop now includes limiting on actuator HV output where required. Command processing includes limits.</p> <p>8.5, “In position check”: Added details of check on whether stage is moving.</p> <p>8.3, “Command processing”: Added details of command limiting.</p> <p>8.7, “Feedforward command trajectory control”: Added note to cover launch acceleration set too low.</p> <p>8.12, “Actuator output”: Added reporting of whether stage is at actuator limits.</p> <p>10, “Resonance detection”: Inserted section in document for new feature. Subsequent sections are renumbered accordingly.</p> <p>13.2, “Stepped position command”: Added details of command limiting.</p> <p>19.2, “Stage short-circuit (overcurrent) protection”: Added section for new feature.</p>

1 Introduction

Queensgate NanoMechanisms combine a flexure-guided mechanism, a piezo actuator and a precision capacitive position sensor to allow highly-accurate position measurement. These NanoMechanisms are generally referred to as “stages”. Where multi-axis control is required, multiple stages may be assembled together, or a single stage may be designed to provide control in two or more axes.

The NanoScan NPC-D-6xxx series digital controllers provide closed-loop positioning control for one or more stage axes. The controller incorporates a high-voltage power amplifier to drive the piezo actuator, a precision measurement circuit to calculate the stage position, digital signal processing to provide closed-loop control of the stage(s), and various interfacing methods to set the commanded position and system calibration.

The same platform is used for the NS-D-6xxx series digital sensor controllers. These use the same platform design and interfacing, but omit all aspects relating to driving piezo actuators and to closed-loop control. This manual covers aspects of the sensor controller for completeness, because these commands exist for both controllers and sensors may also be connected to the NPC-D-6xxx series controllers.

To allow straightforward user control of the system, the controller interface DLL provides a standard method for a PC running Windows or Linux to interact with the controller. The user need only tell the DLL how the PC is connecting to the controller, and the DLL handles all details of the communications link. Requests/commands are provided to the DLL as text strings; the DLL translates these into the appropriate messages to/from the controller over the comms link; and responses from the controller are returned by the DLL as text strings.

Document EN-014217-UM describes the API for the DLL. This document describes the details of commands passed to the `DoCommand()` function from the DLL, to configure and control the digital controller’s operation, and the values passed back from the DLL for these commands.

2 System overview

2.1 Controller security

Customers may naturally wish to configure the system behaviour to meet their own requirements, but configuration must be performed by engineers or technicians who have sufficient expertise to carry it out safely. A simple interlock system is provided which restricts the commands available to users. This is intended to limit accidental misconfiguration of the system, and is not intended to provide cryptographic security.

Section 4 describes the operation of security and associated commands.

2.2 Controller and stage identification

The user needs to be able to identify the controller and stage connected, and other features of the controller and stage such as the number of channels on the controller, or the stage axis. Section 5 describes the associated commands.

2.3 Controller communications

Communication with the controller is mostly handled by the controller interface API, as described in manual EN-014217-UM. However commands are provided to configure the TCP/IP connection for control over Ethernet. Section 6 describes the associated commands.

2.4 Stage calibration presets

Stage calibration settings are stored on an EEPROM inside the stage connector. Controllers and stages may therefore be freely interchanged, enabling rapid replacement in case of failure.

Stages are pre-programmed with factory presets as specified in the stage datasheet. Customers are free to save further settings to presets as required. The intention is that the majority of users, who may not have the expertise or equipment to calibrate the system themselves, can simply load an appropriate preset for their system and start using the system immediately.

Section 5.2 describes the operation of stage preset storage and associated commands.

2.5 Control system

The commanded position may be set from a channel's analogue input or via digital commands from the host PC. Commands from the host PC may be via the comms interfaces, or may be via digital inputs emulating a stepper-motor interface. These may be used simultaneously if required, in which case commands will be summed. If a feedforward edge boost is required on command changes, high-pass filters may be used to add this.

The measured position is corrected for sensor linearity errors, and may have low-pass filtering applied to reduce noise.

The controller may be operated in open-loop or closed-loop modes. Open-loop mode allows direct drive of the actuator from the input(s), without ensuring the stage actually reaches its target; whereas closed-loop operation checks the measured position to ensure the target is reached. When in closed-loop mode, the “in position” status and digital output may be used for external systems to be alerted that the commanded position has been reached.

Closed-loop control is based around the industry-standard PID (proportional, integral, differential) controller. Proportional control includes the more advanced features of setpoint weighting and feedforward gain. Also as is industry-standard, the differential term uses only measured position (technically, using setpoint weighting with a setpoint of zero) to avoid changes to the commanded position causing large spikes on the servo output.

Closed-loop control may control position; position and velocity; or position, velocity and acceleration. As is normal, PID loops are nested to allow control of all three parameters. The control loop is designed to ensure stability in all operating modes.

For both open-loop and closed-loop modes, damping helps to control system resonances. A digital notch filter may be used to prevent the control loop exciting known resonances; and active damping may be used to actively fight any resonances which do occur.

To reduce ringing on larger steps, especially with high-mass or resonant loads, feedforward limiting of the commanded position may be used. This applies trapezoidal trajectory control, in the same way as a typical motor controller. Limiting the acceleration and deceleration can greatly reduce ringing. Limiting the commanded velocity to the velocity achievable by the system can also reduce PID integral overshoot in closed-loop mode.

To reduce system noise in some measurement applications, the actuator output may be temporarily frozen at the current position. This is typically used for fast measurements where actuator drift during sampling will be negligible but noise from closed-loop control may be significant.

The controller includes soft-start features to reduce sudden movements on power-on and on mode changes wherever possible.

Section 8 describes the operation of the control system and associated commands used to control, configure and calibrate the system. Section 9 provides some guidance on how to calibrate the system.

2.6 Resonance detection and frequency spectrum analysis

If the control loop tuning is too aggressive, or if a control loop configuration for a lighter, faster system is inadvertently selected when a heavier load is fitted to the stage, or if the notch filter removing the effects of the stage resonance is set incorrectly, the system may become unstable. This results in large (often end-to-end) oscillations of the stage at some resonant frequency, which may damage the stage flexures if the stage is left resonating for a long period.

The controller detects this resonance and shuts down the channel quickly, preventing stage damage. Detection uses continuous FFT frequency analysis of the stage position, which can also be read back by the user. This allows examination of the noise spectrum, analysis of harmonics when driving the system with a sine wave, or other similar operations.

Section 10 describes the operation of resonance detection protection and frequency spectrum analysis, and the associated commands used to control this.

2.7 Position rebalancing

The natural position of the stage will sink lower with a heavier load against the spring elements in the stage, and the piezo expansion changes with temperature. The total end-to-end stage range does not change significantly, but the exact position of the ends does.

For microscopy and other systems where it is important to deliver the required end-to-end stage range, the controller can automatically sweep the stage over its range to determine exactly where the range lies. It can then automatically self-calibrate and rebalance the stage position to achieve the required closed-loop range, with allow maximum headroom at each end.

Section 11 describes the operation of position rebalancing.

2.8 Analogue inputs, outputs and position command

Much OEM equipment uses analogue signals to set a position command, and the analogue input for each channel supports this. Analogue signals have some disadvantages: they are prone to noise; and good-quality ADCs and DACs can be expensive. The advantage is that almost all equipment supports analogue signals, so OEM customers may prefer this as a known, consistent interface.

Similarly, OEM equipment may use analogue signals to track the stage position. The measured position for each channel is reported on the position monitor analogue output.

Section 12 describes the operation of analogue inputs and outputs.

2.9 Stepped inputs, outputs and position command

The controller provides stepped inputs on its “Digital I/O” connector. These allow the controller to be connected to standard quadrature or step-and-direction signals, as if it were a stepper motor controller. With this interface it is possible to carry out fast, high-resolution and low-noise control without the need for expensive DACs, enabling a considerable saving on the system cost compared to traditional analogue inputs.

The controller also provides stepped outputs on the same connector which can be used for feedback of the commanded position (to ensure commands are received correctly) or to report the measured position. Again this allows a saving on system cost compared to analogue outputs by avoiding the need for expensive ADCs.

Section 13 describes the operation of stepped inputs and outputs.

2.10 Function playback and waveform creation

In addition to direct commands from the analogue input or host PC, the commanded position may also be set by preset waveforms stored for function playback. These may be played once, multiple times, or indefinitely. For multi-channel units, waveforms may be triggered simultaneously for multiple synchronised channels (for example to position a three-axis system), or may be run individually. Waveforms may be triggered from the host PC or from digital inputs, and may also be configured to trigger digital outputs at selected points during the waveform. Sections 14 describes the operation of function playback and the associated commands.

Waveforms for function playback can be set up as a sampled waveform, point by point. Alternatively and more easily, the waveform generator allows construction of a waveform in a similar way to dedicated motion controllers, providing component parts such as ramps, S-curve accelerations and other movement types which can be used to create the desired waveform. Sections 15 describes the waveform generator and the various types of movement which can be constructed through this.

For even easier access to complex waveforms, the waveform builder can construct entire waveforms automatically for some common applications such as Z-stacking. Section 16 describe the operation of the waveform builder and the waveforms available.

Commands from function playback command are summed with the analogue and digital commands. This allows various more sophisticated modes of operation to be configured which may allow better control of the desired response. For example, if the application is to use an XY stage to scan over a small area, function playback may be pre-programmed with a spiral or raster pattern on the channels controlling the X and Y axes. A direct command may then set the initial coordinates for the scan, and a single command will then trigger the functions for the X and Y channels when the stage reaches that point, which greatly reduces the workload instead of having to directly command every movement in the scan. This is particularly effective where the direct command for initial coordinates may come from the analogue input.

2.11 Snapshot capture

Snapshot capture allows sample-accurate acquisition of measured position over a short period of time, or optionally other parameters within the system, to be read back later by the host PC. Snapshots capture data for all channels; there is no provision for running snapshots on separate channels independently.

Snapshots are typically used to capture data whilst some operation is in progress. They may therefore be triggered simultaneously with function playback. Alternatively of course they may also be triggered independently.

Since a common use of snapshots is in measuring the system's step response, snapshots include the ability to generate a step in commanded position during the snapshot, which is easier than setting up a step profile in function playback. Function playback can still be used to create more complex stimuli, of course.

Section 17 describe the operation of snapshot capture and associated commands.

2.12 Scope signals

In normal operation, the analogue output and snapshot capture for each channel will capture the measured position for that channel.

When calibrating/configuring the system, it may be of use to measure internal state within the control loop. The controller provides a "scope" feature which allows any signal within the control loop to be monitored directly. It also allows signals to be routed to an analogue output and/or captured during a snapshot, instead of reporting measured position for a channel. The ID numbers to select each signal within the control loop are indicated in the diagrams in section 8.2 using the format **123** next to the relevant data flow.

Section 18 describes the operation of the scope functionality and associated commands.

2.13 Controller protection

The controller has protection against various situations which could cause damage to the controller. Section 19 describes the operation of these features and associated commands.

2.14 Controller synchronisation

When two or more controllers are used together, it is necessary for the controller internal timings to be synchronised to avoid potential issues with noise on the stage position measurement. Section 20 describes the operation of synchronisation and associated commands.

2.15 Controller operation with sensor-only systems

The controller can also be used to control Queensgate sensors, using the standard mechanisms for reporting measured position. Other functionality relating to commanding movement is naturally disabled for a sensor. Section 21 describes the use of sensors with this controller.

2.16 Diagnostics logging

In the event that a fault occurs in the controller or stage, it can be difficult to diagnose the cause of the fault. To assist in this, key data is captured and stored so that the cause of failure may be identified. This may allow Queensgate to improve product robustness with features which mitigate or prevent certain failure modes, or with real-world data from customer use, it may allow user guides to suggest ways in which customers can avoid situations which are more likely to result in fatigue or may otherwise reduce the system's lifespan. Where Queensgate equipment is incorporated in OEM products, this may also be useful for OEMs to check how end users are working with the system.

Section 22 describes the operation of diagnostics logging and associated commands.

3 Notation used in this document

3.1 Format of command descriptions

For clarity, notation conventions within this document must be explained.

Every command has a standard layout for common command details.

3.1.1 Command name

Commands are listed by name. This is the command name which is passed to the function `DoCommand()`. For example, as described in 5.1.10, a *command* string of “controller.sampling-time.get” passed to `DoCommand()` would read back the control loop sample time.

Frequently we will need to be able to change a value and to read it back. In these cases the two commands are grouped together for clarity. For example, as described in 8.3.4, “stage.position.command.get/set” specifies the commands to change and read back the current commanded position. A *command* string of “stage.position.command.set” passed to `DoCommand()` would set the current commanded position, and “stage.position.command.get” passed to `DoCommand()` would read back the current commanded position.

3.1.2 Description

The command name attempts to be a convenient mnemonic. However some description of the command or the value to be changed is naturally required.

3.1.3 Command

This illustrates the *command* string passed to `DoCommand()`, showing the command name and parameters.

3.1.4 Parameters

Parameters form part of the *command* string passed to `DoCommand()`. They identify which value is to be changed or read back, where there is more than one possibility, and will set the new value.

Some commands need no parameters. For example, a *command* string of “controller.sampling-time.get” as described in 5.1.10 is a complete command in itself, because the system only has one control loop sampling time. In this case the parameters are explicitly specified as “None”.

Most commands do need parameters. For example, the command “stage.position.measured.get” as described in 8.4.1 requires the channel number to be specified, otherwise it is unclear which channel’s position is to be reported. A *command* string of “stage.position.measured.get 2” will read back the position for the stage on channel 2.

Some commands need more than one parameter. For example, the command “function.command.start” as described in 14.4.2 must specify for all channels whether

they are to be started or not. In this case the *command* string must set each parameter in the order specified, separated by a space. A *command* string of “function.command.start 1 0 0 1 1” will start a snapshot and function playback on channels 2 and 3.

Where we have commands to change a value and to read it back, the two commands will usually need different parameters. These are specified in separate sections as “Parameters (get)” and “Parameters (set)”, or whatever suffix is appropriate for the commands. For example, as described in 8.3.4, “stage.position.command.get/set” specifies the commands to change and read back the current commanded position. A *command* string of “stage.position.command.set 2 3500” passed to **DoCommand()** would set the current commanded position for channel 2 to 3500pm, and “stage.position.command.get 2” passed to **DoCommand()** would read back the current commanded position for channel 2.

Units for values are stated where appropriate.

- Times or time constants are specified in seconds (s).
- Position or distance are specified in picometres (pm) for linear stages, or picoradians (prad) for angular stages. For simplicity, commands only state units for these values as pm, but users with angular stages should consider these to be prad. **GetCommandParameterUnits()** reports the correct units for the stage.
- Velocity or speed are specified in nanometres per millisecond (nm/ms) for linear stages, or nanoradians per millisecond (nrad/ms) for angular stages. For simplicity, commands only state units for these values as nm/ms, but users with angular stages should consider these to be nrad/ms. **GetCommandParameterUnits()** reports the correct units for the stage.
- Acceleration or deceleration are specified in nanometres per millisecond per millisecond (nm/ms/ms) for linear stages, or nanoradians per millisecond per millisecond (nrad/ms/ms) for angular stages. For simplicity, commands only state units for these values as nm/ms/ms, but users with angular stages should consider these to be nrad/ms/ms. **GetCommandParameterUnits()** reports the correct units for the stage.
- Frequencies are specified in Hertz (Hz).
- Gains are typically unitless (no units specified), except in cases such as reporting filter Bode plots where it is more appropriate to report them in decibels (dB).
- Enumerated values have units given as “enum”, with the meaning of each value specified in the parameter description.

Note that whilst it may seem confusing to state speeds in nanometres per millisecond when distances and times are in picometres and seconds respectively, units of picometres per second become numerically large and so are less convenient to work with.

Further details are generally required for parameters. Where values must be in a defined range, the minimum and/or maximum range is specified. Some parameters may be integers or floating-point, which will affect whether fractional values are expected or not. Some parameters may also be specified as text rather than numeric values, in which case the relevant text format is described.

If the *command* string contains too few parameters for a command, **DoCommand()** returns an error (negative value). If the *command* string contains more parameters than required for a command, additional parameters are silently ignored.

3.1.5 Results

If a command is sent successfully to the controller and returns data, the function `DoCommand()` returns the number of result values passed back by the controller for this command. When changing a value, this will usually mirror back the value changed; or when reading back a value, this will report the value. Some commands may report more than one result.

The function `GetResultName()` reads back the name of each result, and the function `GetResult()` reads back the result value.

Note that a command may be sent to the controller but may then fail, for example if parameters are out of range. Errors are reported using a standard format of results, as described in 3.2. If the result name for index 0 is “error”, the user knows the command failed. Common errors for all commands are described in 3.2.

If a command succeeds, the result name and results will be as specified in the “Results” section for the command. For example, as described in 5.1.10, a *command* string of “controller.sampling-time.get” passed to `DoCommand()` would return a value of 1, indicating that there is only one result. Calling `GetResultName()` for index 0 would report “value” as the result name, and calling `GetResult()` for index 0 would report a value of “2.0e-5” on an NPC-6000 series controller, which has a 20 μ s sample period when running at 50kHz.

Result values use the same units as parameters (see above). `GetCommandResultUnits()` reports the units for a result where these may be different for linear or angular stages.

Where “get” and “set” commands are specified together, typically the results will be the same for both. The “set” command will report a result which mirrors the value set (or will report an error if it cannot be set), and the “get” command will report the value currently set.

Some commands report more than one result value, for example “controller.status.get” as described in 5.1.11. The order of results for `GetResultName()` and `GetResult()` will always match the order specified, so for example for “controller.status.get”, result index 0 will report “security”, result index 1 will report “channels”, and result index 2 will report “status”. It is always best practise to check the name of the first result to ensure that a command has been carried out successfully, but results for a successful command will always be in the correct order, so it is not necessary to check the result names for subsequent results.

3.1.6 Possible error reports

The common error result format and common errors for all commands are described in 3.2. Each command will also typically have errors which are specific to the command: for example, the channel number may be invalid, or the commanded value may be out of range.

For each command, the possible error reports in the “errcode” result (index 1) are described. The user may assume that only error reports from 3.2 or from this section of the command description will occur.

Where “get” and “set” commands are specified together, typically there will be different error cases for the two commands. In this case the error reports for each are described separately.

3.1.7 Minimum security level

Many commands which only return values do not require security and will always be available. It is usually possible to observe system operation without security.

To avoid accidental modification of key settings though, many commands require a minimum security level to be carried out, as described in 2.1 and 4. Where this is required, it is specified for each command.

Where “get” and “set” commands are specified together, typically there will be different security levels required for the two commands. In this case these are specified separately.

3.1.8 Supported in

New features are being continuously developed for the controller. This manual documents features and commands for the latest versions of controller firmware and controller interface library software. In case users have not yet upgraded to the latest firmware or software, it is important that this document specifies the minimum versions of firmware and software necessary to use a particular command.

3.2 Error reporting

3.2.1 Error result fields

As described in 3.1.5, unsuccessful commands do not report the results specified for each command. Instead, all unsuccessful commands report the following results fields to report an error.

Name	Type	Units	Minimum	Maximum	Description
error	string				Always set to “FAILED”
errcode	string				Error code text

3.2.2 Error code text

The following standard error code text strings may be reported for every command.

Error result “errcode” string reported	Description
Command locked by security	The command is not valid for the security permissions unlocked (see section 4). Typically the interface DLL will prevent this happening, by concealing not-permitted commands from the user.
Command invalid	The command is not valid for this firmware.

Error result “errcode” string reported	Description
	Typically the interface DLL will prevent this happening, by concealing invalid commands from the user.
Command not available	The command is intended to be implemented for a later version of firmware, but is not currently implemented. Typically this will be seen for beta firmware during development. It will not normally been seen in production firmware.

In addition, every command has error codes specific to that command, as described in 3.1.6.

4 Controller security

4.1 Security access and security levels

The controller implements a simple form of security, where each security level is unlocked by entering a “key” value corresponding to the security level. This is not intended to prevent determined attacks, nor to provide cryptographic security, but is simply intended as an interlock to stop accidental changes to the configuration which would adversely affect the system behaviour. OEM customers may choose whether to inform end users of these keys, depending on whether end users should be permitted to change settings or not.

Security is implemented on a per-comms-link basis. It is possible to have multiple operators connected to the same controller on different comms links. In the case of Ethernet comms it is possible to have multiple operators connected over the same comms link, and in this case security is implemented on a per-connection basis.

The intention is that one operator could be in control, with “user” or “superuser” permissions. Other operators without security permissions set would be able to observe the system, but would be prevented from accidentally interfering with its behaviour. This provides a simple interlock method if the controller is required to be monitored by a SCADA system.

A further intention is that an operator with “user” permissions may control stage movement, but is prevented from changing parameters which may affect system stability. “User” level operators may load preset calibrations (see section 5.2) but may not themselves change settings from those calibrations. “Superuser” level operators may configure calibrations to tune the system, and save those calibrations to the stage for use by “users”. This provides a level of security if the controller will typically be operated by software on a production line, where users may not have the necessary skills or equipment to change calibrations safely.

To provide some limited protection against users brute-forcing key values, entering an invalid “key” value removes any current “user” or “superuser” permissions, and prevents further unlock requests for 5s.

The relevant “key” values are:-

Security level	Key value
“User” level security	233573869 (hex: 0xDEC0DED)
“Superuser” level security	2954754766 (hex: 0xB01DFACE)

4.1.1 controller.security.user.get/set

4.1.1.1 Description

Controller security level

4.1.1.2 Command

```
controller.security.user.get
controller.security.user.set <code>
```

4.1.1.3 Parameters (get)

None

4.1.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
code	32-bit unsigned integer				Key value to unlock desired security level

4.1.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
security	String	Enumeration “security-level”			Current controller security level “None” : No security “User”: User level security “Superuser”: Superuser level security

Note that after a “set” request with an invalid command unlock code, security will always return to level “None”.

4.1.1.6 Possible error reports

Error return “errcode” string reported	Description
Wait for 5s after invalid command unlock code	A previous request to unlock a security level was unsuccessful, and another request has been made before the 5s timeout has elapsed

4.1.1.7 Minimum security level

controller.security.user.get	No security required
controller.security.user.set	No security required

4.1.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

4.1.2 controller.security.lock

4.1.2.1 Description

Lock controller security

4.1.2.2 Command

controller.security.lock

4.1.2.3 Parameters

None

4.1.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
security	String	Enumeration “security-level”			Current controller security level Always returns “None” : No security

4.1.2.5 Possible error reports

None

4.1.2.6 Minimum security level

controller.security.lock	No security required
--------------------------	----------------------

4.1.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5 Controller and stage identification

5.1 Controller identification

5.1.1 identity.hardware.part.get

5.1.1.1 Description

Controller part number

5.1.1.2 Command

identity.hardware.part.get

5.1.1.3 Parameters

None

5.1.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
part	String				Controller part number

5.1.1.5 Possible error reports

None

5.1.1.6 Minimum security level

identity.hardware.part.get	No security required
----------------------------	----------------------

5.1.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.2 identity.hardware.serial.get

5.1.2.1 Description

Controller serial number

5.1.2.2 Command

identity.hardware.serial.get

5.1.2.3 Parameters

None

5.1.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
serial	32-bit unsigned integer				Controller serial number

5.1.2.5 Possible error reports

None

5.1.2.6 Minimum security level

identity.hardware.serial.get	No security required
------------------------------	----------------------

5.1.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.3 identity.hardware.mandate.get

5.1.3.1 Description

Controller manufacture date

5.1.3.2 Command

identity.hardware.mandate.get

5.1.3.3 Parameters

None

5.1.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
mandate	32-bit unsigned integer	MS Excel serial day			Controller manufacture date

5.1.3.5 Possible error reports

None

5.1.3.6 Minimum security level

identity.hardware.mandate.get	No security required
-------------------------------	----------------------

5.1.3.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.4 identity.hardware.caldate.get

5.1.4.1 Description

Controller calibration date

5.1.4.2 Command

identity.hardware.caldate.get

5.1.4.3 Parameters

None

5.1.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
caldate	32-bit unsigned integer	MS Excel serial day			Controller calibration date

5.1.4.5 Possible error reports

None

5.1.4.6 Minimum security level

identity.hardware.caldate.get	No security required
-------------------------------	----------------------

5.1.4.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.5 identity.hardware.bootloader-version.get

5.1.5.1 Description

Controller software bootloader version number

5.1.5.2 Command

identity.hardware.bootloader-version.get

5.1.5.3 Parameters

None

5.1.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
version	32-bit unsigned integer				<p>Version number</p> <p>Bits 31-24: Major version</p> <p>Bits 23-16: Minor version</p> <p>Bits 15-0: Build number</p>

5.1.5.5 Possible error reports

None

5.1.5.6 Minimum security level

identity.hardware.bootloader-version.get	No security required
--	----------------------

5.1.5.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.6 identity.hardware.platform-version.get

5.1.6.1 Description

Controller software platform version number

5.1.6.2 Command

```
identity.hardware.platform-version.get
```

5.1.6.3 Parameters

None

5.1.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
version	32-bit unsigned integer				<p>Version number</p> <p>Bits 31-24: Major version</p> <p>Bits 23-16: Minor version</p> <p>Bits 15-0: Build number</p>

5.1.6.5 Possible error reports

None

5.1.6.6 Minimum security level

identity.hardware.platform-version.get	No security required
--	----------------------

5.1.6.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.7 identity.software.version.get

5.1.7.1 Description

Control application software version number

5.1.7.2 Command

identity.software.version.get

5.1.7.3 Parameters

None

5.1.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
version	32-bit unsigned integer				<p>Version number</p> <p>Bits 31-24: Major version</p> <p>Bits 23-16: Minor version</p> <p>Bits 15-0: Build number</p>

5.1.7.5 Possible error reports

None

5.1.7.6 Minimum security level

identity.software.version.get	No security required
-------------------------------	----------------------

5.1.7.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.8 identity.software.reldate.get

5.1.8.1 Description

Control application software release date

5.1.8.2 Command

identity.software.reldate.get

5.1.8.3 Parameters

None

5.1.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
caldate	32-bit unsigned integer	MS Excel serial day			Software release date

5.1.8.5 Possible error reports

None

5.1.8.6 Minimum security level

identity.software.reldate.get	No security required
-------------------------------	----------------------

5.1.8.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.9 identity.software.part.get

5.1.9.1 Description

Control application software part number

5.1.9.2 Command

identity.software.part.get

5.1.9.3 Parameters

None

5.1.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
caldate	32-bit unsigned integer				Software part number

5.1.9.5 Possible error reports

None

5.1.9.6 Minimum security level

identity.software.part.get	No security required
----------------------------	----------------------

5.1.9.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.10 controller.sampling-time.get

5.1.10.1 Description

Control loop sample time

5.1.10.2 Command

controller.sampling-time.get

5.1.10.3 Parameters

None

5.1.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s			Control loop sample time

5.1.10.5 Possible error reports

None

5.1.10.6 Minimum security level

controller.sampling-time.get	No security required
------------------------------	----------------------

5.1.10.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.11 controller.channels.get

5.1.11.1 Description

Number of channels on controller

5.1.11.2 Command

```
controller.channels.get
```

5.1.11.3 Parameters

None

5.1.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Number of channels on controller

5.1.11.5 Possible error reports

None

5.1.11.6 Minimum security level

controller.channels.get	No security required
-------------------------	----------------------

5.1.11.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.1.12 controller.status.get

5.1.12.1 Description

Controller system status

5.1.12.2 Command

```
controller.status.get
```

5.1.12.3 Parameters

None

5.1.12.4 Results

Name	Type	Units	Minimum	Maximum	Description
security	String	Enumeration “security-level”			Current controller security level “None” : No security “User”: User level security “Superuser”: Superuser level security
channels	8-bit unsigned integer				Number of controller channels
status	16-bit unsigned integer	Bitfield			Reserved

5.1.12.5 Possible error reports

None

5.1.12.6 Minimum security level

controller.status.get	No security required
-----------------------	----------------------

5.1.12.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.2 Stage presence

5.2.1 stage.status.stage-connected.get

5.2.1.1 Description

Stage is connected and ready for use

5.2.1.2 Command

```
stage.status.stage-connected.get <channel>
```

5.2.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

5.2.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage is connected and ready for use

5.2.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

5.2.1.6 Minimum security level

stage.status.stage-connected.get	No security required
----------------------------------	----------------------

5.2.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.3 Stage identification

5.3.1 identity.stage.part.get

5.3.1.1 Description

Stage part number

5.3.1.2 Command

```
identity.stage.part.get <channel>
```

5.3.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

5.3.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
part	String				Stage part number

5.3.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

5.3.1.6 Minimum security level

identity.stage.part.get	No security required
-------------------------	----------------------

5.3.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.3.2 identity.stage.serial.get

5.3.2.1 Description

Stage serial number

5.3.2.2 Command

```
identity.stage.serial.get <channel>
```

5.3.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

5.3.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
part	32-bit unsigned integer				Stage serial number

5.3.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

5.3.2.6 Minimum security level

identity.stage.serial.get	No security required
---------------------------	----------------------

5.3.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.3.3 identity.stage.axisid.get

5.3.3.1 Description

Stage axis for this controller channel

5.3.3.2 Command

```
identity.stage.axisid.get <channel>
```

5.3.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

5.3.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
axisid	String	Enumeration “stage-axis”			Stage axis for this controller channel: “x”, “y”, “z”, “theta”, “gamma”, “phi” or “unspecified”

Note that the axis will typically be reported as “unspecified” for single-axis stages. This is primarily intended for axis identification on multi-axis systems.

5.3.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

5.3.3.6 Minimum security level

identity.stage.axisid.get	No security required
---------------------------	----------------------

5.3.3.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.3.4 identity.stage.mandate.get

5.3.4.1 Description

Stage manufacture date

5.3.4.2 Command

```
identity.stage.mandate.get <channel>
```

5.3.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

5.3.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
mandate	32-bit unsigned integer	MS Excel serial day			Stage manufacture date

5.3.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

5.3.4.6 Minimum security level

identity.stage.mandate.get	No security required
----------------------------	----------------------

5.3.4.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

5.3.5 identity.stage.caldate.get

5.3.5.1 Description

Stage calibration date

5.3.5.2 Command

```
identity.stage.caldate.get <channel>
```

5.3.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

5.3.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
caldate	32-bit unsigned integer	MS Excel serial day			Stage calibration date

5.3.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

5.3.5.6 Minimum security level

identity.stage.caldate.get	No security required
----------------------------	----------------------

5.3.5.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

6 Controller communications

6.1 USB and RS-232 communications

The RS-232 connection handled by the controller interface API as documented in manual EN-014217-UM. It uses a fixed baudrate and is not configurable, so has no associated commands. The USB connection uses an emulated serial port which similarly is not configurable.

6.2 Ethernet communications

TCP/IP v4 is used for Ethernet connections. A static IP address must be assigned to the controller. By default this IP address is 192.168.0.7, but this may be changed by the customer. Note that this static IP address must also be used when the controller is reprogrammed using the Ethernet connection.

By default the TCP/IP connection uses TCP port 18881, which is not in use by other communication protocols or major applications. This may also be changed if required. Connected PCs may have firewalls whose configuration is locked down by corporate IT policies and cannot be changed; or other in-house applications may have been assigned this TCP port. In this case the TCP port used by the controller should be changed to one which the firewall will permit, or which does not conflict with other applications.

Changes to the IP address or TCP port do not take effect until the controller is powered off and back on.

6.2.1 controller.tcpip-comms.ip-address.get/set

6.2.1.1 Description

Controller static IP address for TCP/IP (Ethernet) comms

6.2.1.2 Command

```
controller.tcpip-comms.ip-address.get  
controller.tcpip-comms.ip-address.set <value>
```

6.2.1.3 Parameters (get)

None

6.2.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	Dotted-quad IP address				Controller static IP address

6.2.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	Dotted-quad IP address				Controller static IP address

6.2.1.6 Possible error reports

None from the controller.

However the API will report that the command could not be run if the dotted-quad is invalid. The dotted-quad must contain four numeric values separated by “.” characters, and these values must be between 0 and 255. If the dotted-quad is invalid, this is blocked by the API and is not sent to the controller.

6.2.1.7 Minimum security level

controller.tcpip-comms.ip-address.get	No security required
controller.tcpip-comms.ip-address.set	Superuser level security required

6.2.1.8 Supported in

Controller application firmware	6.6.14 onwards
Controller interface library	2.6.15 onwards

6.2.2 controller.tcpip-comms.tcp-port.get/set

6.2.2.1 Description

Controller TCP port number for TCP/IP (Ethernet) comms

6.2.2.2 Command

```
controller.tcpip-comms.tcp-port.get  
controller.tcpip-comms.tcp-port.set <value>
```

6.2.2.3 Parameters (get)

None

6.2.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		0	65535	Controller TCP port number

6.2.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		0	65535	Controller TCP port number

6.2.2.6 Possible error reports

Error return “errcode” string reported	Description
Value out of range	Port number is not within the specified range (set only)

6.2.2.7 Minimum security level

controller.tcpip-comms.tcp-port.get	No security required
controller.tcpip-comms.tcp-port.set	Superuser level security required

6.2.2.8 Supported in

Controller application firmware	6.6.14 onwards
Controller interface library	2.6.15 onwards

7 Stage calibration presets

7.1 Stage presets

Calibration settings for stage control (i.e. all settings whose commands start with “stage.”) may be stored as preset calibrations. These presets are stored in an EEPROM inside the stage connector. This allows the controller to be changed without affecting the stage calibration, enabling easy replacement in case of failure.

A total of 8 calibration presets are available. Some factory-set presets (typically named “Fast”, “Medium” and “Slow”) are configured during production and may not be changed by customers. The stage datasheet lists the factory-set presets available for a stage, as well as the characteristics of that preset (position noise, bandwidth, step response time, and so on). The remaining presets are available for customer use.

The factory-set presets are often sufficient for the majority of applications. Where the application requires a more specific configuration, users with “superuser” security level (see section 4) may set up a calibration matching their requirements and save this to a preset. Tuning the calibration for a system will generally require expertise in control systems, may often require specialist measurement equipment, and may present some risk of damage to the stage or to the system if incorrectly configured, so appropriate security access is required. Because several presets are available for customer user, it is possible to tailor presets to different configurations if required – perhaps for different speeds of response, noise levels, or loads.

Presets may be given names to identify the configuration for that preset. Preset names are set and read as zero-terminated strings of bytes, allowing both ASCII and UTF-8 text for the benefit of non-English-speaking customers.

Since most users will not have the expertise or equipment to configure the system correctly (or perhaps safely), users with “user” security level (see section 4) are not permitted to modify these settings. They are permitted only to load pre-programmed presets, either factory-set or as configured by a “superuser”. For most users this is an advantage, since they are able to choose appropriate settings for their application without needing expert knowledge of control systems. Presets can be loaded at any time, as required for the application.

The default preset to load on stage connection/power-up may be set by the “superuser”. Where only one preset will be used, this ensures the system is immediately functional out of the box. Where multiple presets may be used, this will typically be set to load a preset which best ensures system stability; other presets may then be loaded afterwards as required.

7.2 Other stage data

Further data may be stored on the stage EEPROM which is not generally accessible to the user. Position linearisation for example (see diagram 8.2.3) requires position measurement using an interferometer in order to calibrate the stage during manufacture. The controller does not allow customers to read this data, because the customer cannot make use of it themselves.

Some further settings may be stored on the stage EEPROM which may be read by the user but not changed. Diagnostics logging for example (see 19) captures data over the stage’s life so that causes of failures can be identified more easily when a faulty stage is returned. This information can be read, but the controller does not allow it to be changed at will.

7.3 Stage calibration storage

Whilst a “superuser” is calibrating the system, changes to settings are only stored in the controller RAM. Changes are not automatically stored to the stage EEPROM, and will be lost if a new preset is loaded, the stage is disconnected, or the controller is powered off. This is intentional, allowing an “emergency stop” of the controller in case of instability which could damage the stage or system, and ensuring the controller does not then restart with that unstable configuration. Calibrators must therefore ensure they save settings to a preset if they are required to be kept.

When a calibration preset is saved, it takes a short period of time for the data to be fully written back to the stage. Controller operation is unaffected, but if power is lost during this period then this calibration preset may be corrupted and its data lost. It is vital after saving a calibration preset that the user reads the command `stage.calibration.status.get` until it reports “idle”. While it reports “busy”, the controller is still saving data to the stage and controller power must not be removed.

It can be difficult during calibration to keep track of progress. It is common to change preset names as required to assist with this. Where this is not desirable, presets also include a “configuration ID” which can be freely set to any value, to help track changes to presets as the tuning is developed. This value may correspond to an entry in a calibration database, a test number or a timestamp, for example.

7.4 Stage preset index

In commands below, the index of the stage preset to check/load/save must be specified. For historical reasons, these are numbered 3 to 10. The default preset arrangement for most stages is:-

Preset index	Factory/customer preset	Preset name
3	Factory	Fast
4	Factory	Medium
5	Factory	Slow
6	Customer	Customer 1
7	Customer	Customer 2
8	Customer	Customer 3
9	Customer	Customer 4
10	Customer	Customer 5

This will vary depending on the stage. Stages may have more or less factory presets, and these presets may be named to reflect their application such as the intended load. See the stage datasheet for further details.

Factory presets may be loaded but not saved or deleted. Customer presets may be freely loaded, saved and deleted as required, and may be renamed. Stages are supplied with all customer presets empty initially.

At additional cost, OEM customers may specify presets to be loaded during stage manufacture. Contact a distributor for further information.

7.5 Stage calibration preset commands

7.5.1 stage.calibration.status.get

7.5.1.1 Description

Get status of current stage operation

7.5.1.2 Command

```
stage.calibration.status.get <channel>
```

7.5.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

7.5.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				<p>Stage operation status</p> <p>0 = Loading stage data</p> <p>1 = Idle</p> <p>2 = Busy</p> <p>4 = Load failure</p> <p>5 = Write failure</p> <p>6 = Verify failure</p>

7.5.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

7.5.1.6 Minimum security level

stage.calibration.status.get	No security required
------------------------------	----------------------

7.5.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.2 stage.calibration.preset.current.get

7.5.2.1 Description

Current calibration preset selected for stage

7.5.2.2 Command

```
stage.calibration.preset.current.get <channel>
```

7.5.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

7.5.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		See section 7.4		Current calibration preset

7.5.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

7.5.2.6 Minimum security level

stage.calibration.preset.current.get	No security required
--------------------------------------	----------------------

7.5.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.3 stage.calibration.preset.default.get

7.5.3.1 Description

Default (startup) calibration preset selected for stage

7.5.3.2 Command

```
stage.calibration.preset.default.get <channel>
```

7.5.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

7.5.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		See section 7.4		Default calibration preset

7.5.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

7.5.3.6 Minimum security level

stage.calibration.preset.default.get	No security required
--------------------------------------	----------------------

7.5.3.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.4 stage.calibration.preset.default.save

7.5.4.1 Description

Set the current calibration preset as the default (startup) calibration preset for stage

7.5.4.2 Command

```
stage.calibration.preset.default.save <channel>
```

7.5.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

7.5.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Carried out successfully 1 = started No other values possible

7.5.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Stage calibration data storage fault	Non-volatile memory storing stage calibration has failed so stage calibration data cannot be loaded or saved

7.5.4.6 Minimum security level

stage.calibration.preset.default.save	For firmware 6.6.1 onwards, user level security required For earlier firmware, superuser level security required
---------------------------------------	---

7.5.4.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.5 stage.calibration.preset.exists

7.5.5.1 Description

Check whether calibration preset exists

7.5.5.2 Command

```
stage.calibration.preset.exists <channel> <value>
```

7.5.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer		See section 7.4		Calibration preset

7.5.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Calibration preset exists 1 = exists 0 = does not exist

7.5.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Calibration preset value is not a valid preset ID

7.5.5.6 Minimum security level

stage.calibration.preset.exists	No security required
---------------------------------	----------------------

7.5.5.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.6 stage.calibration.preset.is-factory.get

7.5.6.1 Description

True if preset is factory-set and cannot be changed

7.5.6.2 Command

```
stage.calibration.preset.is-factory.get <channel> <preset>
```

7.5.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
preset	32-bit unsigned integer		See section 7.4		Calibration preset

7.5.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Calibration preset is factory-set 1 = factory-set preset 0 = customer preset

7.5.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Calibration preset value is not a valid preset ID

7.5.6.6 Minimum security level

stage.calibration.preset.is-factory.get	No security required
---	----------------------

7.5.6.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

7.5.7 stage.calibration.preset.load

7.5.7.1 Description

Load a calibration preset for stage

7.5.7.2 Command

```
stage.calibration.preset.load <channel> <value>
```

7.5.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer		See section 7.4		Load calibration preset

7.5.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Carried out successfully 1 = started No other values possible

7.5.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Calibration preset value is not a valid preset ID, or preset does not exist
Stage calibration data storage fault	Non-volatile memory storing stage calibration has failed so stage calibration data cannot be loaded or saved

7.5.7.6 Minimum security level

stage.calibration.preset.load	User level security required
-------------------------------	------------------------------

7.5.7.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.8 stage.calibration.preset.save

7.5.8.1 Description

Save current calibration as a calibration preset for stage. The preset name is left unchanged; stage settings are saved to this preset without changing the preset name.

See also `stage.calibration.preset.save-with-name` (section 7.5.9).

7.5.8.2 Command

```
stage.calibration.preset.save <channel> <value>
```

7.5.8.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer		See section 7.4		Save calibration preset

7.5.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Carried out successfully 1 = started No other values possible

7.5.8.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Calibration preset value is not a valid preset ID, or is a factory preset which cannot be saved to
Stage calibration data storage fault	Non-volatile memory storing stage calibration has failed so stage calibration data cannot be loaded or saved

7.5.8.6 Minimum security level

stage.calibration.preset.save	Superuser level security required
-------------------------------	-----------------------------------

7.5.8.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.9 stage.calibration.preset.save-with-name

7.5.9.1 Description

Save current calibration as a calibration preset for stage, including name. The new preset name must not be the same as the name of any other preset currently saved.

See also `stage.calibration.preset.save` (section 7.5.8).

7.5.9.2 Command

```
stage.calibration.preset.save-with-name <channel> <value>
```

7.5.9.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer		See section 7.4		Save calibration preset

7.5.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Carried out successfully 1 = started No other values possible

7.5.9.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Calibration preset value is not a valid preset ID, or is a factory preset which cannot be saved to
Stage calibration data storage fault	Non-volatile memory storing stage calibration has failed so stage calibration data cannot be loaded or saved
Value must not be duplicate	Another stage preset exists with the same name

7.5.9.6 Minimum security level

stage.calibration.preset.save	Superuser level security required
-------------------------------	-----------------------------------

7.5.9.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

7.5.10 stage.calibration.preset.delete**7.5.10.1 Description**

Delete a calibration preset for stage

7.5.10.2 Command

```
stage.calibration.preset.delete <channel> <value>
```

7.5.10.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer		See section 7.4		Delete calibration preset

7.5.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Carried out successfully 1 = started No other values possible

7.5.10.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Calibration preset value is not a valid preset ID, or is a factory preset which cannot be deleted
Stage calibration data storage fault	Non-volatile memory storing stage calibration has failed so stage calibration data cannot be loaded or saved
Command could not be carried out	Delete command attempted to delete current or default calibration preset, which is not permitted

7.5.10.6 Minimum security level

stage.calibration.preset.delete	Superuser level security required
---------------------------------	-----------------------------------

7.5.10.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

7.5.11 stage.calibration.preset.name.get/set

7.5.11.1 Description

Name for this stage preset

7.5.11.2 Command

```
stage.calibration.preset.name.get <channel>  
stage.calibration.preset.name.set <channel> <value>
```

7.5.11.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

7.5.11.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	String				Preset name

The preset name is a byte string. All characters following the channel number will be interpreted as forming the preset name. The preset name may be up to 32 bytes long, not including any zero terminator. If the name is longer than 32 bytes, it will be truncated to 32 bytes.

Spaces may be used in the preset name. The preset name may not begin with a space.

Newline characters (ASCII codes 10 or 13, C escaped characters “\r” or “\n”) may not be used, as they will be recognised as the end of the command. It is highly recommended that other control characters (ASCII codes 1-31) are not used, since the result may be unpredictable when displayed in different PC applications.

The string may be represented using ASCII or UTF-8 characters. If using UTF-8 with strings which contain multi-byte characters, the user’s application is responsible for encoding/decoding the byte string to be stored. The interface library and controller firmware do not attempt to validate a UTF-8 string. Note that if a UTF-8 string is specified which is longer than 32 bytes, truncating the name to 32 bytes may result in corruption of a multi-byte character, which the user’s application must tolerate.

7.5.11.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String				Preset name

7.5.11.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

7.5.11.7 Minimum security level

stage.calibration.preset.name.get	No security required
stage.calibration.preset.name.set	Superuser level security required

7.5.11.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

7.5.12 stage.calibration.preset.configuration-id.get/set**7.5.12.1 Description**

Configuration ID for this stage preset

7.5.12.2 Command

```
stage.calibration.preset.configuration-id.get <channel>
```

```
stage.calibration.preset.configuration-id.set <channel> <value>
```

7.5.12.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

7.5.12.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer				Configuration ID

7.5.12.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Configuration ID

7.5.12.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

7.5.12.7 Minimum security level

stage.calibration.preset.configuration-id.get	No security required
stage.calibration.preset.configuration-id.set	Superuser level security required

7.5.12.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8 Control system

8.1 Overview

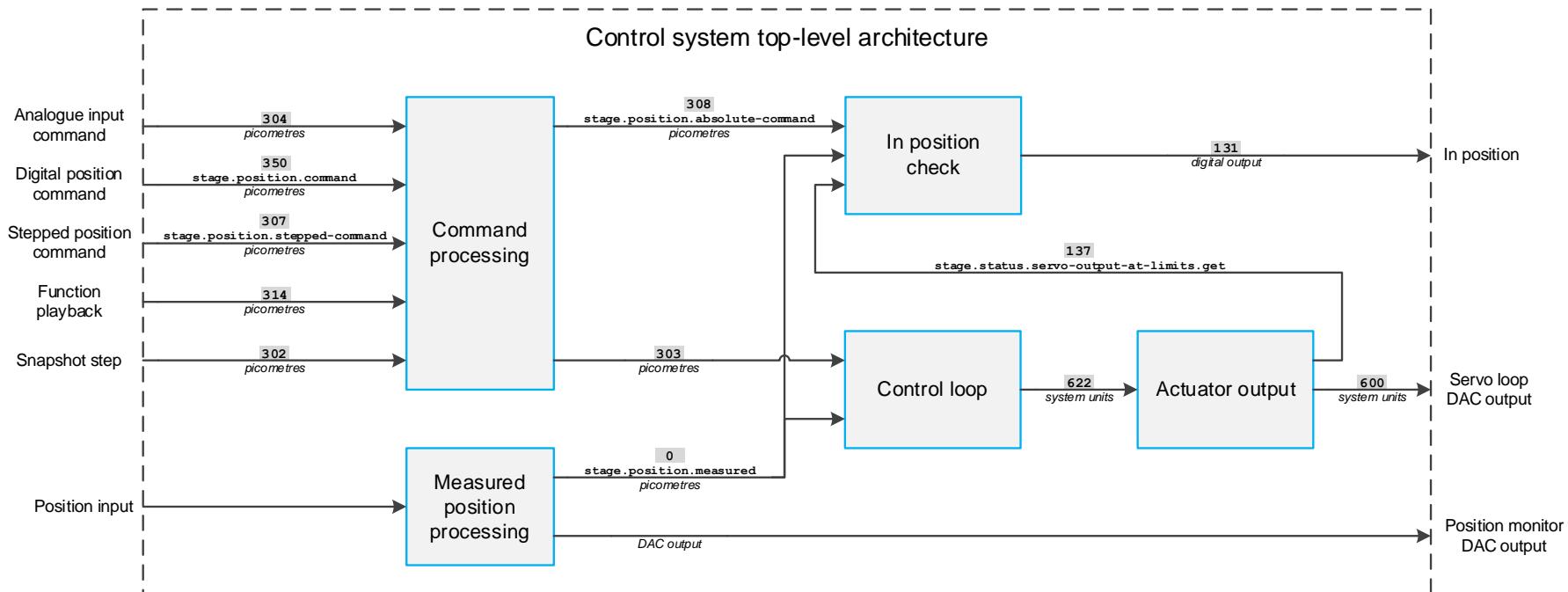
This section describes the control system within the controller. The controller is primarily intended to operate closed-loop, but is capable of operating open-loop if required.

Section 8.2 documents the signal processing within the control loop using diagrams. Subsequent sections describe each element of the control loop, with full descriptions of each command used to read or modify settings. Where some details of the control loop may not be immediately clear, or where care may need to be taken during calibration, these sections provide further details.

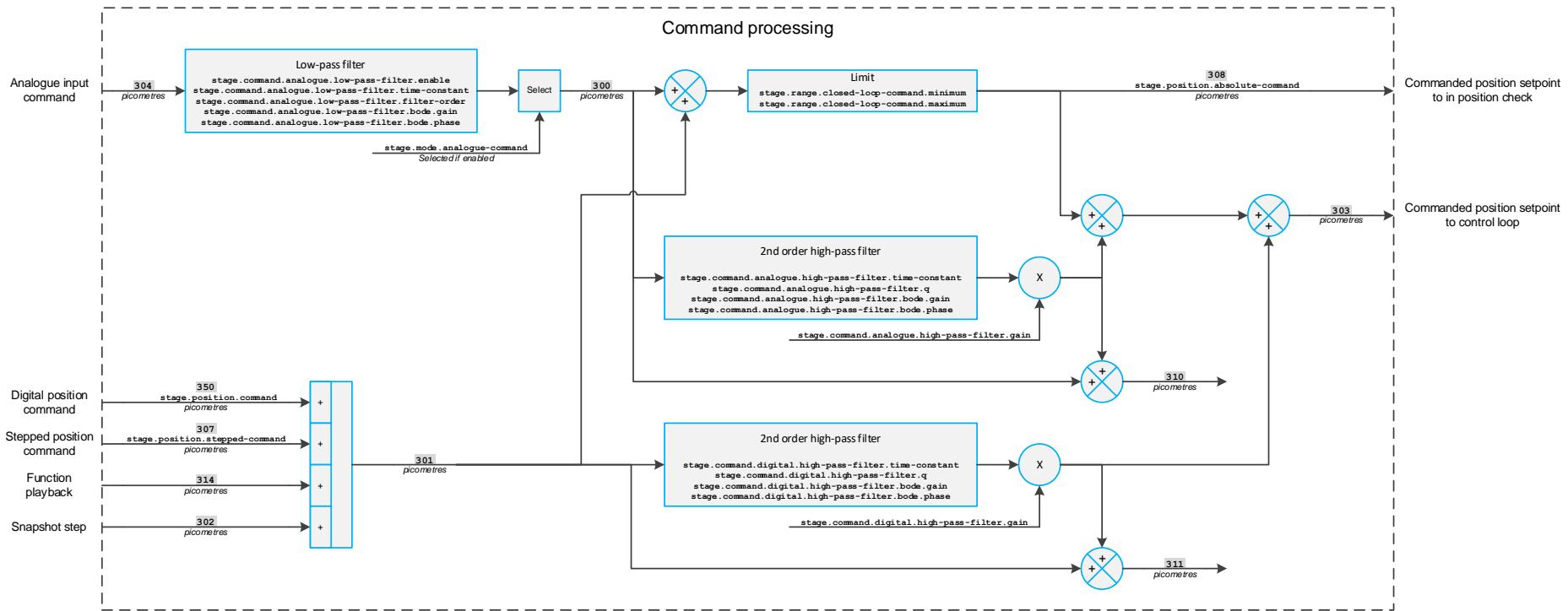
As described in section 8.2, scope points may be used to monitor internal control loop signals at key locations. The ID numbers to select each signal within the control loop are indicated in the diagrams in using the format **123** next to the relevant data flow.

8.2 Control system architecture

8.2.1 Control system top-level architecture



8.2.2 Command processing



See 8.3 for description of functionality and details of commands.

See 12.1 for details of analogue input command, 13.2 for details of stepped position command, 14 for details of function playback, and 17 for details of snapshot step. These require additional explanation which will not be duplicated here.

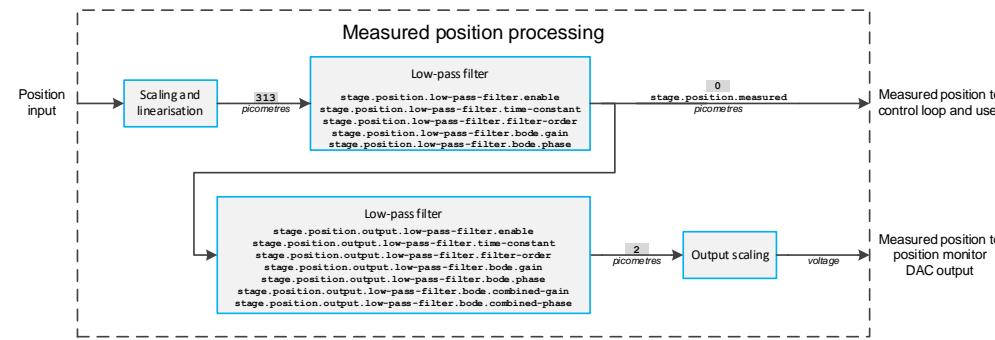
The value `stage.position.absolute-command` may be set as well as read. When this is set, the digital position command is set to the difference between this value and the sum of all other (enabled) commands, so that the overall commanded position setpoint is as specified.

Note that command limiting does not just clip the absolute command, but also limits the digital command if the absolute command exceeds the limit. See 8.3 for further details of this.

Because the analogue input is inherently noisy, the edge boost filters (if enabled) may require different settings for analogue and digital commands.

Note that the absolute command used by the in position check only considers the input commands and does not include the edge boost.

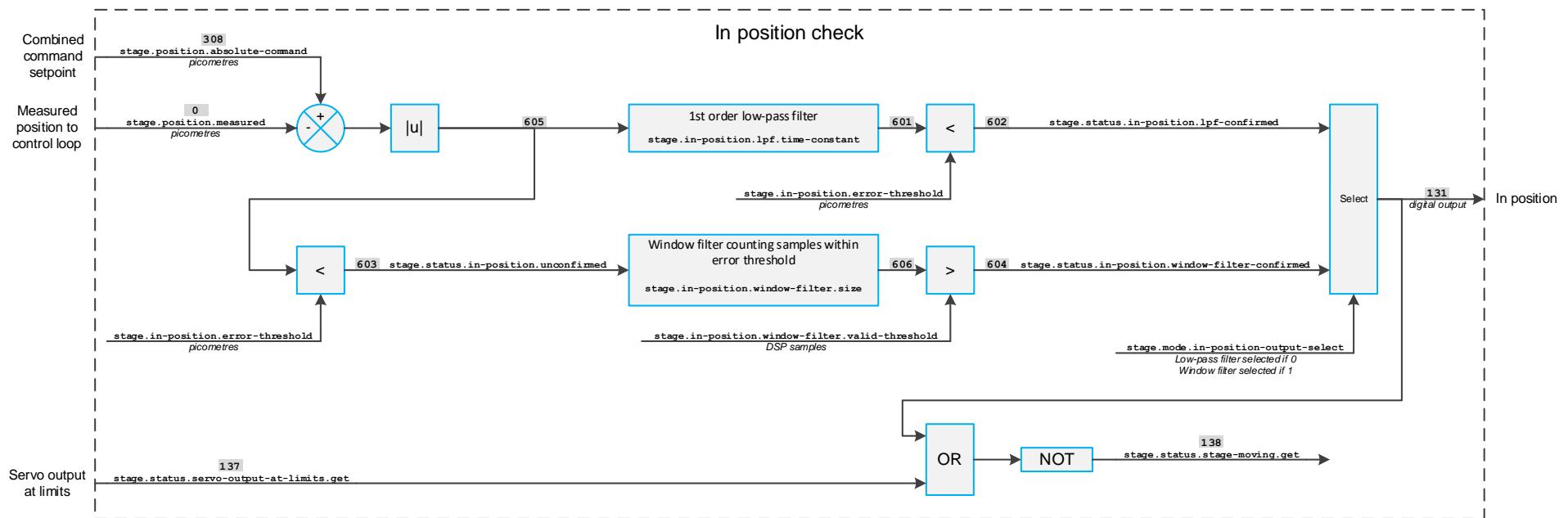
8.2.3 Measured position processing



See 8.4 for description of functionality and details of commands.

See 12.2 for details of position monitor analogue output. This requires additional explanation which will not be duplicated here.

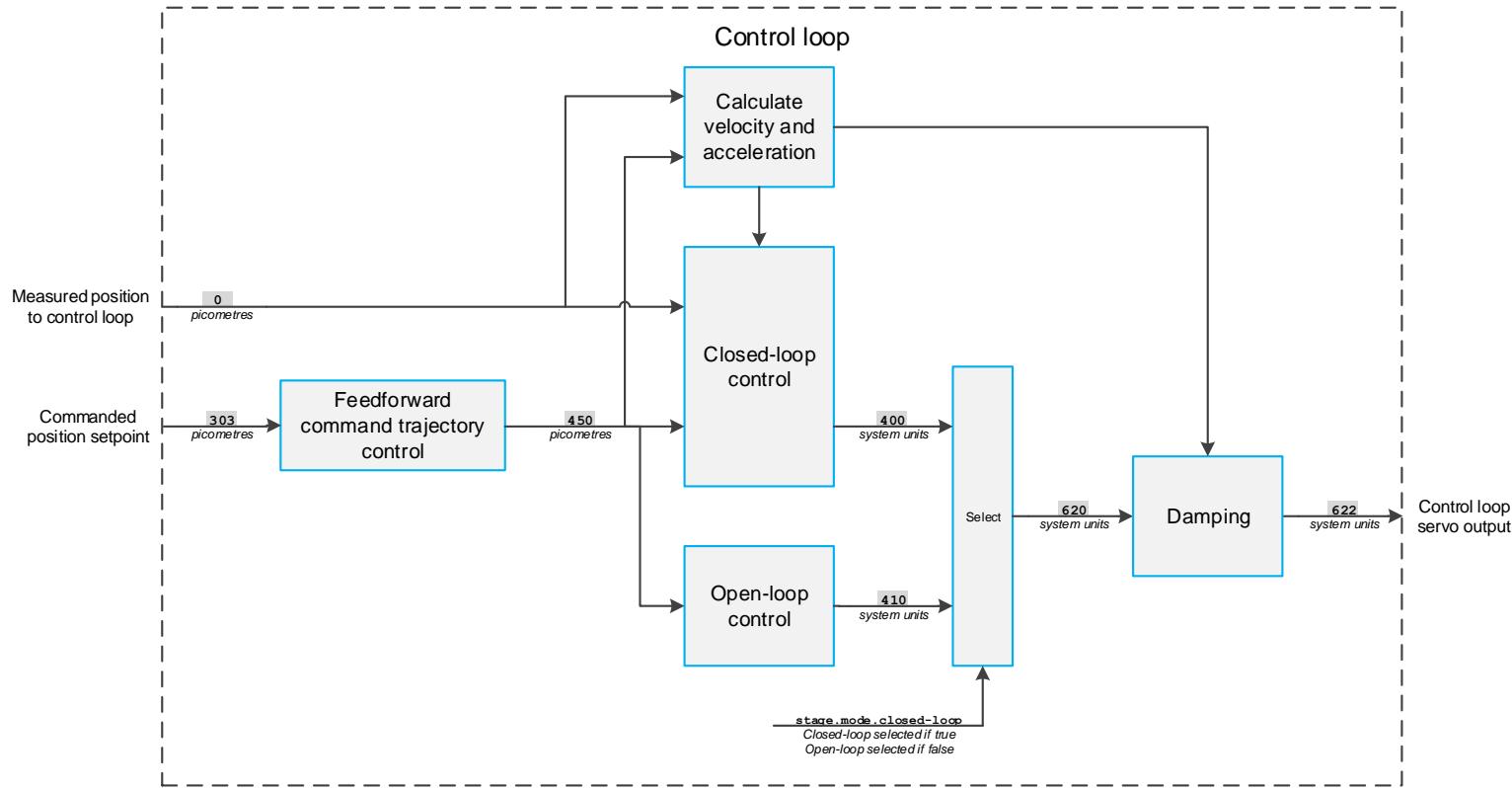
8.2.4 In position check



See 8.5 for description of functionality and details of commands.

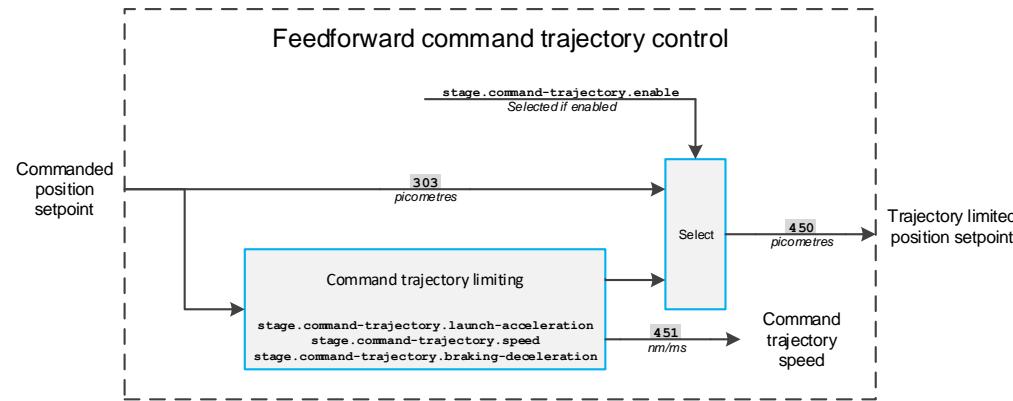
The in position ("ready") check only runs when the control loop for the channel is in closed-loop mode (see 8.6.1, `stage.mode.closed-loop`). In open-loop mode, the command does not relate to an actual position, so the in position status/output is always false or logic-low in open-loop mode.

8.2.5 Control loop



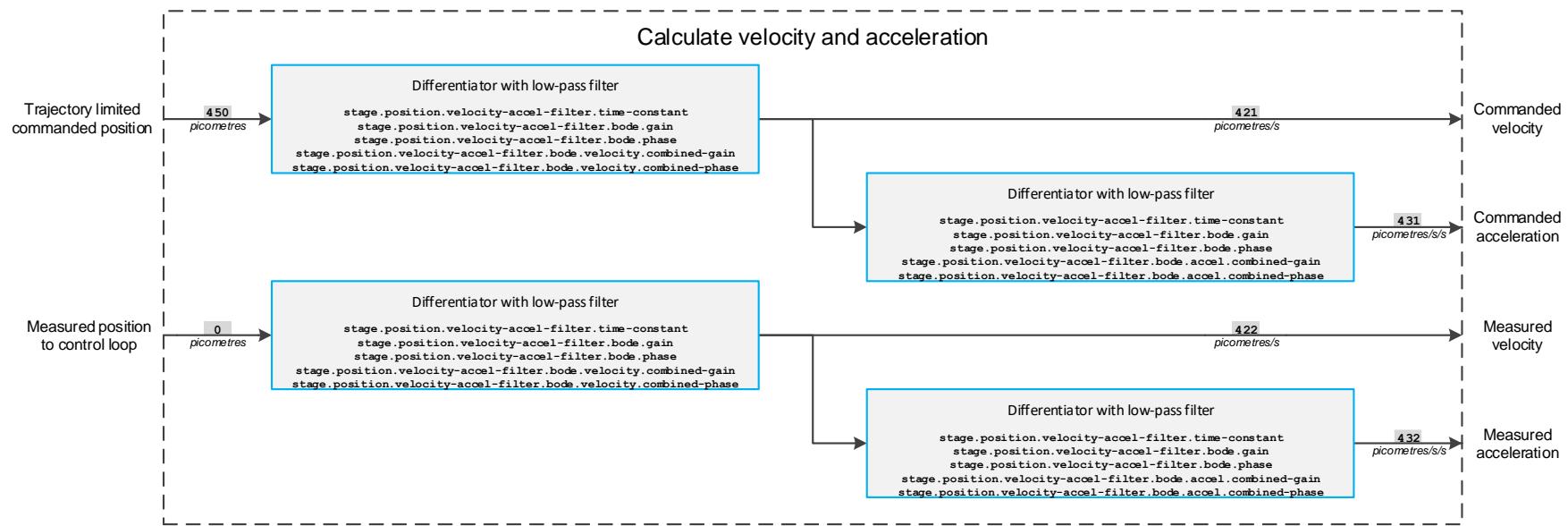
See 8.6 for description of functionality and details of commands.

8.2.6 Feedforward command trajectory control



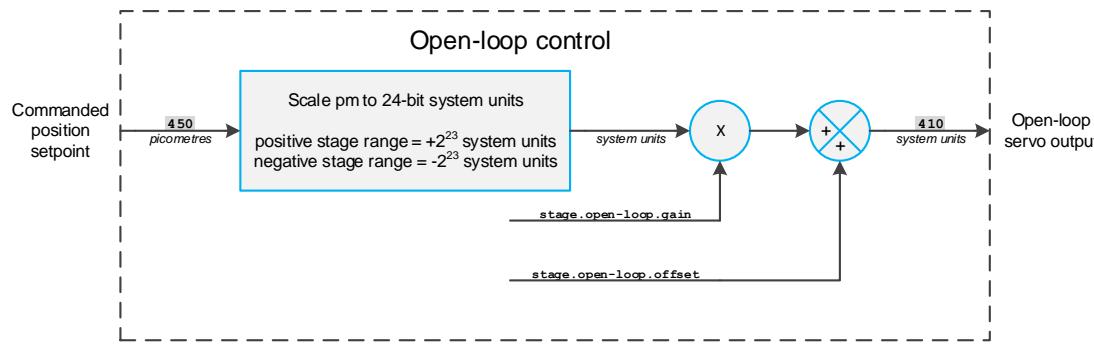
See 8.7 for description of functionality and details of commands.

8.2.7 Calculate velocity and acceleration



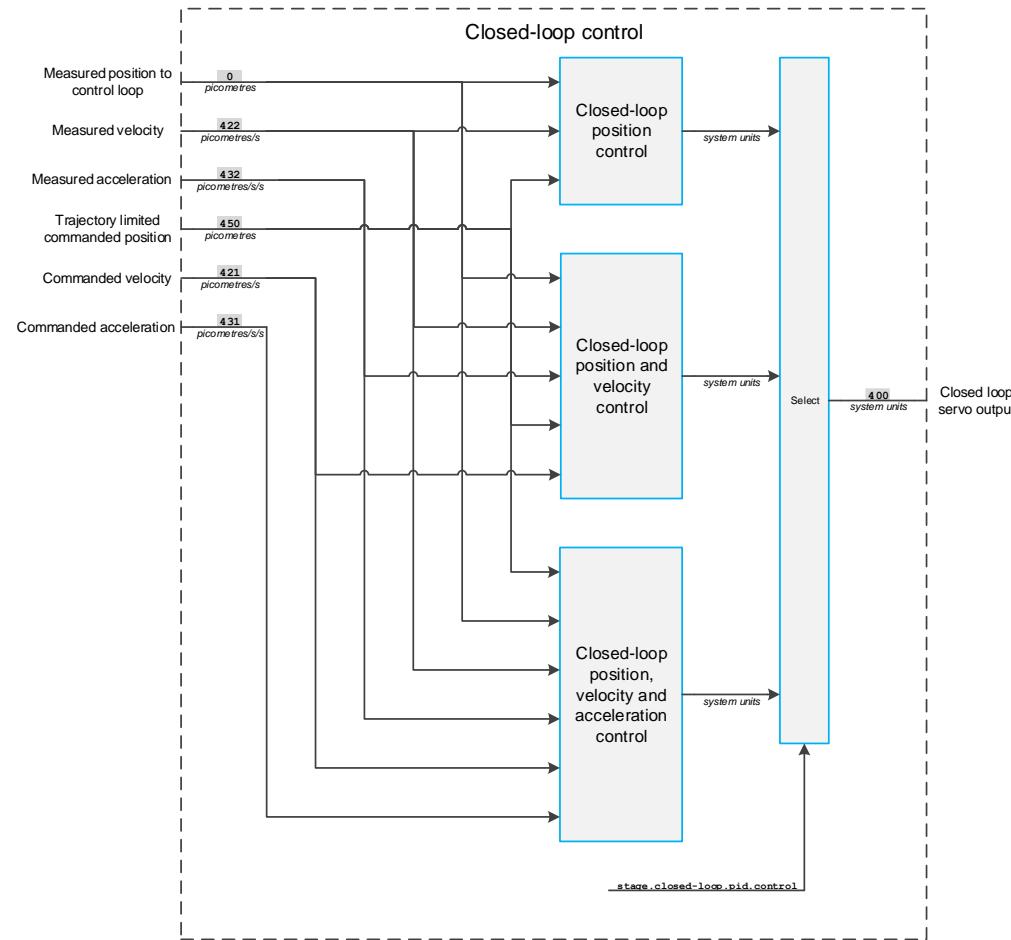
See 8.8 for description of functionality and details of commands.

8.2.8 Open-loop control



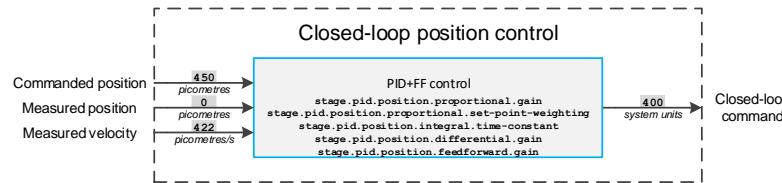
See 8.9 for description of functionality and details of commands.

8.2.9 Closed-loop control



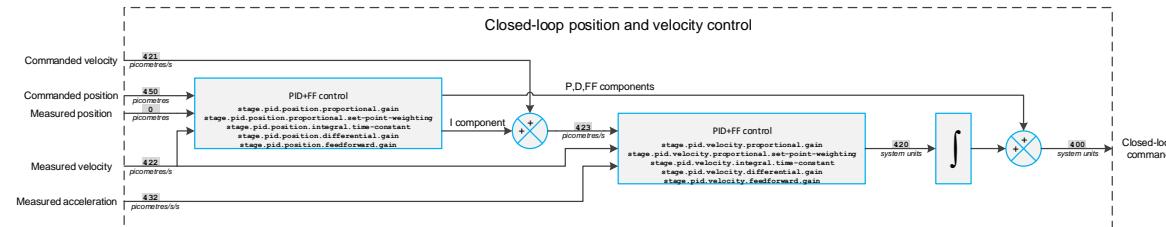
See 8.10 for description of functionality and details of commands.

8.2.10 Closed-loop position control



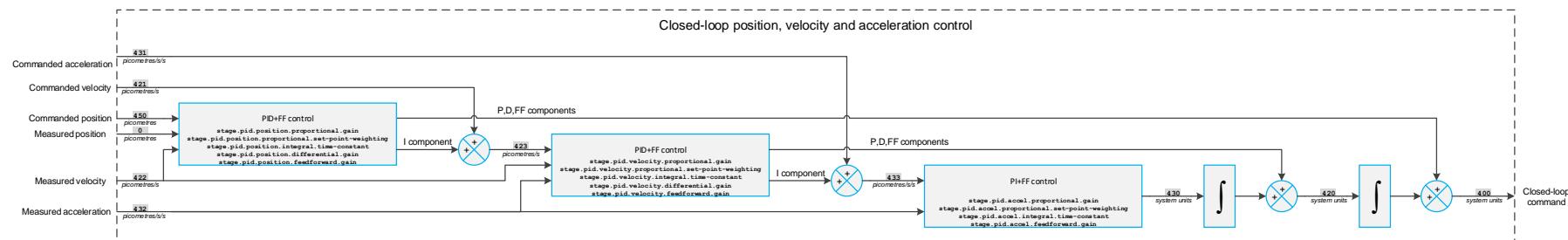
See 8.10 for description of functionality and details of commands.

8.2.11 Closed-loop position and velocity control



See 8.10 for description of functionality and details of commands.

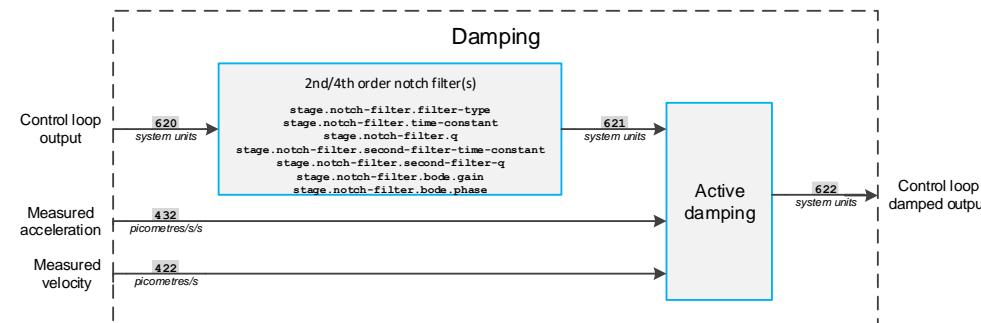
8.2.12 Closed-loop position, velocity and acceleration control



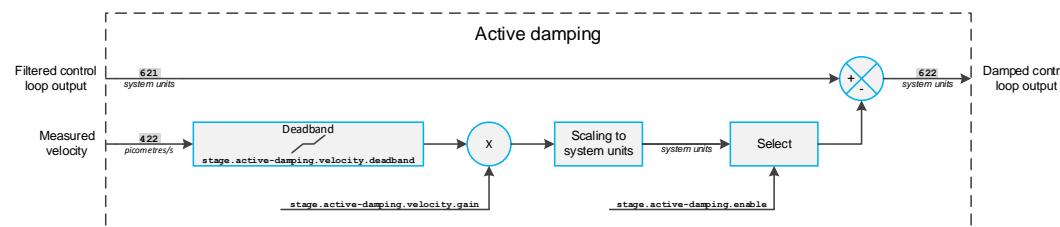
See 8.10 for description of functionality and details of commands.

8.2.13 Damping

See 8.11 for description of functionality and details of commands.

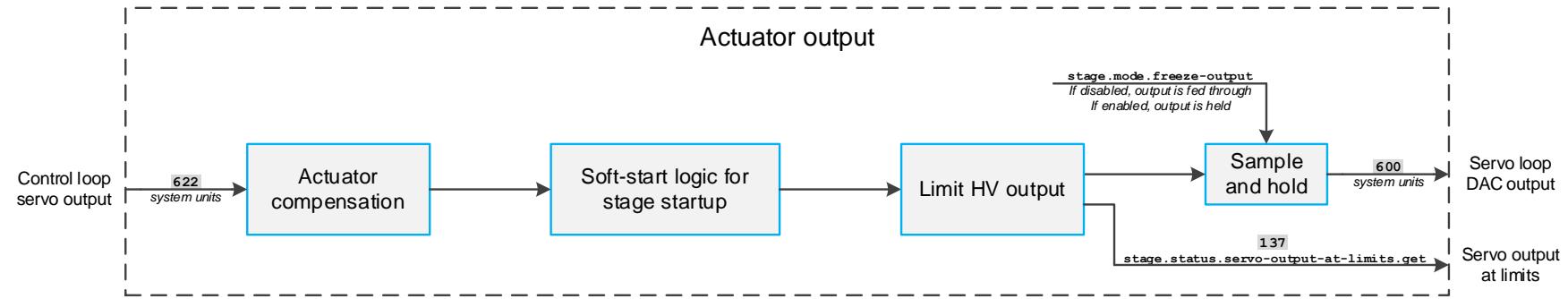


8.2.14 Active damping



See 8.11 for description of functionality and details of commands.

8.2.15 Actuator output



See 8.12 for description of functionality and details of commands.

8.3 Command processing

Command processing (see diagram 8.2.2) sums position commands from multiple sources to produce an overall absolute position command setpoint for the control loop. Multiple command sources may be active simultaneously.

These position commands may come from other areas of controller functionality. See 13.2 for details of stepped position command, 14 for details of function playback, and 17 for details of snapshot step. These require additional explanation which will not be duplicated here.

The overall absolute position command `stage.position.absolute-command` (see 8.3.1) may be set as well as read. When this is set, the digital position command `stage.position.command` (see 8.3.4) is set to the difference between this value and the sum of any other (enabled) commands, so that the absolute position command setpoint is set as specified.

The absolute position command has limits applied. This prevents excessively large commands being set, which can cause delays in command trajectory control (see 8.7) responding when the command returns to a more normal range.

If Nanobench or the interface API is used to set the digital position command `stage.position.command` such that the absolute position command has to be limited, or to set `stage.position.absolute-command` directly, the digital position command is set to the difference between the limited value and the sum of any other (enabled) commands. In both cases, the response to the initial “set” command will report the non-limited value set, and a subsequent “get” command will report the value when limiting has been applied. This allows users of the interface API to be aware of any command limiting taking place.

The analogue input command will inevitably have some noise present. Low-pass filtering may be added to reduce this noise. The filter cutoff frequency and filter order (1st-order to 4th-order) are configurable. A Bessel filter is used to ensure the filtered command has no overshoot. If analogue input commands are not required, noise on the absolute position command and hence also on the stage can be reduced by disabling the analogue input.

High-pass filters may optionally be used to briefly boost step changes in command and thereby give a faster response from the control loop. These are known as “edge boost filters”. In practise they are rarely used, because a faster overall step response is usually obtained by using feedforward command trajectory control (see 8.6 and diagram 8.2.6) to slow down edges and thereby reduce excitation of system resonances. In addition, it is usually better to calibrate the stage for a fast response by retuning the position PID (see 8.6 and diagram 8.2.9). The option is provided though, for systems which may benefit from it.

Because the analogue input is inherently noisy, the edge boost filters (if enabled) may require different settings for analogue and digital commands. Note that the absolute position command used by the in position check (see 8.5 and diagram 8.2.4) only considers the input commands and does not include the edge boost.

Bode plots for the low-pass and edge boost filters can be generated by requesting the gain and phase at frequencies of interest.

8.3.1 stage.position.absolute-command.get/set

8.3.1.1 Description

Stage absolute commanded position.

When this is set, the value which is actually set is `stage.position.command` (see 8.3.4). The value set for `stage.position.command` is the difference between the value set in this command and the sum of all other commands, such that `stage.position.absolute-command` is the desired value.

8.3.1.2 Command

```
stage.position.absolute-command.get <channel>
stage.position.absolute-command.set <channel> <value>
```

8.3.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	picometres			Stage absolute commanded position

8.3.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage absolute commanded position

8.3.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.1.7 Minimum security level

stage.position.absolute-command.get	No security required
stage.position.absolute-command.set	User level security required

8.3.1.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

8.3.2 stage.range.closed-loop-command.minimum.get

8.3.2.1 Description

Stage closed-loop command range minimum value

8.3.2.2 Command

```
stage.range.closed-loop-command.minimum.get <channel>
```

8.3.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage closed-loop command range minimum value

8.3.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.2.6 Minimum security level

stage.range.closed-loop-command.minimum.get	User level security required
---	------------------------------

8.3.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

8.3.3 stage.range.closed-loop-command.maximum.get

8.3.3.1 Description

Stage closed-loop command range maximum value

8.3.3.2 Command

```
stage.range.closed-loop-command.maximum.get <channel>
```

8.3.3.3 Parameters

Name	Type	Units	Maximum	Minimum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.3.4 Results

Name	Type	Units	Maximum	Minimum	Description
value	32-bit floating-point	picometres			Stage closed-loop command range maximum value

8.3.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.3.6 Maximum security level

stage.range.closed-loop-command.maximum.get	User level security required
---	------------------------------

8.3.3.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

8.3.4 stage.position.command.get/set

8.3.4.1 Description

Stage commanded position (digital position)

8.3.4.2 Command

```
stage.position.command.get <channel>
stage.position.command.set <channel> <value>
```

8.3.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	picometres			Stage commanded position

8.3.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage commanded position

8.3.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.4.7 Minimum security level

stage.position.command.get	No security required
stage.position.command.set	User level security required

8.3.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.5 stage.mode.analogue-command.get/set

8.3.5.1 Description

Enable analogue input position commands

8.3.5.2 Command

```
stage.mode.analogue-command.get <channel>
stage.mode.analogue-command.set <channel> <value>
```

8.3.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Enable analogue input position commands 1 = enable 0 = disable

8.3.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Enable analogue input position commands 1 = enable 0 = disable

8.3.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.5.7 Minimum security level

stage.mode.analogue-command.get	No security required
stage.mode.analogue-command.set	Superuser level security required

8.3.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.6 stage.command.analogue.low-pass-filter.enable.get/set

8.3.6.1 Description

Enable filtering on analogue input command

8.3.6.2 Command

stage.command.analogue.low-pass-filter.enable.get <channel>

stage.command.analogue.low-pass-filter.enable.set <channel> <value>

8.3.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	Boolean	0	1	Enable filtering

8.3.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	Boolean	0	1	Enable filtering

8.3.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Enable value is invalid (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.6.7 Minimum security level

stage.command.analogue.low-pass-filter.enable.get	User level security required
stage.command.analogue.low-pass-filter.enable.set	Superuser level security required

8.3.6.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.7 stage.command.analogue.low-pass-filter.time-constant.get/set**8.3.7.1 Description**

Analogue input command LPF time constant

8.3.7.2 Command

```
stage.command.analogue.low-pass-filter.time-constant.get <channel>
```

```
stage.command.analogue.low-pass-filter.time-constant.set <channel>
<value>
```

8.3.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.3.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.3.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.7.7 Minimum security level

stage.command.analogue.low-pass-filter.time-constant.get	User level security required
stage.command.analogue.low-pass-filter.time-constant.set	Superuser level security required

8.3.7.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.8 stage.command.analogue.low-pass-filter.filter-order.get/set**8.3.8.1 Description**

Order of filter on analogue input command

8.3.8.2 Command

```
stage.command.analogue.low-pass-filter.filter-order.get <channel>
```

```
stage.command.analogue.low-pass-filter.filter-order.set <channel>
<value>
```

8.3.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer		0	4	Order of filter 0=filter is disabled 1-4=1 st -order to 4 th -order filter selected

8.3.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer		0	4	Order of filter 0=filter is disabled 1-4=1 st -order to 4 th -order filter selected

8.3.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

Error return “errcode” string reported	Description
Value out of range	Low-pass filter order value is invalid (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.8.7 Minimum security level

stage.command.analogue.low-pass-filter.filter-order.get	User level security required
stage.command.analogue.low-pass-filter.filter-order.set	Superuser level security required

8.3.8.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.9 stage.command.analogue.low-pass-filter.bode.gain.get

8.3.9.1 Description

Analogue input command LPF Bode plot gain

8.3.9.2 Command

```
stage.command.analogue.low-pass-filter.bode.gain.get <channel>
<frequency>
```

8.3.9.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.3.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.3.9.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.9.6 Minimum security level

stage.command.analogue.low-pass-filter.bode.gain.get	User level security required
--	------------------------------

8.3.9.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.10 stage.command.analogue.low-pass-filter.bode.phase.get**8.3.10.1 Description**

Analogue input command LPF Bode plot phase

8.3.10.2 Command

```
stage.command.analogue.low-pass-filter.bode.phase.get <channel>
<frequency>
```

8.3.10.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.3.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.3.10.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.10.6 Minimum security level

stage.command.analogue.low-pass-filter.bode.phase.get	User level security required
---	------------------------------

8.3.10.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.11 stage.command.analogue.high-pass-filter.gain.get/set

8.3.11.1 Description

Analogue input command feed-forward high-pass filter gain

8.3.11.2 Command

```
stage.command.analogue.high-pass-filter.gain.get <channel>
```

```
stage.command.analogue.high-pass-filter.gain.set <channel> <value>
```

8.3.11.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.11.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		High-pass filter gain

8.3.11.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		High-pass filter gain

8.3.11.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.11.7 Minimum security level

stage.command.analogue.high-pass-filter.gain.get	User level security required
stage.command.analogue.high-pass-filter.gain.set	Superuser level security required

8.3.12 stage.command.analogue.high-pass-filter.time-constant.get/set**8.3.12.1 Description**

Analogue input command feed-forward high-pass filter time constant

8.3.12.2 Command

```
stage.command.analogue.high-pass-filter.time-constant.get <channel>
stage.command.analogue.high-pass-filter.time-constant.set <channel>
<value>
```

8.3.12.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.12.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	10	Filter time constant

8.3.12.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	10	Filter time constant

8.3.12.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.12.7 Minimum security level

stage.command.analogue.high-pass-filter.time-constant.get	User level security required
stage.command.analogue.high-pass-filter.time-constant.set	Superuser level security required

8.3.12.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.13 stage.command.analogue.high-pass-filter.q.get/set

8.3.13.1 Description

Analogue input command feed-forward high-pass filter Q factor

8.3.13.2 Command

```
stage.command.analogue.high-pass-filter.q.get <channel>  
stage.command.analogue.high-pass-filter.q.set <channel> <value>
```

8.3.13.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.13.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		Filter Q factor

8.3.13.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Filter Q factor

8.3.13.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Q factor must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.13.7 Minimum security level

stage.command.analogue.high-pass-filter.q.get	User level security required
stage.command.analogue.high-pass-filter.q.set	Superuser level security required

8.3.13.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.14 stage.command.analogue.high-pass-filter.bode.gain.get

8.3.14.1 Description

Analogue input command edge boost high-pass filter Bode plot gain

8.3.14.2 Command

```
stage.command.analogue.high-pass-filter.bode.gain.get <channel>
<frequency>
```

8.3.14.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.3.14.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.3.14.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.14.6 Minimum security level

stage.command.analogue.high-pass-filter.bode.gain.get	User level security required
---	------------------------------

8.3.14.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.15 stage.command.analogue.high-pass-filter.bode.phase.get

8.3.15.1 Description

Analogue input command edge boost high-pass filter Bode plot phase

8.3.15.2 Command

```
stage.command.analogue.high-pass-filter.bode.phase.get <channel>
<frequency>
```

8.3.15.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.3.15.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.3.15.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.15.6 Minimum security level

stage.command.analogue.high-pass-filter.bode.phase.get	User level security required
--	------------------------------

8.3.15.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.16 stage.command.digital.high-pass-filter.gain.get/set

8.3.16.1 Description

Digital command feed-forward high-pass filter gain

8.3.16.2 Command

```
stage.command.digital.high-pass-filter.gain.get <channel>
```

```
stage.command.digital.high-pass-filter.gain.set <channel> <value>
```

8.3.16.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.16.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		High-pass filter gain

8.3.16.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		High-pass filter gain

8.3.16.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.16.7 Minimum security level

stage.command.digital.high-pass-filter.gain.get	User level security required
stage.command.digital.high-pass-filter.gain.set	Superuser level security required

8.3.16.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.17 stage.command.digital.high-pass-filter.time-constant.get/set**8.3.17.1 Description**

Digital command feed-forward high-pass filter time constant

8.3.17.2 Command

```
stage.command.digital.high-pass-filter.time-constant.get <channel>
stage.command.digital.high-pass-filter.time-constant.set <channel>
<value>
```

8.3.17.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.17.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.3.17.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.3.17.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.17.7 Minimum security level

stage.command.digital.high-pass-filter.time-constant.get	User level security required
stage.command.digital.high-pass-filter.time-constant.set	Superuser level security required

8.3.17.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.18 stage.command.digital.high-pass-filter.q.get/set

8.3.18.1 Description

Digital command feed-forward high-pass filter Q factor

8.3.18.2 Command

stage.command.digital.high-pass-filter.q.get <channel>

stage.command.digital.high-pass-filter.q.set <channel> <value>

8.3.18.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.3.18.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		Filter Q factor

8.3.18.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Filter Q factor

8.3.18.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Q factor must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.18.7 Minimum security level

stage.command.digital.high-pass-filter.q.get	User level security required
stage.command.digital.high-pass-filter.q.set	Superuser level security required

8.3.18.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.3.19 stage.command.digital.high-pass-filter.bode.gain.get

8.3.19.1 Description

Digital input command edge boost high-pass filter Bode plot gain

8.3.19.2 Command

```
stage.command.digital.high-pass-filter.bode.gain.get <channel>
<frequency>
```

8.3.19.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.3.19.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.3.19.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.19.6 Minimum security level

stage.command.digital.high-pass-filter.bode.gain.get	User level security required
--	------------------------------

8.3.19.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.3.20 stage.command.digital.high-pass-filter.bode.phase.get

8.3.20.1 Description

Digital input command edge boost high-pass filter Bode plot phase

8.3.20.2 Command

```
stage.command.digital.high-pass-filter.bode.phase.get <channel>
<frequency>
```

8.3.20.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.3.20.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.3.20.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.3.20.6 Minimum security level

stage.command.digital.high-pass-filter.bode.phase.get	User level security required
---	------------------------------

8.3.20.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4 Measured position processing

8.4.1 stage.position.measured.get

8.4.1.1 Description

Stage measured position

8.4.1.2 Command

```
stage.position.measured.get <channel>
```

8.4.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage measured position

8.4.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

8.4.1.6 Minimum security level

stage.position.measured.get	No security required
-----------------------------	----------------------

8.4.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.4.2 stage.position.low-pass-filter.enable.get/set

8.4.2.1 Description

Enable filtering on position used by control loop

8.4.2.2 Command

```
stage.position.low-pass-filter.enable.get <channel>
stage.position.low-pass-filter.enable.set <channel> <value>
```

8.4.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	Boolean	0	1	Enable filtering

8.4.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	Boolean	0	1	Enable filtering

8.4.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Enable value is invalid (set only)

8.4.2.7 Minimum security level

stage.position.low-pass-filter.enable.get	User level security required
stage.position.low-pass-filter.enable.set	Superuser level security required

8.4.2.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.3 stage.position.low-pass-filter.time-constant.get/set

8.4.3.1 Description

Position input LPF time constant

8.4.3.2 Command

```
stage.position.low-pass-filter.time-constant.get <channel>
```

```
stage.position.low-pass-filter.time-constant.set <channel> <value>
```

8.4.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.4.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.4.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)

8.4.3.7 Minimum security level

stage.position.low-pass-filter.time-constant.get	User level security required
stage.position.low-pass-filter.time-constant.set	Superuser level security required

8.4.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.4.4 stage.position.low-pass-filter.filter-order.get/set**8.4.4.1 Description**

Order of filter on position

8.4.4.2 Command

```
stage.position.low-pass-filter.filter-order.get <channel>
```

```
stage.position.low-pass-filter.filter-order.set <channel> <value>
```

8.4.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer		0	4	Order of filter 0=filter is disabled 1-4=1 st -order to 4 th -order filter selected

8.4.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer		0	4	Order of filter 0=filter is disabled 1-4=1 st -order to 4 th -order filter selected

8.4.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

Error return “errcode” string reported	Description
Value out of range	Low-pass filter order value is invalid (set only)

8.4.4.7 Minimum security level

stage.position.low-pass-filter.filter-order.get	User level security required
stage.position.low-pass-filter.filter-order.set	Superuser level security required

8.4.4.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.5 stage.position.low-pass-filter.bode.gain.get

8.4.5.1 Description

Position filter Bode plot gain

8.4.5.2 Command

```
stage.position.low-pass-filter.bode.gain.get <channel> <frequency>
```

8.4.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.4.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.4.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range

8.4.5.6 Minimum security level

stage.position.low-pass-filter.bode.gain.get	User level security required
--	------------------------------

8.4.5.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.6 stage.position.low-pass-filter.bode.phase.get

8.4.6.1 Description

Position filter Bode plot phase

8.4.6.2 Command

```
stage.position.low-pass-filter.bode.phase.get <channel> <frequency>
```

8.4.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.4.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.4.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range

8.4.6.6 Minimum security level

stage.position.low-pass-filter.bode.phase.get	User level security required
---	------------------------------

8.4.6.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.7 stage.position.output.low-pass-filter.enable.get/set

8.4.7.1 Description

Enable filtering on position sent to position monitor output

8.4.7.2 Command

```
stage.position.output.low-pass-filter.enable.get <channel>
```

```
stage.position.output.low-pass-filter.enable.set <channel> <value>
```

8.4.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	Boolean	0	1	Enable filtering

8.4.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	Boolean	0	1	Enable filtering

8.4.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Enable value is invalid (set only)

8.4.7.7 Minimum security level

stage.position.output.low-pass-filter.enable.get	User level security required
stage.position.output.low-pass-filter.enable.set	Superuser level security required

8.4.7.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.8 stage.position.output.low-pass-filter.time-constant.get/set**8.4.8.1 Description**

Position output LPF time constant

8.4.8.2 Command

```
stage.position.output.low-pass-filter.time-constant.get <channel>
```

```
stage.position.output.low-pass-filter.time-constant.set <channel>
<value>
```

8.4.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.4.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.4.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)

8.4.8.7 Minimum security level

stage.position.output.low-pass-filter.time-constant.get	User level security required
stage.position.output.low-pass-filter.time-constant.set	Superuser level security required

8.4.8.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.9 stage.position.output.low-pass-filter.filter-order.get/set

8.4.9.1 Description

Order of filter on position output

8.4.9.2 Command

```
stage.position.output.low-pass-filter.filter-order.get <channel>
```

```
stage.position.output.low-pass-filter.filter-order.set <channel>
<value>
```

8.4.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.4.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer		0	4	Order of filter 0=filter is disabled 1-4=1 st -order to 4 th -order filter selected

8.4.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer		0	4	Order of filter 0=filter is disabled 1-4=1 st -order to 4 th -order filter selected

8.4.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

Error return “errcode” string reported	Description
Value out of range	Low-pass filter order value is invalid (set only)

8.4.9.7 Minimum security level

stage.position.output.low-pass-filter.filter-order.get	User level security required
stage.position.output.low-pass-filter.filter-order.set	Superuser level security required

8.4.9.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.10 stage.position.output.low-pass-filter.bode.gain.get

8.4.10.1 Description

Position output filter Bode plot gain

8.4.10.2 Command

```
stage.position.output.low-pass-filter.bode.gain.get <channel>
<frequency>
```

8.4.10.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.4.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.4.10.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range

8.4.10.6 Minimum security level

stage.position.output.low-pass-filter.bode.gain.get	User level security required
---	------------------------------

8.4.10.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.11 stage.position.output.low-pass-filter.bode.phase.get

8.4.11.1 Description

Position output filter Bode plot phase

8.4.11.2 Command

```
stage.position.output.low-pass-filter.bode.phase.get <channel>
<frequency>
```

8.4.11.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.4.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.4.11.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range

8.4.11.6 Minimum security level

stage.position.output.low-pass-filter.bode.phase.get	User level security required
--	------------------------------

8.4.11.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.12 stage.position.output.low-pass-filter.bode.combined-gain.get

8.4.12.1 Description

Position output filter combined Bode plot gain

8.4.12.2 Command

```
stage.position.output.low-pass-filter.bode.combined-gain.get <channel>
<frequency>
```

8.4.12.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.4.12.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.4.12.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range

8.4.12.6 Minimum security level

stage.position.output.low-pass-filter.bode.combined-gain.get	User level security required
--	------------------------------

8.4.12.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.4.13 stage.position.output.low-pass-filter.bode.combined-phase.get

8.4.13.1 Description

Position output filter combined Bode plot phase

8.4.13.2 Command

```
stage.position.output.low-pass-filter.bode.combined-phase.get  
<channel> <frequency>
```

8.4.13.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.4.13.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.4.13.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range

8.4.13.6 Minimum security level

stage.position.output.low-pass-filter.bode.combined-phase.get	User level security required
---	------------------------------

8.4.13.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.5 In position check

The in position (“ready”) check notifies the user that the stage position has reached the desired position after moving, within some tolerance (see diagram 8.2.4). This status is available as an digital output, and is frequently used by external equipment to schedule measurements.

The in-position check only runs when the control loop for the channel is in closed-loop mode (see 8.6.1, `stage.mode.closed-loop`). In open-loop mode, the command does not relate to an actual position, so the in position status/output is always false or logic-low in open-loop mode.

The threshold for reporting in-position is configurable. This will frequently relate to a tolerance in the system, for example a half-wavelength of a specific frequency of light in microscopy applications.

Two methods of debouncing/confirming the in-position status are provided. The “low-pass filter” debouncing method is used by default, and is normally recommended. This simply uses a low-pass filter to average (or smooth) the position error. This method has the advantage that large position errors will more quickly cause the check to report that the stage is out of position.

However this can be a disadvantage when the stage has reached the desired position, because a large error causes a longer delay in reporting

The “window filter” debouncing method is preferred by certain customers. This considers only whether the stage is in position for a processing step, and debounces this by requiring the stage to have been in position for some fraction of the number of steps checked. This method has the advantage that the debouncing delay is not affected by the magnitude of the error. However it has the disadvantage that it can report that the stage is in position when the stage is overshooting, potentially affecting the accuracy of subsequent measurements by other equipment.

If a stage position is commanded which is outside the achievable range of the actuator, the “in position” checks will always report that the stage is not in position. For some applications, it is more useful to know whether the stage is moving than to know whether the commanded position has been achieved. For these applications, an “is moving” status is provided. The stage is considered to be not moving if it is in position or if it has reached the end of its travel.

8.5.1 stage.in-position.error-threshold.get/set

8.5.1.1 Description

Stage “in position” error threshold (ready limit)

8.5.1.2 Command

```
stage.in-position.error-threshold.get <channel>
```

```
stage.in-position.error-threshold.set <channel> <value>
```

8.5.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	picometres	0		Stage “in position” error threshold (ready limit)

8.5.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Stage “in position” error threshold (ready limit)

8.5.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Error threshold must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.1.7 Minimum security level

stage.in-position.error-threshold.get	User level security required
stage.in-position.error-threshold.set	Superuser level security required

8.5.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.2 stage.in-position.lpf.time-constant.get/set

8.5.2.1 Description

Stage “in position” confirmation low-pass filter time constant

8.5.2.2 Command

```
stage.in-position.lpf.time-constant.get <channel>
```

```
stage.in-position.lpf.time-constant.set <channel> <value>
```

8.5.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Stage “in position” confirmation low-pass filter time constant

8.5.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Stage “in position” confirmation low-pass filter time constant

8.5.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.2.7 Minimum security level

stage.in-position.lpf.time-constant.get	User level security required
stage.in-position.lpf.time-constant.set	Superuser level security required

8.5.2.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.3 stage.in-position.window-filter.size.get/set

8.5.3.1 Description

Stage “in position” confirmation window filter size

8.5.3.2 Command

```
stage.in-position.window-filter.size.get <channel>
stage.in-position.window-filter.size.set <channel> <value>
```

8.5.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	DSP loop ticks	1	1200	Stage “in position” confirmation window filter size

8.5.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	DSP loop ticks	1	1200	Stage “in position” confirmation window filter size

8.5.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Window filter size must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.3.7 Minimum security level

stage.in-position.window-filter.size.get	User level security required
stage.in-position.window-filter.size.set	Superuser level security required

8.5.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.4 stage.in-position.window-filter.valid-threshold.get/set**8.5.4.1 Description**

Stage “in position” confirmation window filter valid count threshold to set debounced output

8.5.4.2 Command

```
stage.in-position.window-filter.valid-threshold.get <channel>
```

```
stage.in-position.window-filter.valid-threshold.set <channel> <value>
```

8.5.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	DSP loop ticks	1	1200	Stage “in position” confirmation window filter valid count threshold

8.5.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	DSP loop ticks	1	1200	Stage “in position” confirmation window filter valid count threshold

8.5.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Window filter valid count threshold must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.4.7 Minimum security level

stage.in-position.window-filter.size.get	User level security required
stage.in-position.window-filter.size.set	Superuser level security required

8.5.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.5 stage.status.in-position.unconfirmed.get

8.5.5.1 Description

Stage unconfirmed “in position” state

8.5.5.2 Command

```
stage.status.in-position.unconfirmed.get <channel>
```

8.5.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage unconfirmed “in position” state

8.5.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.5.6 Minimum security level

stage.status.in-position.unconfirmed.get	No security required
--	----------------------

8.5.5.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.6 stage.status.in-position.lpf-confirmed.get

8.5.6.1 Description

Stage “in position” state confirmed by LPF algorithm

8.5.6.2 Command

stage.status.in-position.lpf-confirmed.get <channel>

8.5.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage “in position” state confirmed by LPF algorithm

8.5.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.6.6 Minimum security level

stage.status.in-position.lpf-confirmed.get	No security required
--	----------------------

8.5.6.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.7 stage.status.in-position.window-filter-confirmed.get

8.5.7.1 Description

Stage “in position” state confirmed by window filter algorithm

8.5.7.2 Command

```
stage.status.in-position.window-filter-confirmed.get <channel>
```

8.5.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage “in position” state confirmed by window filter algorithm

8.5.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.7.6 Minimum security level

stage.status.in-position.window-filter-confirmed.get	No security required
--	----------------------

8.5.7.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.8 stage.mode.in-position-output-select.get/set

8.5.8.1 Description

Select stage “in position” digital output to be confirmed by low-pass filter or window filter algorithm

8.5.8.2 Command

```
stage.mode.in-position-output-select.get <channel>
```

```
stage.mode.in-position-output-select.set <channel> <value>
```

8.5.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Select confirmation algorithm 1 = window filter confirmation selected 0 = LPF confirmation selected

8.5.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Select confirmation algorithm 1 = window filter confirmation selected 0 = LPF confirmation selected

8.5.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.8.7 Minimum security level

stage.mode.in-position-output-select.get	No security required
stage.mode.in-position-output-select.get	Superuser level security required

8.5.8.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.5.9 stage.status.stage-moving.get

8.5.9.1 Description

Stage is currently moving

8.5.9.2 Command

```
stage.status.stage-moving.get <channel>
```

8.5.9.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.5.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage is currently moving

8.5.9.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.5.9.6 Minimum security level

stage.status.stage-moving.get	No security required
-------------------------------	----------------------

8.5.9.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

8.6 Control loop

The control loop provides the key functionality of the controller (see diagram 8.2.5 and sub-diagrams).

Typically the controller will normally be operated in closed-loop mode. However it can also be operated in open-loop mode if required, with an external control system providing closed-loop control using other sensors. Stage damping is available in both closed-loop and open-loop modes.

Switching from closed-loop to open-loop mode will typically result in an instantaneous step to the new commanded value.

Switching from open-loop mode to closed-loop mode is stepless. The controller simply treats the difference between the current position and the new commanded position as any other position error, and uses closed-loop control to bring the system smoothly to the commanded position.

8.6.1 stage.mode.closed-loop.get/set

8.6.1.1 Description

Select closed-loop or open-loop operation

8.6.1.2 Command

```
stage.mode.closed-loop.get <channel>  
stage.mode.closed-loop.set <channel> <value>
```

8.6.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.6.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Select closed/open-loop operation 1 = closed-loop 0 = open-loop

8.6.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Select closed/open-loop operation 1 = closed-loop 0 = open-loop

8.6.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.6.1.7 Minimum security level

stage.mode.closed-loop.get	No security required
stage.mode.closed-loop.set	Superuser level security required

8.6.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

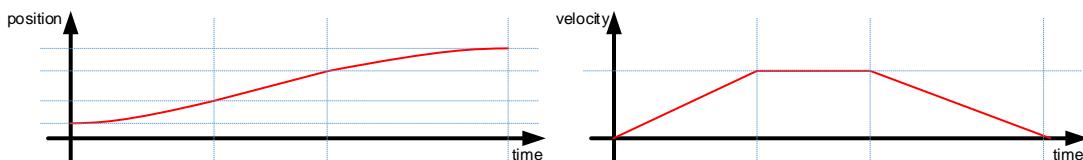
8.7 Feedforward command trajectory control

Command trajectory control provides feedforward move shaping using trapezoidal velocity limiting (see diagram 8.2.6).

High accelerations and decelerations will often excite system resonances. This includes stage resonances of course, but typically the most significant resonances occur in the wider system, where mounting structures will have their own resonances. To avoid exciting these resonances, command trajectory control allows limits to be placed to be placed on commanded acceleration and deceleration. Whilst this makes the initial stage movement slightly slower, the stage will tend to settle to the commanded position in a shorter time because the system is not resonating.

Resonances excited during acceleration may die away naturally during movement, but resonances excited during deceleration will immediately affect settling to the commanded position. Separate limits are therefore provided for acceleration and deceleration, where the deceleration limit is generally set lower than acceleration.

When the commanded position changes faster than the stage can move, the resulting position error causes the position PID integral term to ramp up. This tends to produce overshoot, which again causes the stage to take longer to settle to the commanded position. Command trajectory control allows a limit to be placed on the commanded speed, which typically is set close to the maximum speed at which the stage can move, ensuring the position error remains small during the move. This prevents the position PID integral term from ramping up to the same degree, and hence reduces the degree of overshoot. Again, this may make the initial stage movement slightly slower, but the time taken to settle to the commanded position is reduced.



Controlling acceleration, speed and deceleration creates what is known as a “trapezoidal move profile”. Traditionally this would be provided by a separate motion control module or PC software, and moves often need to be configured in advance. The Queensgate controller provides this as a built-in feature running in real-time, so that all command sources can benefit from this.

This is termed “feedforward” move shaping because these limits on acceleration, speed and deceleration are applied only to the commanded position (and hence commanded velocity and acceleration). The system is not feeding back measured velocity and acceleration and taking some action when actual velocity or acceleration exceeds acceptable limits. Instead, closed-loop control (see section 8.10 and diagram 8.2.9) is allowed to track this move-shaped command using a tuning which is considered acceptable. The actual velocity and acceleration may overshoot past these limits as the control loop attempts to track the command. In general though this gives a better system response compared to hard-limiting velocity and acceleration, which tends to produce greater jerk and thereby excite system resonances.

The intention of command trajectory control is to improve the final settle to a commanded position, so the design assumes that launch acceleration will not be slower than braking deceleration. If the launch acceleration is set slower than the braking deceleration, the controller limits the launch acceleration to the braking deceleration rate.

8.7.1 stage.command-trajectory.enable.get/set

8.7.1.1 Description

Enable/disable command trajectory limiting

8.7.1.2 Command

```
stage.command-trajectory.enable.get <channel>  
stage.command-trajectory.enable.set <channel> <value>
```

8.7.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.7.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Enable/disable command trajectory limiting

8.7.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Enable/disable command trajectory limiting

8.7.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller; or channel parameter is valid but no stage is connected to the specified channel
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.7.1.7 Minimum security level

stage.command-trajectory.enable.get	User level security required
stage.command-trajectory.enable.set	Superuser level security required

8.7.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.7.2 stage.command-trajectory.launch-acceleration.get/set

8.7.2.1 Description

Command trajectory maximum launch acceleration limit

8.7.2.2 Command

```
stage.command-trajectory.launch-acceleration.get <channel>
```

```
stage.command-trajectory.launch-acceleration.set <channel> <value>
```

8.7.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.7.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	nm/ms/ms	0		Launch acceleration limit

8.7.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	nm/ms/ms	0		Launch acceleration limit

8.7.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Launch acceleration limit must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.7.2.7 Minimum security level

stage.command-trajectory.launch-acceleration.get	User level security required
stage.command-trajectory.launch-acceleration.set	Superuser level security required

8.7.2.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.7.3 stage.command-trajectory.speed.get/set

8.7.3.1 Description

Command trajectory maximum speed limit

8.7.3.2 Command

```
stage.command-trajectory.speed.get <channel>
stage.command-trajectory.speed.set <channel> <value>
```

8.7.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer	picometres			Controller channel for stage

8.7.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	nm/ms	0		Maximum speed limit

8.7.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	nm/ms	0		Maximum speed limit

8.7.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Maximum speed limit must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.7.3.7 Minimum security level

stage.command-trajectory.speed.get	User level security required
stage.command-trajectory.speed.set	Superuser level security required

8.7.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.7.4 stage.command-trajectory.braking-deceleration.get/set

8.7.4.1 Description

Command trajectory maximum braking deceleration limit

8.7.4.2 Command

```
stage.command-trajectory.braking-deceleration.get <channel>
```

```
stage.command-trajectory.braking-deceleration.set <channel> <value>
```

8.7.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.7.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	nm/ms/ms	0		Launch acceleration limit

8.7.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	nm/ms/ms	0		Launch acceleration limit

8.7.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Launch acceleration limit must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.7.4.7 Minimum security level

stage.command-trajectory.braking-deceleration.get	User level security required
stage.command-trajectory.braking-deceleration.set	Superuser level security required

8.7.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.8 Calculate velocity and acceleration

In order to carry out closed-loop control of velocity and acceleration, and to carry out active damping, these signals must be calculated (see diagram 8.2.7). Differentiation of commanded and measured positions provides velocity values, and differentiation of these velocities provides acceleration values.

Differentiation inherently amplifies noise at higher frequencies. To produce velocity and acceleration signals which are as accurate as possible with little noise, the differentiation step is combined with a 2nd-order low-pass filter to reduce this higher-frequency noise and leave the desired signal. A Bessel filter is used to ensure the filtered measurement has no overshoot. Unlike other low-pass filters in the controller, the filter order here is not selectable: a 1st-order filter does not reduce the higher-frequency noise sufficiently, making double-differentiation unacceptably noisy; and higher-order filters produce undesirable additional phase shift on the signals.

The same filter is used for calculation of velocity and acceleration. It is not desirable to have a separate cutoff frequency for acceleration, because this would give a more “lumpy” response to the resulting filter characteristics.

As usual, the filter phase shift becomes more significant as the filter cutoff frequency is reduced. The phase shift on measured velocity and acceleration will affect the stability of closed-loop control when these are used, particularly in PID differential terms, and also the stability of active damping. Care must therefore be taken to ensure the filter cutoff frequency is set as high as possible whilst still providing acceptably low-noise velocity and acceleration signals.

The phase shift for measured velocity and acceleration is also affected by any low-pass filtering applied to measured position (see 8.4 and diagram 8.2.3). When considering the phase shift for these signals, this must also be included. When using closed-loop control of velocity and acceleration, or when using active damping, this filter must often be adjusted to give velocity and acceleration signals with acceptable phase response, even if this results in higher noise on the measured position.

The controller provides three Bode plots for this process, reporting:

- The Bode plot of the basic differentiator and 2nd-order low-pass filter. Note that the passband gain for the differentiator is a continuously-ascending line, unlike a normal low-pass filter where the passband gain is a horizontal line. Note also that due to differentiation, the passband phase is +90°.
- The “combined” Bode plot for measured velocity, combining the low-pass filter from measured position and the differentiator and low-pass filter from this section.
- The “combined” Bode plot for measured acceleration, combining the low-pass filter from measured position and two iterations of the differentiator and low-pass filter from this section.

Some control systems use a second sensor to measure acceleration, most notably when implementing active damping. With high quality filtering on the differentiation process, similar results are achieved without the extra cost and complexity of an additional sensor. Whilst the differentiation process does introduce phase shift, this is well-controlled and easily understood, whereas adding a second sensor frequently introduces problems of timing issues between the measurements.

8.8.1 stage.position.velocity-accel-filter.time-constant.get/set**8.8.1.1 Description**

Velocity and acceleration calculation low-pass filter cutoff time constant

8.8.1.2 Command

```
stage.position.velocity-accel-filter.time-constant.get <channel>
stage.position.velocity-accel-filter.time-constant.set <channel>
<value>
```

8.8.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.8.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.8.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Filter time constant

8.8.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.1.7 Minimum security level

stage.position.velocity-accel-filter.time-constant.get	User level security required
stage.position.velocity-accel-filter.time-constant.set	Superuser level security required

8.8.1.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.8.2 stage.position.velocity-accel-filter.bode.gain.get

8.8.2.1 Description

Velocity and acceleration calculation differentiator with LPF Bode plot gain

8.8.2.2 Command

```
stage.position.velocity-accel-filter.bode.gain.get <channel>
<frequency>
```

8.8.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.8.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.8.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.2.6 Minimum security level

stage.position.velocity-accel-filter.bode.gain.get	User level security required
--	------------------------------

8.8.2.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.8.3 stage.position.velocity-accel-filter.bode.phase.get

8.8.3.1 Description

Velocity and acceleration calculation differentiator with LPF Bode plot phase

8.8.3.2 Command

```
stage.position.velocity-accel-filter.bode.phase.get <channel>
<frequency>
```

8.8.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.8.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.8.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.3.6 Minimum security level

stage.position.velocity-accel-filter.bode.phase.get	User level security required
---	------------------------------

8.8.3.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.8.4 stage.position.velocity-accel-filter.bode.velocity.combined-gain.get**8.8.4.1 Description**

Velocity calculation differentiator with LPF combined Bode plot gain

8.8.4.2 Command

```
stage.position.velocity-accel-filter.bode.velocity.combined-gain.get
<channel> <frequency>
```

8.8.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.8.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.8.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.4.6 Minimum security level

stage.position.velocity-accel-filter.bode.velocity.combined-gain.get	User level security required
--	------------------------------

8.8.4.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.8.5 stage.position.velocity-accel-filter.bode.velocity.combined-phase.get**8.8.5.1 Description**

Velocity calculation differentiator with LPF combined Bode plot phase

8.8.5.2 Command

```
stage.position.velocity-accel-filter.bode.velocity.combined-phase.get
<channel> <frequency>
```

8.8.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.8.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.8.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.5.6 Minimum security level

stage.position.velocity-accel-filter.bode.velocity.combined-phase.get	User level security required
---	------------------------------

8.8.5.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.8.6 stage.position.velocity-accel-filter.bode.accel.combined-gain.get**8.8.6.1 Description**

Accel calculation differentiator with LPF combined Bode plot gain

8.8.6.2 Command

```
stage.position.velocity-accel-filter.bode.accel.combined-gain.get
<channel> <frequency>
```

8.8.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.8.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.8.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.6.6 Minimum security level

stage.position.velocity-accel-filter.bode.accel.combined-gain.get	User level security required
---	------------------------------

8.8.6.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.8.7 stage.position.velocity-accel-filter.bode.accel.combined-phase.get**8.8.7.1 Description**

Accel calculation differentiator with LPF combined Bode plot phase

8.8.7.2 Command

```
stage.position.velocity-accel-filter.bode.velocity.combined-phase.get
<channel> <frequency>
```

8.8.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.8.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.8.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.8.7.6 Minimum security level

stage.position.velocity-accel-filter.bode.accel.combined-phase.get	User level security required
--	------------------------------

8.8.7.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.9 Open-loop control

Open-loop control of stage movement (see diagram 8.2.8) is available for use in applications where closed-loop control is carried out by external equipment.

Whilst many of the features of the controller are redundant in open-loop mode, the controller is still able to carry out damping of resonances (see diagram 8.2.13) which can improve the overall closed-loop performance. This can typically be carried out more effectively by the controller than by an external control system.

8.9.1 stage.open-loop.gain.get/set

8.9.1.1 Description

Open-loop control scaling gain

8.9.1.2 Command

```
stage.open-loop.gain.get <channel>  
stage.open-loop.gain.set <channel> <value>
```

8.9.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.9.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point				Open-loop control scaling gain

8.9.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point				Open-loop control scaling gain

8.9.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.9.1.7 Minimum security level

stage.open-loop.gain.get	User level security required
stage.open-loop.gain.set	Superuser level security required

8.9.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.9.2 stage.open-loop.offset.get/set

8.9.2.1 Description

Open-loop control scaling offset

8.9.2.2 Command

```
stage.open-loop.offset.get <channel>  
stage.open-loop.offset.set <channel> <value>
```

8.9.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.9.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	system units	-16777215	16777215	Open-loop control scaling offset

8.9.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	system units	-16777215	16777215	Open-loop control scaling offset

8.9.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Offset must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.9.2.7 Minimum security level

stage.open-loop.offset.get	User level security required
stage.open-loop.offset.set	Superuser level security required (set only)

8.9.2.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.9.3 stage.open-loop.output-offset-gain.get

8.9.3.1 Description

Approximate scaling from actuator desired movement in picometres to required open-loop output in bits

8.9.3.2 Command

```
stage.open-loop.output-offset-gain.get <channel>
```

8.9.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.9.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point				Output scaling

8.9.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.9.3.6 Minimum security level

stage.open-loop.output-offset-gain.get	No security required
--	----------------------

8.9.3.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.10 Closed-loop control

Closed-loop operation (see diagram 8.2.9) is intended to be the primary operating mode for the controller. The controller can carry out closed-loop control of position only (see diagram 8.2.10), of position and velocity (see diagram 8.2.11), or of position, velocity and acceleration (see diagram 8.2.12).

Users who intend to tune closed-loop control for their applications are expected to be familiar with PID loops. However the techniques used with this nested PID architecture may be unfamiliar to some users. Section 9 describes how to configure and tune closed-loop control in much greater detail.

8.10.1 Closed-loop control design

As is industry-standard for closed-loop operation, this is implemented using nested PID loops with feedforward (PID+FF). Calculated velocity signals (see diagram 8.2.7 and section 8.8) are used for velocity control, and the measured velocity is also used as the derivative (D) term for the position PID loop. Similarly, calculated acceleration signals are used for acceleration control, and the measured acceleration is also used as the derivative (D) term for the velocity PID loop. The acceleration loop is actually a PI+FF loop since it lacks a derivative (D) term: differentiating position twice to give velocity and acceleration produces acceptable results, but a third differentiation step to give jerk is dominated by noise and cannot be used.

The PID+FF architecture provides the following key features. The position loop is described here, but the same principles and architecture apply to the inner loops too.

- Feedforward control takes advantage of the near-linear response of the actuator by allowing part of a command to be carried out open-loop. This can improve response times, and can also reduce overshoot by requiring less integral action. However it can also make the system less stable.
- Proportional control similarly improves response times and can reduce overshoot, but can also make the system less stable.
- To improve stability, setpoint weighting allows the amount of commanded position used in proportional control to be reduced. This makes the system slower to respond to a command, but more stable in response to system perturbations.
- Derivative control considers only the measured derivative of position (i.e. measured velocity), so that rapid changes in commanded position do not cause large glitches in derivative control.

8.10.2 Nested PIDs for velocity and acceleration control

For the nested velocity and acceleration PID loops, the following features also apply. The velocity loop is described here, but the same principles and architecture apply to the acceleration loop too.

- For a constant position error, integral control will ramp at a constant rate set by the integral time constant. A constant ramp of position is (by definition) a velocity. By summing this with the commanded velocity, the velocity PID loop carries out both closed-loop control of velocity and integral control of position, and can be enabled or disabled without affecting system stability.
- The position feedforward, proportional and derivative terms are required to act immediately, in phase with the measured position. These terms are therefore not fed to the inner velocity loop.

- The contribution of the position integral term into the velocity PID loop is normalised for the sum of the velocity control feedforward and proportional gains. Changing these gains would otherwise effectively change the position integral gain, with a risk of making the position loop unstable. By normalising for these gains, the calibrations of the position and velocity loops can remain relatively independent.
- Each inner loop operates on the derivative of the loop above – position, then velocity, then acceleration. The control output from each inner loop is therefore integrated before it is summed with the feedforward, proportional and derivative terms from the outer loop.
- Protection logic is added to ensure the system remains stable in case any of the PID loops saturates, preventing limit cycles which are a known issue for nested control loops.

8.10.3 stage.closed-loop.pid.control.get/set**8.10.3.1 Description**

Closed-loop PID control selection

8.10.3.2 Command

```
stage.closed-loop.pid.control.get <channel>
stage.closed-loop.pid.control.set <channel> <value>
```

8.10.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	String	Enumeration “closed-loop-pid-control”			Closed-loop PID control selection One of <ul style="list-style-type: none"> • position • velocity • accel

8.10.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “closed-loop-pid-control”			Closed-loop PID control selection One of <ul style="list-style-type: none"> • position • velocity • accel

8.10.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Channel not available	No stage is connected to the specified channel
Value out of range	Selection value is invalid (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.3.7 Minimum security level

stage.closed-loop.pid.control.get	User level security required
stage.closed-loop.pid.control.set	Superuser level security required

8.10.3.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.4 stage.closed-loop.pid.position.proportional.gain.get/set

8.10.4.1 Description

Closed-loop position control PID loop proportional gain

8.10.4.2 Command

```
stage.closed-loop.pid.position.proportional.gain.get <channel>
```

```
stage.closed-loop.pid.position.proportional.gain.set <channel> <value>
```

8.10.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		-1000	1000	Position PID loop proportional gain

8.10.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		-1000	1000	Position PID loop proportional gain

8.10.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.4.7 Minimum security level

stage.closed-loop.pid.position.proportional.gain.get	User level security required
stage.closed-loop.pid.position.proportional.gain.set	Superuser level security required

8.10.4.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.5 stage.closed-loop.pid.position.proportional.set-point-weighting.get/set**8.10.5.1 Description**

Closed-loop control position PID loop proportional setpoint weighting

8.10.5.2 Command

```
stage.closed-loop.pid.position.proportional.set-point-weighting.get
<channel>
```

```
stage.closed-loop.pid.position.proportional.set-point-weighting.set
<channel> <value>
```

8.10.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	1	Position PID loop proportional setpoint weighting

8.10.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	1	Position PID loop proportional setpoint weighting

8.10.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Setpoint weighting must be within specified range (set only)

Error return “errcode” string reported	Description
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.5.7 Minimum security level

stage.closed-loop.pid.position.proportional.set-point-weighting.get	User level security required
stage.closed-loop.pid.position.proportional.set-point-weighting.set	Superuser level security required

8.10.5.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.6 stage.closed-loop.pid.position.integral.time-constant.get/set

8.10.6.1 Description

Closed-loop control position PID loop integral time constant

8.10.6.2 Command

```
stage.closed-loop.pid.position.integral.time-constant.get <channel>
stage.closed-loop.pid.position.integral.time-constant.set <channel>
<value>
```

8.10.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	10	Position PID loop integral time constant

8.10.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	10	Position PID loop integral time constant

8.10.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.6.7 Minimum security level

stage.closed-loop.pid.position.integral.time-constant.get	User level security required
stage.closed-loop.pid.position.integral.time-constant.set	Superuser level security required

8.10.6.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.7 stage.closed-loop.pid.position.differential.gain.get/set

8.10.7.1 Description

Closed-loop position control PID loop differential gain

8.10.7.2 Command

```
stage.closed-loop.pid.position.differential.gain.get <channel>
```

```
stage.closed-loop.pid.position.differential.gain.set <channel> <value>
```

8.10.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	10	Position PID loop differential gain

8.10.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	10	Position PID loop differential gain

8.10.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.7.7 Minimum security level

stage.closed-loop.pid.position.proportional.differential.gain.get	User level security required
stage.closed-loop.pid.position.proportional.differential.gain.set	Superuser level security required

8.10.7.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.8 stage.closed-loop.pid.position.feedforward.gain.get/set

8.10.8.1 Description

Closed-loop control position PID loop feedforward gain

8.10.8.2 Command

stage.closed-loop.pid.position.feedforward.gain.get <channel>

stage.closed-loop.pid.position.feedforward.gain.set <channel> <value>

8.10.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		-10	10	Position PID loop feedforward gain

8.10.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		-10	10	Position PID loop feedforward gain

8.10.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.8.7 Minimum security level

stage.closed-loop.pid.position.feedforward.gain.get	User level security required
stage.closed-loop.pid.position.feedforward.gain.set	Superuser level security required

8.10.8.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.9 stage.closed-loop.pid.velocity.proportional.gain.get/set

8.10.9.1 Description

Closed-loop velocity control PID loop proportional gain

8.10.9.2 Command

```
stage.closed-loop.pid.velocity.proportional.gain.get <channel>
```

```
stage.closed-loop.pid.velocity.proportional.gain.set <channel> <value>
```

8.10.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	1000	Velocity PID loop proportional gain

8.10.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	1000	Velocity PID loop proportional gain

8.10.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.9.7 Minimum security level

stage.closed-loop.pid.velocity.proportional.gain.get	User level security required
stage.closed-loop.pid.velocity.proportional.gain.set	Superuser level security required

8.10.9.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.10 stage.closed-loop.pid.velocity.proportional.set-point-weighting.get/set**8.10.10.1 Description**

Closed-loop control velocity PID loop proportional setpoint weighting

8.10.10.2 Command

```
stage.closed-loop.pid.velocity.proportional.set-point-weighting.get  
<channel>
```

```
stage.closed-loop.pid.velocity.proportional.set-point-weighting.set  
<channel> <value>
```

8.10.10.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.10.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	1	Velocity PID loop proportional setpoint weighting

8.10.10.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	1	Velocity PID loop proportional setpoint weighting

8.10.10.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Setpoint weighting must be within specified range (set only)

Error return “errcode” string reported	Description
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.10.7 Minimum security level

stage.closed-loop.pid.velocity.proportional.set-point-weighting.get	User level security required
stage.closed-loop.pid.velocity.proportional.set-point-weighting.set	Superuser level security required

8.10.10.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.11 stage.closed-loop.pid.velocity.integral.time-constant.get/set**8.10.11.1 Description**

Closed-loop control velocity PID loop integral time constant

8.10.11.2 Command

```
stage.closed-loop.pid.velocity.integral.time-constant.get <channel>
```

```
stage.closed-loop.pid.velocity.integral.time-constant.set <channel>
<value>
```

8.10.11.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.11.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	10	Velocity PID loop integral time constant

8.10.11.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	10	Velocity PID loop integral time constant

8.10.11.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.11.7 Minimum security level

stage.closed-loop.pid.velocity.integral.time-constant.get	User level security required
stage.closed-loop.pid.velocity.integral.time-constant.set	Superuser level security required

8.10.11.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.12 stage.closed-loop.pid.velocity.differential.gain.get/set**8.10.12.1 Description**

Closed-loop velocity control PID loop differential gain

8.10.12.2 Command

```
stage.closed-loop.pid.velocity.differential.gain.get <channel>
```

```
stage.closed-loop.pid.velocity.differential.gain.set <channel> <value>
```

8.10.12.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.12.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	10	Velocity PID loop differential gain

8.10.12.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	10	Velocity PID loop differential gain

8.10.12.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.12.7 Minimum security level

stage.closed-loop.pid.velocity.proportional.differential.gain.get	User level security required
stage.closed-loop.pid.velocity.proportional.differential.gain.set	Superuser level security required

8.10.12.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.13 stage.closed-loop.pid.velocity.feedforward.gain.get/set**8.10.13.1 Description**

Closed-loop control velocity PID loop feedforward gain

8.10.13.2 Command

```
stage.closed-loop.pid.velocity.feedforward.gain.get <channel>
```

```
stage.closed-loop.pid.velocity.feedforward.gain.set <channel> <value>
```

8.10.13.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.13.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		>0	10	Velocity PID loop feedforward gain

8.10.13.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		>0	10	Velocity PID loop feedforward gain

8.10.13.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.13.7 Minimum security level

stage.closed-loop.pid.velocity.feedforward.gain.get	User level security required
stage.closed-loop.pid.velocity.feedforward.gain.set	Superuser level security required

8.10.13.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.14 stage.closed-loop.pid.accel.proportional.gain.get/set**8.10.14.1 Description**

Closed-loop acceleration control PID loop proportional gain

8.10.14.2 Command

```
stage.closed-loop.pid.accel.proportional.gain.get <channel>
```

```
stage.closed-loop.pid.accel.proportional.gain.set <channel> <value>
```

8.10.14.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.14.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	1000	Acceleration PID loop proportional gain

8.10.14.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	1000	Acceleration PID loop proportional gain

8.10.14.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.14.7 Minimum security level

stage.closed-loop.pid.accel.proportional.gain.get	User level security required
stage.closed-loop.pid.accel.proportional.gain.set	Superuser level security required

8.10.14.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.15 stage.closed-loop.pid.accel.proportional.set-point-weighting.get/set

8.10.15.1 Description

Closed-loop control acceleration PID loop proportional setpoint weighting

8.10.15.2 Command

```
stage.closed-loop.pid.accel.proportional.set-point-weighting.get
<channel>
```

```
stage.closed-loop.pid.accel.proportional.set-point-weighting.set
<channel> <value>
```

8.10.15.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.15.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	1	Acceleration PID loop proportional setpoint weighting

8.10.15.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	1	Acceleration PID loop proportional setpoint weighting

8.10.15.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Setpoint weighting must be within specified range (set only)

Error return “errcode” string reported	Description
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.15.7 Minimum security level

stage.closed-loop.pid.accel.proportional.set-point-weighting.get	User level security required
stage.closed-loop.pid.accel.proportional.set-point-weighting.set	Superuser level security required

8.10.15.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.16 stage.closed-loop.pid.accel.integral.time-constant.get/set**8.10.16.1 Description**

Closed-loop control acceleration PID loop integral time constant

8.10.16.2 Command

```
stage.closed-loop.pid.accel.integral.time-constant.get <channel>
```

```
stage.closed-loop.pid.accel.integral.time-constant.set <channel>
<value>
```

8.10.16.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.16.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	10	Acceleration PID loop integral time constant

8.10.16.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	10	Acceleration PID loop integral time constant

8.10.16.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.16.7 Minimum security level

stage.closed-loop.pid.accel.integral.time-constant.get	User level security required
stage.closed-loop.pid.accel.integral.time-constant.set	Superuser level security required

8.10.16.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.10.17 stage.closed-loop.pid.accel.feedforward.gain.get/set**8.10.17.1 Description**

Closed-loop control acceleration PID loop feedforward gain

8.10.17.2 Command

```
stage.closed-loop.pid.accel.feedforward.gain.get <channel>
```

```
stage.closed-loop.pid.accel.feedforward.gain.set <channel> <value>
```

8.10.17.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.10.17.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		>0	10	Acceleration PID loop feedforward gain

8.10.17.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		>0	10	Acceleration PID loop feedforward gain

8.10.17.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.10.17.7 Minimum security level

stage.closed-loop.pid.accel.feedforward.gain.get	User level security required
stage.closed-loop.pid.accel.feedforward.gain.set	Superuser level security required

8.10.17.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.11 Damping

Damping (see diagram 8.2.13) provides control over system resonances, to improve system stability and thereby allow operation at much higher system bandwidths. This is achieved using notch filtering and active damping.

The notch filter(s) remove frequencies from the control output which may excite resonances in the system. Where resonances are known and stable, this can be highly effective at preventing the control loop causing ringing at those frequencies. The user has the option of adding a single 2nd-order filter, two 2nd-order filters at separate frequencies, two 2nd-order filters at the same frequency for a more “rounded-off” filter, or a single 4th-order filter.

Whilst notch filters stop closed-loop control from directly exciting these resonances, they cannot prevent the rest of the system from exciting them, for example from rapid acceleration or deceleration during a fast step. If the stage is very stiff or well-damped then this may be less of a problem, but less stiff stages may be easily excited by environmental factors.

Active damping feeds back measured velocity in anti-phase to damp stage movement. Active damping on velocity adds “virtual friction”. The result is to add stiffness to the system through software control.

Notch filters and active damping are available in both closed-loop and open-loop modes. This can make the digital controller an attractive proposition even in open-loop applications where digital control would traditionally be considered overly complex or expensive.

As with closed-loop control (see section 8.10), active damping requires that the phase shifts on measured velocity calculation (see section 8.8) are acceptable at the frequencies of interest. This is especially relevant for active damping; closed-loop control may have a bandwidth below the stage’s first resonance, but active damping will typically be required to operate at frequencies well above this. Setting the filter cutoff appropriately for velocity calculations is therefore vital when using active damping.

Differentiation is inherently noisy, and in spite of filtering there will always be some noise on the velocity signal. Adding a proportion of this signal directly to the servo output can increase position noise level. A deadband on velocity is therefore provided below which no damping will be applied. This does not affect stability, because if the stage is not moving then of course no damping is required, but it removes the effect of active damping on steady-state noise.

8.11.1 stage.notch-filter.filter-type.get/set**8.11.1.1 Description**

Type of notch filter(s) in control loop

8.11.1.2 Command

```
stage.notch-filter.filter-type.get <channel>
stage.notch-filter.filter-type.set <channel> <value>
```

8.11.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	String	Enumeration "notch-filter-type"			<p>Type of notch filter(s) in control loop</p> <p>One of:-</p> <ul style="list-style-type: none"> • disabled • single-2nd-order: One 2nd-order Bessel notch filter • dual-2nd-order: Two independent 2nd-order Bessel notch filters • single-paired-2nd-order: One 4th-order filter comprising two 2nd-order Bessel notch filters in series • single-4th-order: One 4th-order Bessel notch filter

8.11.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration "notch-filter-type"			<p>Type of notch filter(s) in control loop</p> <p>One of:-</p>

Name	Type	Units	Minimum	Maximum	Description
					<ul style="list-style-type: none"> disabled single-2nd-order: One 2nd-order Bessel notch filter dual-2nd-order: Two independent 2nd-order Bessel notch filters single-paired-2nd-order: One 4th-order filter comprising two 2nd-order Bessel notch filters in series single-4th-order: One 4th-order Bessel notch filter

8.11.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Notch filter location value is invalid (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.1.7 Minimum security level

stage.notch-filter.filter-type.get	User level security required
stage.notch-filter.filter-type.set	Superuser level security required

8.11.1.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.11.2 stage.notch-filter.time-constant.get/set

8.11.2.1 Description

First (or single) notch filter time constant

8.11.2.2 Command

```
stage.notch-filter.time-constant.get <channel>  
stage.notch-filter.time-constant.set <channel> <value>
```

8.11.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Notch filter time constant

8.11.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Notch filter time constant

8.11.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.2.7 Minimum security level

stage.notch-filter.time-constant.get	User level security required
stage.notch-filter.time-constant.set	Superuser level security required

8.11.2.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.11.3 stage.notch-filter.q.get/set

8.11.3.1 Description

First (or single) notch filter Q factor

8.11.3.2 Command

```
stage.notch-filter.q.get <channel>  
stage.notch-filter.q.set <channel> <value>
```

8.11.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		Notch filter Q factor

8.11.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Notch filter Q factor

8.11.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Q factor must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.3.7 Minimum security level

stage.notch-filter.q.get	User level security required
stage.notch-filter.q.set	Superuser level security required

8.11.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.11.4 stage.notch-filter.second-filter-time-constant.get/set

8.11.4.1 Description

Second notch filter time constant

8.11.4.2 Command

```
stage.notch-filter.second-filter-time-constant.get <channel>
```

```
stage.notch-filter.second-filter-time-constant.set <channel> <value>
```

8.11.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	Notch filter time constant

8.11.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	Notch filter time constant

8.11.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.4.7 Minimum security level

stage.notch-filter.second-filter-time-constant.get	User level security required
stage.notch-filter.second-filter-time-constant.set	Superuser level security required

8.11.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.11.5 stage.notch-filter.second-filter-q.get/set

8.11.5.1 Description

Second notch filter Q factor

8.11.5.2 Command

```
stage.notch-filter.second-filter-q.get <channel>
```

```
stage.notch-filter.second-filter-q.set <channel> <value>
```

8.11.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		Notch filter Q factor

8.11.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Notch filter Q factor

8.11.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Q factor must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.5.7 Minimum security level

stage.notch-filter.second-filter-q.get	User level security required
stage.notch-filter.second-filter-q.set	Superuser level security required

8.11.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.11.6 stage.notch-filter.bode.gain.get

8.11.6.1 Description

Notch filter Bode plot gain

8.11.6.2 Command

```
stage.notch-filter.bode.gain.get <channel> <frequency>
```

8.11.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.11.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	dB			Bode plot gain

8.11.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.6.6 Minimum security level

stage.notch-filter.bode.gain.get	User level security required
----------------------------------	------------------------------

8.11.6.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.11.7 stage.notch-filter.bode.phase.get

8.11.7.1 Description

Notch filter Bode plot phase

8.11.7.2 Command

```
stage.notch-filter.bode.phase.get <channel> <frequency>
```

8.11.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
frequency	32-bit floating-point	Hz	>0	Nyquist limit	Frequency

8.11.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	degrees	-180	180	Bode plot phase

8.11.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Frequency must be within specified range
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.7.6 Minimum security level

stage.notch-filter.bode.phase.get	User level security required
-----------------------------------	------------------------------

8.11.7.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.11.8 stage.active-damping.enable.get/set

8.11.8.1 Description

Active damping enable

8.11.8.2 Command

```
stage.active-damping.enable.get <channel>
```

```
stage.active-damping.enable.set <channel> <value>
```

8.11.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Active damping enable

8.11.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Active damping enable

8.11.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Enable value is invalid (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.8.7 Minimum security level

stage.active-damping.enable.get	User level security required
stage.active-damping.enable.set	Superuser level security required

8.11.8.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.11.9 stage.active-damping.velocity.gain.get/set

8.11.9.1 Description

Active damping velocity damping gain

8.11.9.2 Command

```
stage.active-damping.velocity.gain.get <channel>
```

```
stage.active-damping.velocity.gain.set <channel> <value>
```

8.11.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	10	Velocity damping gain

8.11.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	10	Velocity damping gain

8.11.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.9.7 Minimum security level

stage.active-damping.velocity.gain.get	User level security required
stage.active-damping.velocity.gain.set	Superuser level security required

8.11.9.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.11.10 stage.active-damping.velocity.deadband.get/set**8.11.10.1 Description**

Active damping velocity deadband

8.11.10.2 Command

stage.active-damping.velocity.deadband.get <channel>

stage.active-damping.velocity.deadband.set <channel> <value>

8.11.10.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.11.10.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm/s	0		Velocity deadband

8.11.10.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm/s	0		Velocity deadband

8.11.10.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Deadband must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.11.10.7 Minimum security level

stage.active-damping.velocity.deadband.get	User level security required
stage.active-damping.velocity.deadband.set	Superuser level security required

8.11.10.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

8.12 Actuator output

The final element of the control system is to drive the piezo actuator with the control signal (see diagram 8.2.15).

Different piezo actuators will expand at different rates for the same applied voltage, giving slightly different dynamic performance. If the actuator expansion rate is measured during stage calibration, actuator compensation allows the controller to adjust the applied voltage to give consistent actuator expansion across different stages. This makes dynamic performance more similar across stages, improving stage interchangeability. This is factory-calibrated for the stage, as an optional feature on request, and is not user-configurable.

Soft-start protection ensures that there is no unexpected stage movement when a stage is first connected, and that the stage begins control operations with a smooth ramp to the initial commanded position.

Some low-noise applications are adversely affected by closed-loop control, where noise on the measured position inevitably adds noise on the servo output. For these applications, the stage can be driven to a desired location, after which the servo output (and hence the applied voltage on the piezo actuator) is frozen whilst measurements are taken by other equipment. Noise from the control loop is thereby removed during measurement. The piezo actuator will continue to creep whilst the servo output is held, but this is typically acceptable for measurements which do not take significant time. When the servo output is unfrozen, closed-loop control resumes steplessly (assuming the system is operating with closed-loop control).

Some stages require the piezo actuator output to be limited so that the stage is not driven to extreme positions which could damage the stage. For most stages, no limits are imposed and the stage will simply be driven over the maximum available range. In either case, if the stage cannot be driven further, this status is reported to the user. For some applications the user may need to be aware that further movement in this direction will not be possible.

8.12.1 stage.mode.freeze-servo-output.get/set**8.12.1.1 Description**

Freeze/unfreeze servo output

8.12.1.2 Command

```
stage.mode.freeze-servo-output.get <channel>
stage.mode.freeze-servo-output.set <channel> <value>
```

8.12.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.12.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Freeze servo output 1 = freeze 0 = unfreeze

8.12.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Freeze servo output 1 = freeze 0 = unfreeze

8.12.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.12.1.7 Minimum security level

stage.mode.freeze-servo-output.get	No security required
stage.mode.freeze-servo-output.set	User level security required

8.12.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

8.12.2 stage.status.servo-output-at-limits.get

8.12.2.1 Description

Stage servo output is driven to minimum or maximum limit

8.12.2.2 Command

```
stage.status.servo-output-at-limits.get <channel>
```

8.12.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

8.12.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage servo output is driven to minimum or maximum limit

8.12.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

8.12.2.6 Minimum security level

stage.status.servo-output-at-limits.get	No security required
---	----------------------

8.12.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

9 Closed-loop control system calibration

9.1 Overview

Tuning a basic PID loop is well-documented in engineering literature, using methods such as Ziegler-Nichols. Even without a formal method, the PID terms lend themselves well to manual tuning, since the effects of each term on the stage response are relatively intuitive.

Engineers are frequently unfamiliar with methods of tuning nested PID loops, and the control loop design contains features which are relatively novel (at least in a nanopositioning application). The greatest risk when incorrectly calibrating the control loop is not that control is less-than-optimal, but that the control loop causes the system to become unstable and oscillate. The resulting forces may damage the stage or reduce its lifespan (which is rated in terms of the number of cycles). As far as possible, the stage must never be allowed to go unstable, and the following procedures focus on how to ensure this.

Note though that whilst the system must not be allowed to go into resonance, we must excite the system sufficiently that ringing at the natural resonant frequencies is observed. These will normally be the frequencies at which any unstable resonance will occur. The calibration process can then ensure that these resonant frequencies are damped, so that the system becomes more stable. With a more stable system, the stage can subsequently be tuned to give a faster response.

9.2 Initial position PID loop calibration

The calibration procedure must initially start with a working closed-loop configuration. Stages are shipped with factory presets which may be used as the basis for other configurations. If the user wishes to start from scratch though, the following configuration provides a baseline.

1. All loads removed from stage, so that testing is carried out in no-load configuration.
2. Closed-loop control selected.
3. Position control selected.
4. Position proportional, derivative and feedforward gains set to zero.
5. Position integral time constant set to a relatively slow value (10ms is suggested).
6. Notch filter disabled.
7. Active damping disabled.
8. Position low-pass filter disabled.
9. Command trajectory control disabled.
10. Command edge boost filters disabled.
11. Velocity and acceleration filter cutoff set to Nyquist (25kHz for the NPC-D-6000 controller).

9.3 Set position, velocity and acceleration filters

The first step in the configuration process is to ensure closed-loop control is working with sufficiently clean signals.

1. Using Nanobench, set up a snapshot with 1um step, 0.2s delay before the rising edge, 0.5s before the falling edge, and total snapshot time of 1.2s (giving 0.5s capture after the falling edge). Capture measured position, measured velocity (scope point 422) and

measured acceleration (scope point 432). See section 17 for details of snapshots and section 18 for details of scope points.

2. Capture a snapshot and verify that some ringing is visible on the edges of steps. If no ringing is visible, reduce the position integral time constant progressively until some ringing can be seen on the snapshot results. Retain this snapshot for comparison against results from the following steps.
3. Enable the position low-pass filter. Adjust the filter configuration to find a setting which reduces the steady-state noise on measured position to acceptable levels, checking the noise in the 0.2s before the rising edge.
4. Examine the rising and falling edges of measured position, and the phase of ringing on measured position, velocity and acceleration. Verify that the position low-pass filter does not adversely affect these, compared to the snapshot without filtering. If it does, the filter settings must be adjusted. There will always be some degree of trade-off between reduced noise and phase response.
5. Adjust the velocity and acceleration filter time constant to find a setting which reduces noise on measured velocity and acceleration to an acceptable level whilst not adversely affecting the phase of ringing.
6. Examine Bode plots if necessary to confirm the assessment of impact on phase response.

Factory presets already have a configuration which is known to work. Using a factory preset as a baseline should remove the need for users to configure this themselves.

9.4 Configure damping and fit load

Damping of stage resonance must be handled before a good tuning can be achieved.

1. If active damping is to be used, enable active damping. Configure the deadband for velocity to the observed steady-state noise levels, then repeatedly capture snapshots and progressively increase active damping gain until resonances are damped as desired. Note that this can take the system into instability, so this must be done cautiously.
2. Note that active damping must be configured with no load. With a load present, the load will tend to “smooth out” the effects of active damping. Removing the load can then cause the system to become unstable.
3. Once active damping is configured, fit any load to the stage which is required for the desired configuration.
4. Capture a snapshot with the load fitted and measure the frequency of the ringing, which will indicate the first stage resonance. Set a single 2nd-order notch filter at this frequency, starting with a Q factor of 1. Repeat the snapshot and check whether a further resonant frequency should also be eliminated; if so, set up a second notch filter at this frequency.
5. Note that notch filtering should be configured with the intended load present, because resonant frequencies tend to vary with load. In practise, the degree to which this is required depends on the stage design. Note also that if the configuration is intended to deal with a varying load, the Q factors may need to be widened to ensure the notch filter remains effective over the varying resonant frequencies corresponding to the varying load.

Where active damping is relevant for a stage, factory presets will generally set this appropriately. Using preset settings will remove the risk of driving the system into instability whilst calibrating active damping.

Care should also be taken when changing or removing a notch filter in an existing preset. Faster PID settings rely on the notch filter to provide stability, so removing the notch filter or changing the frequency can cause system instability. Ensure the PID is configured for a slow response as described in section 9.2 before adjusting the notch filter.

9.5 Tune position PID+FF loop

Numerous methods exist for tuning a position PID loop, such as Ziegler-Nichols. This section will not attempt to add to these methods, beyond noting general features of the PID loop.

1. The primary control from the PID is the integral term. However a fast integral term tends to give a larger overshoot.
2. The feedforward term tends to reduce overshoot by adding an open-loop command in the correct direction. This can be seen most easily by setting the integral term to be very much slower. Note that this relies on the linearity of the piezo actuator. Since the piezo actuator has significant non-linearity and hysteresis, it is recommended that the feedforward term should not command more than around 75% of the move, otherwise it is likely that the feedforward term will create overshoot instead. Note also that a rapid change from feedforward control may excite resonances and hence make the system less stable.
3. The proportional term tends to reduce overshoot by correcting larger position errors before the integral term can ramp up. Because measured position will always have some noise, increasing proportional control can increase the system noise.
4. The proportional setpoint weighting allows proportional control to provide stronger correction for position deviations and hence improved stability, as opposed to tracking the command. In practise this is rarely beneficial for nanopositioning. Note that the feedforward term must not be used if the proportional setpoint weighting is used, because the two control techniques oppose each other.
5. The derivative term acts to damp ringing caused by the integral term overshoot. Because measured velocity will always have some noise, increasing derivative control can increase the system noise.

A snapshot of measured position with a step is usually used to observe the system response when tuning the position PID.

Even if velocity and acceleration control will be added subsequently, stage tuning should always begin with an adequately-tuned position PID+FF loop. If velocity and acceleration control will not be used, the user need only consider tuning command trajectory control (see section 9.10).

9.6 Prepare velocity PID+FF loop

If velocity control is to be added, care must be taken to ensure that the system remains stable when velocity control is first enabled. This is relatively straightforward, requiring only the following steps.

1. Velocity feedforward gain set to 1.
2. Velocity proportional and derivative gains set to zero.
3. Velocity integral time constant set to 1s.

With velocity feedforward gain set to 1, the position integral term acts as normal. With the velocity integral term set to the longest time constant possible, the velocity loop is slow enough to have virtually no impact. If position control is stable, then velocity control with these settings will always be stable.

Note that because the velocity proportional and feedforward terms handle position integral control, the velocity feedforward gain must always be non-zero and positive.

9.7 Tune velocity PID+FF loop

The velocity PID+FF loop may be tuned using the snapshot response to a velocity step, in the same way as tuning the position PID+FF loop. A step-change in velocity indicates a constant-velocity ramp starting and ending. A ramp may be set up by using the function playback waveform generator to create constant-velocity ramp segments (see section 14). Alternatively a ramp may also be set up by temporarily enabling command trajectory control (see section 8.7) with a velocity limit so that a step-change produces a velocity-limited ramp, and setting extremely large values for acceleration and deceleration limits which essentially disable those limits.

When capturing a snapshot for the velocity step, it is recommended to capture measured position, commanded velocity (scope point 421), commanded velocity plus position control commanded velocity (scope point 423), and measured velocity (scope point 422).

The position PID+FF tuning features described in section 9.5 apply to the velocity PID+FF loop as well. The following points must also be considered with velocity control, relating to the nested PID+FF loop architecture.

1. The strength of the position control loop affects the velocity loop. If accurate velocity control is essential, it may be necessary to detune the position control loop so that the velocity loop is dominant. In particular, since the position integral term is usually the primary term for position control, compare scope points 421 and 422 to examine the effect of the position integral term on the target for the velocity control loop.
2. The velocity feedforward term ensures that the position control loop has closed-loop control over position (via the position integral term). If the velocity loop is to be dominant, it may be desirable to reduce the velocity feedforward term. Ensure that the velocity loop is well-tuned before this, otherwise the system will lose position stability. There must always be some velocity feedforward or proportional gain set, in order to retain position stability.
3. With velocity control enabled and well-tuned, the position PID derivative term is largely redundant. The position derivative term attempts to damp position movements using velocity, but velocity control inherently provides better control of this.

9.8 Prepare acceleration PI+FF loop

As described in section 9.6 for velocity control, care must be taken to ensure the system remains stable when acceleration control is first enabled. This requires only the following steps.

1. Acceleration feedforward gain set to 1.
2. Acceleration proportional gain set to zero.
3. Acceleration integral time constant set to 1s.

9.9 Tune acceleration PI+FF loop

As described in section 9.7 for velocity control, the acceleration loop may also be tuned using snapshots with step-changes in acceleration. Step-changes in acceleration may be set up by using the function playback waveform generator to create constant-acceleration segments or a constant acceleration/deceleration position step segment (see section 14). Alternatively this can also be set up by temporarily enabling command trajectory control (see section 8.7) with acceleration and deceleration limits so that a step-change produces an acceleration-limited ramp.

When capturing a snapshot for the velocity step, it is recommended to capture measured position, commanded acceleration (scope point 431), commanded acceleration plus velocity control commanded acceleration (scope point 433), and measured acceleration (scope point 432). Alternatively, if accurate velocity control is essential and the velocity control loop is to be dominant, consider capturing measured velocity (scope point 422) instead of measured position.

The same features of the nested control loop apply for acceleration control as described for velocity control in section 9.7.

9.10 Tune command trajectory control

Command trajectory control (see section 8.7) should typically be enabled after closed-loop control has been tuned for the desired performance in isolation. If it is enabled whilst tuning the control loop(s), it will tend to obscure the effects of calibration changes, and may cause incorrect conclusions to be drawn from observations.

1. Carry out a snapshot step as usual, capturing measured position, measured velocity (scope point 422) and measured acceleration (scope point 432).
2. Enable command trajectory control, initially setting acceleration and deceleration limits very large so that these limits are essentially disabled.
3. Set the command trajectory speed limit to the fastest velocity achieved by the stage. The system cannot move faster than this speed, so any command greater than this will only result in overshoot from PID integrators without necessarily improving the step response.
4. In systems where some overshoot is acceptable and does improve the step response, the speed limit may then be increased from this point. This adds back a well-controlled amount of overshoot to allow the step response to be better fitted to the system requirements.
5. Set the acceleration limit to approximately match the acceleration of the stage. Similarly to the speed limit, the system cannot accelerate faster than this, so any command greater than this will only result in overshoot. Also similarly to the speed limit, the acceleration limit may be increased to allow some well-controlled overshoot for performance which better fits the system requirements.
6. Set the deceleration limit to improve settling time by reducing overshoot and ringing. This does not easily fit a tuning “rule”, because setting too slow a deceleration will make the system behave as if it is over-damped and will therefore make the settling time worse. This will usually require some trial and error to configure.

9.11 Iterate control loop tunings

All control loop parameters inevitably interact to some extent. When a reasonable first tuning has been established using the rules above, it is normal to examine how well the tuning meets application requirements and adjust settings as necessary.

As usual for control loop tuning, this is an iterative process, but with nested control loops this will naturally require more iterations than for a simple position PID loop. Since the basic control architecture still uses the standard PID loop though, it is relatively straightforward for users to become familiar with the system and calibrate it effectively.

10 Resonance detection and frequency spectrum analysis

10.1 Overview

Resonance may be simply defined as “continuous uncommanded stage movement at some frequency”. The lifespan of stage flexures is dictated by the number of cycles. Rapid continuous movement for long periods of time will increase the wear on flexures and therefore reduce the lifespan of the stage, which is undesirable.

All systems will have some natural resonant frequencies – a metal rod will have a frequency it “rings” at, for example. These resonances will be excited when the stage moves, which may be seen when examining the system step response or other events, but normally this will naturally decay away. The action of the control loop may continue to excite these frequencies though, causing them to ring continuously. Typically the notch filter (see 8.11) is used to avoid exciting these frequencies, but changing stage loads will often change the resonant frequency, and if the correct settings are not selected (typically by selecting an inappropriate preset) then the notch filter may be ineffective.

Gain/phase effects in the system may also cause resonance due to the combined actions of the sensor, actuator, control loop and physical system, in the same way as a microphone near a loudspeaker can develop “feedback”. Again, changing stage loads will change the behaviour of the physical system, and selecting inappropriate control loop settings (typically by selecting an inappropriate preset for the load) may cause this to occur.

To detect this, the controller continuously captures commanded position and measured position for each stage, and analyses the frequency response. As a closed-loop system, the measured position should track the commanded position, so resonances can be detected as peaks in the measured position frequency spectrum which are not present (or are greater than) peaks for the corresponding frequency in the commanded position frequency spectrum. This technique is common in post-processing analysis, but here it is carried out in real time on the 6000-series controller. If a significant error persists for some time, it can be assumed to be a continuous resonance and not simply “ringing” after some transient event.

Because this feature is enabled by real-time frequency analysis, it also allows frequency spectra to be captured by the user in the same way as other data such as snapshots (see 17). Real-time analysis gives slightly less accurate results than can be achieved by capturing a snapshot and post-processing (with dedicated software such as Nanobench, or in numerical software packages such as Labview or Excel), because snapshots are always captured at the controller sample rate and can acquire up to 10s of data. However it is convenient to have this available independently of other features and without requiring additional processing.

Note that resonance detection can only take place when operating in closed-loop control. If the stage is operated in open-loop control (see section 8 for details of the control system) then resonance detection is disabled for the stage, because we cannot know what the intended stage position is. Resonance detection is also not relevant for sensors, because the controller does not cause movement.

10.2 Resonance detection protective shutdown

If resonance is detected, the controller channel is shut down to protect the stage. This is shown with the front-panel LEDs for the channel (IN POS and CLOSED) flashing green, and is reported with the command `resonance-detect.protective-shutdown.state.get` (see 10.2.1). Other channels are unaffected and continue operating as normal.

The protective shutdown may be cleared with the command `resonance-detect.protective-shutdown.clear` (see 10.2.2). The stage will then start up as if it had just been powered on, and any previous changes to control loop parameters will be discarded. This ensures the stage starts up in a safe state, in case the resonance was due to inappropriate selection of a preset or control loop settings.

10.2.1 resonance-detect.protective-shutdown.state.get

10.2.1.1 Description

State of protective shutdown for resonance

10.2.1.2 Command

```
resonance-detect.protective-shutdown.state.get <channel>
```

10.2.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.2.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	True if protective shutdown is active for channel

10.2.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.2.1.6 Minimum security level

resonance-detect.protective-shutdown.state.get	No security required
--	----------------------

10.2.1.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.2.2 resonance-detect.protective-shutdown.clear

10.2.2.1 Description

Clear protective shutdown after resonance is detected

10.2.2.2 Command

resonance-detect.protective-shutdown.clear <channel>

10.2.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.2.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Always returns 1 on success

10.2.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.2.2.6 Minimum security level

resonance-detect.protective-shutdown.clear	User level security required
--	------------------------------

10.2.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.3 Frequency spectrum analysis

Resonance detection uses 1024-point FFTs (fast Fourier transforms) to carry out the necessary frequency analysis. This is configured in production for each stage to give a suitable frequency range and resolution for the stage dynamics. Fast stages with resonances in kHz may use the maximum frequency range available, with correspondingly less accurate frequency resolution; other slower stages with lower-frequency resonances will typically downsample to use a lower frequency range but with better resolution.

This frequency range/resolution is used when frequency spectrum data is captured for protective shutdowns (see 10.4) and when captured in normal operation (see 10.6). Amplitudes for measured position, commanded position and error are reported with this frequency resolution, dividing the reported frequency range into 1024 “bins”. If the frequency range is 25kHz, say, bin 0 captures the amplitude for frequencies 0Hz to 24.4Hz, and bin 1023 captures the amplitude for frequencies 24.975kHz to 25kHz.

10.3.1 resonance-detect.frequency-range.maximum.get

10.3.1.1 Description

FFT maximum measurement frequency

10.3.1.2 Command

```
resonance-detect.frequency-range.maximum.get <channel>
```

10.3.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.3.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	Hz	0		FFT maximum measurement frequency

10.3.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.3.1.6 Minimum security level

resonance-detect.frequency-range.maximum.get	User level security required
--	------------------------------

10.3.1.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.3.2 resonance-detect.frequency-range.resolution.get

10.3.2.1 Description

FFT measurement frequency resolution

10.3.2.2 Command

```
resonance-detect.frequency-range.resolution.get <channel>
```

10.3.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.3.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	Hz	0		FFT measurement frequency resolution

10.3.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.3.2.6 Minimum security level

resonance-detect.frequency-range.resolution.get	User level security required
---	------------------------------

10.3.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.4 Protective shutdown frequency spectrum capture

Whilst the priority of the protective shutdown is to prevent damage to the stage, it may also be helpful for the user to be aware of the resonance which was detected. Knowing the resonant frequency may give useful information for diagnosis and fault-finding.

The controller reports the peak error frequency and amplitude, as well as providing a full 1024-point FFT capture for measured position, commanded position, and error amplitudes. The frequency resolution for the 1024 “bins” is common for the whole resonance detection system (see 10.3).

The protective shutdown frequency spectrum capture can be read back until the user clears the protective shutdown for the channel (see 10.2), at which point the frequency spectrum capture is cleared too.

10.4.1 resonance-detect.protective-shutdown.capture.status.get**10.4.1.1 Description**

FFT protective shutdown capture status

10.4.1.2 Command

```
resonance-detect.protective-shutdown.capture.status.get <channel>
```

10.4.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.4.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	True if protective shutdown FFT is captured for channel

10.4.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.4.1.6 Minimum security level

resonance-detect.protective-shutdown.capture.status.get	User level security required
---	------------------------------

10.4.1.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.4.2 resonance-detect.protective-shutdown.capture.peak-error.frequency.get

10.4.2.1 Description

FFT protective shutdown capture frequency at peak error magnitude

10.4.2.2 Command

```
resonance-detect.protective-shutdown.capture.peak-error.frequency.get  
<channel>
```

10.4.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.4.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	Hz	0		FFT protective shutdown capture frequency at peak error magnitude

10.4.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No resonance detection protective shutdown has taken place, so there is no FFT captured

10.4.2.6 Minimum security level

resonance-detect.protective-shutdown.capture.peak-error.frequency.get	User level security required
---	------------------------------

10.4.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.4.3 resonance-detect.protective-shutdown.capture.peak-error.magnitude.get

10.4.3.1 Description

FFT protective shutdown capture peak error magnitude

10.4.3.2 Command

```
resonance-detect.protective-shutdown.capture.peak-error.magnitude.get  
<channel>
```

10.4.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.4.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT protective shutdown capture peak error magnitude

10.4.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No resonance detection protective shutdown has taken place, so there is no FFT captured

10.4.3.6 Minimum security level

resonance-detect.protective-shutdown.capture.peak-error.magnitude.get	User level security required
---	------------------------------

10.4.3.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.4.4 resonance-detect.protective-shutdown.capture.measured-position.magnitude.get

10.4.4.1 Description

FFT protective shutdown capture measured position magnitude

10.4.4.2 Command

```
resonance-detect.protective-shutdown.capture.measured-
position.magnitude.get <channel> <index>
```

10.4.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer		0	1023	Index of FFT measurement bin 0 = 0Hz 1023 = Maximum frequency (see resonance-detect.frequency-range.maximum.get , 10.3.1)

10.4.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT protective shutdown capture measured position magnitude

10.4.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No resonance detection protective shutdown has taken place, so there is no FFT captured

Error return “errcode” string reported	Description
Index out of range	FFT bin index must be within specified range

10.4.4.6 Minimum security level

resonance-detect.protective-shutdown.capture.measured-position.magnitude.get	User level security required
--	------------------------------

10.4.4.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.4.5 resonance-detect.protective-shutdown.capture.commanded-position.magnitude.get

10.4.5.1 Description

FFT protective shutdown capture commanded position magnitude

10.4.5.2 Command

```
resonance-detect.protective-shutdown.capture.commanded-
position.magnitude.get <channel> <index>
```

10.4.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer		0	1023	Index of FFT measurement bin 0 = 0Hz 1023 = Maximum frequency (see resonance-detect.frequency-range.maximum.get , 10.3.1)

10.4.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT protective shutdown capture commanded position magnitude

10.4.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No resonance detection protective shutdown has taken place, so there is no FFT captured

Error return “errcode” string reported	Description
Index out of range	FFT bin index must be within specified range

10.4.5.6 Minimum security level

resonance-detect.protective-shutdown.capture.commanded-position.magnitude.get	User level security required
---	------------------------------

10.4.5.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.4.6 resonance-detect.protective-shutdown.capture.error.magnitude.get**10.4.6.1 Description**

FFT protective shutdown capture error magnitude

10.4.6.2 Command

```
resonance-detect.protective-shutdown.capture.error.magnitude.get
<channel> <index>
```

10.4.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer		0	1023	Index of FFT measurement bin 0 = 0Hz 1023 = Maximum frequency (see resonance-detect.frequency-range.maximum.get , 10.3.1)

10.4.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT protective shutdown capture error magnitude

10.4.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No resonance detection protective shutdown has taken place, so there is no FFT captured
Index out of range	FFT bin index must be within specified range

10.4.6.6 Minimum security level

resonance-detect.protective-shutdown.capture.error.magnitude.get	User level security required
--	------------------------------

10.4.6.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.5 Protective shutdown configuration

Resonance detection is an important safety feature to prevent damage to the stage, and does not produce false triggers in normal operation. There is normally no need for this to be disabled or reconfigured.

For superuser customers who are developing their own calibrations for a stage, it may be acceptable for the stage to occasionally resonate whilst tuning the control loop. It would be inconvenient for the calibration process if the stage was repeatedly shut down and had to be restarted. Superuser customers may therefore temporarily set parameters to make detection less sensitive, or may disable resonance detection for a channel entirely.

Resonance detection operates by checking for the amplitude difference between measured and commanded positions being greater than some threshold for some time. Typically the threshold would not be changed, but a superuser may wish to make the detection time longer. During the calibration process, a longer detection time (tens of seconds, or perhaps even longer) would allow the calibrator to change control loop parameters to stop the resonance, but would still protect the stage if the system was inadvertently left to resonate for a long period.

Resonance detection also allows maximum and minimum limits on the frequency range where resonance is detected. The minimum limit is typically set to a frequency which rejects frequencies close to DC (which cannot be measured accurately). The maximum limit is not typically used. Where some resonant feature of the system should be ignored, these limits may be changed as required.

Note that changes to resonance detection settings cannot be saved by the customer. To prevent damage to the stage, any changes to settings will be lost on power-off, and the stage will start next time with resonance detection active and operating as usual.

10.5.1 resonance-detect.protective-shutdown.enable.get/set

10.5.1.1 Description

Enable protective shutdown when resonance is detected

10.5.1.2 Command

```
resonance-detect.protective-shutdown.enable.get <channel>
```

```
resonance-detect.protective-shutdown.enable.set <channel> <value>
```

10.5.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.5.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Enable protective shutdown when resonance is detected

10.5.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Enable protective shutdown when resonance is detected

10.5.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Value out of range	Enable value is invalid (set only)

10.5.1.7 Minimum security level

resonance-detect.protective-shutdown.enable.get	User level security required
resonance-detect.protective-shutdown.enable.set	Superuser level security required

10.5.1.8 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.5.2 resonance-detect.protective-shutdown.error-threshold.get/set**10.5.2.1 Description**

Error threshold to detect resonance for protective shutdown

10.5.2.2 Command

```
resonance-detect.protective-shutdown.error-threshold.get <channel>
resonance-detect.protective-shutdown.error-threshold.set <channel>
<value>
```

10.5.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.5.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm	>0		Error threshold to detect resonance for protective shutdown

10.5.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	>0		Error threshold to detect resonance for protective shutdown

10.5.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Value out of range	Error threshold value is invalid (set only)

10.5.2.7 Minimum security level

resonance-detect.protective-shutdown.error-threshold.get	User level security required
resonance-detect.protective-shutdown.error-threshold.set	Superuser level security required

10.5.2.8 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.5.3 resonance-detect.protective-shutdown.error-time.get/set**10.5.3.1 Description**

Error time to detect resonance for protective shutdown

10.5.3.2 Command

```
resonance-detect.protective-shutdown.error-time.get <channel>
```

```
resonance-detect.protective-shutdown.error-time.set <channel> <value>
```

10.5.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.5.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	>0		Error time to detect resonance for protective shutdown

10.5.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	>0		Error time to detect resonance for protective shutdown

10.5.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Value out of range	Error time value is invalid (set only)

10.5.3.7 Minimum security level

resonance-detect.protective-shutdown.error-time.get	User level security required
resonance-detect.protective-shutdown.error-time.set	Superuser level security required

10.5.3.8 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.5.4 resonance-detect.protective-shutdown.frequency-range.minimum.get/set**10.5.4.1 Description**

Minimum frequency to detect resonance for protective shutdown

10.5.4.2 Command

```
resonance-detect.protective-shutdown.frequency-range.minimum.get
<channel>
```

```
resonance-detect.protective-shutdown.frequency-range.minimum.set
<channel> <value>
```

10.5.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.5.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	Hz	0	25000	Minimum frequency to detect resonance for protective shutdown

10.5.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	Hz	0	25000	Minimum frequency to detect resonance for protective shutdown

10.5.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Value out of range	Minimum frequency value is invalid (set only)

10.5.4.7 Minimum security level

resonance-detect.protective-shutdown.frequency-range.minimum.get	User level security required
resonance-detect.protective-shutdown.frequency-range.minimum.set	Superuser level security required

10.5.4.8 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.5.5 resonance-detect.protective-shutdown.frequency-range.maximum.get/set**10.5.5.1 Description**

Maximum frequency to detect resonance for protective shutdown

10.5.5.2 Command

```
resonance-detect.protective-shutdown.frequency-range.maximum.get
<channel>
```

```
resonance-detect.protective-shutdown.frequency-range.maximum.set
<channel> <value>
```

10.5.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.5.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	Hz	0	25000	Maximum frequency to detect resonance for protective shutdown

10.5.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	Hz	0	25000	Maximum frequency to detect resonance for protective shutdown

10.5.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Value out of range	Maximum frequency value is invalid (set only)

10.5.5.7 Minimum security level

resonance-detect.protective-shutdown.frequency-range.maximum.get	User level security required
resonance-detect.protective-shutdown.frequency-range.maximum.set	Superuser level security required

10.5.5.8 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6 Frequency spectrum snapshot capture

Since the resonance detection system continuously carries out FFT analysis of measured position and commanded position, the user may request to capture a frequency spectrum snapshot at any time. When captured, the controller reports the peak error frequency and amplitude, as well as providing a full 1024-point FFT capture for measured position, commanded position, and error amplitudes. The frequency resolution for the 1024 “bins” is common for the whole resonance detection system (see 10.3).

Like snapshots (see 17), the captured FFT is stored until another capture is triggered. Unlike snapshots, FFTs are captured independently for each channel.

10.6.1 resonance-detect.capture.trigger

10.6.1.1 Description

Clear protective shutdown after resonance is detected

10.6.1.2 Command

resonance-detect.protective-shutdown.clear <channel>

10.6.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.6.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Always returns 1 on success

10.6.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.6.1.6 Minimum security level

resonance-detect.protective-shutdown.clear	User level security required
--	------------------------------

10.6.1.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6.2 resonance-detect.capture.status.get

10.6.2.1 Description

FFT capture status

10.6.2.2 Command

```
resonance-detect.capture.status.get <channel>
```

10.6.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.6.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	True if FFT is captured for channel

10.6.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

10.6.2.6 Minimum security level

resonance-detect.capture.status.get	User level security required
-------------------------------------	------------------------------

10.6.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6.3 resonance-detect.capture.peak-error.frequency.get

10.6.3.1 Description

FFT capture frequency at peak error magnitude

10.6.3.2 Command

```
resonance-detect.capture.peak-error.frequency.get <channel>
```

10.6.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.6.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	Hz	0		FFT capture frequency at peak error magnitude

10.6.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No FFT is currently captured

10.6.3.6 Minimum security level

resonance-detect.capture.peak-error.frequency.get	User level security required
---	------------------------------

10.6.3.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6.4 resonance-detect.capture.peak-error.magnitude.get

10.6.4.1 Description

FFT capture peak error magnitude

10.6.4.2 Command

```
resonance-detect.capture.peak-error.magnitude.get <channel>
```

10.6.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

10.6.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT capture peak error magnitude

10.6.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No FFT is currently captured

10.6.4.6 Minimum security level

resonance-detect.capture.peak-error.magnitude.get	User level security required
---	------------------------------

10.6.4.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6.5 resonance-detect.capture.measured-position.magnitude.get**10.6.5.1 Description**

FFT capture measured position magnitude

10.6.5.2 Command

```
resonance-detect.capture.measured-position.magnitude.get <channel>
<index>
```

10.6.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer		0	1023	Index of FFT measurement bin 0 = 0Hz 1023 = Maximum frequency (see resonance-detect.frequency-range.maximum.get , 10.3.1)

10.6.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT capture measured position magnitude

10.6.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No FFT is currently captured
Index out of range	FFT bin index must be within specified range

10.6.5.6 Minimum security level

resonance-detect.capture.measured-position.magnitude.get	User level security required
--	------------------------------

10.6.5.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6.6 resonance-detect.capture.commanded-position.magnitude.get**10.6.6.1 Description**

FFT capture commanded position magnitude

10.6.6.2 Command

```
resonance-detect.capture.commanded-position.magnitude.get <channel>
<index>
```

10.6.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer		0	1023	Index of FFT measurement bin 0 = 0Hz 1023 = Maximum frequency (see resonance-detect.frequency-range.maximum.get , 10.3.1)

10.6.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT capture commanded position magnitude

10.6.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No FFT is currently captured
Index out of range	FFT bin index must be within specified range

10.6.6.6 Minimum security level

resonance-detect.capture.commanded-position.magnitude.get	User level security required
---	------------------------------

10.6.6.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

10.6.7 resonance-detect.capture.error.magnitude.get**10.6.7.1 Description**

FFT capture error magnitude

10.6.7.2 Command

```
resonance-detect.capture.error.magnitude.get <channel> <index>
```

10.6.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer		0	1023	Index of FFT measurement bin 0 = 0Hz 1023 = Maximum frequency (see resonance-detect.frequency-range.maximum.get , 10.3.1)

10.6.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		FFT capture error magnitude

10.6.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).
Command could not be carried out	No FFT is currently captured
Index out of range	FFT bin index must be within specified range

10.6.7.6 Minimum security level

resonance-detect.capture.error.magnitude.get	User level security required
--	------------------------------

10.6.7.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.25 onwards

11 Position rebalancing for closed-loop range

11.1 Rebalancing design

Stages are built around a piezoelectric actuator, which has a natural stiffness in a similar way to a spring.

Under load, the actuator will naturally compress. The total end-to-end stage range does not significantly change, but the position of the ends does. For example, a stage with a nominal closed-loop range of 0 to 400 μm might have an open-loop range of -50 μm to 450 μm to allow headroom for closed-loop control overshooting and piezo hysteresis effects. When fully loaded, the same stage's open-loop range might become -150 μm to 350 μm , with a reliable closed-loop range of -100 μm to 300 μm . If the user (or their software application) is only aware that the nominal closed-loop range is 0 to 400 μm , it will appear to them that they have lost 25% of this range.

Changes in temperature have similar effects, which is significant for microscopy applications. Some microscopy stages are required to move incubators and will naturally be in relatively high temperature and humidity. Even with benchtop microscopy, stages may not be operated in air-conditioned laboratories.

For microscopy and other systems where it is important to deliver the required end-to-end stage range, the controller can automatically sweep the stage over its range to determine exactly where the range lies. It can then automatically self-calibrate and rebalance the stage position to give the nominal closed-loop range with equal headroom at each end. This is run at startup, and can also be commanded by the user when changes such as adding or removing loads may have caused the stage range to have shifted.

Because this sweeps the stage over its full range, users must ensure that lenses, slides and other equipment cannot be contacted when starting the controller or when commanding rebalancing at other times. Damage to connected equipment may occur if the stage cannot safely be run over its range when rebalancing takes place.

In order to carry out this self-calibration, the stage is calibrated (at the factory) with the required closed-loop range to be provided. These values may be used by customer applications as well, where the stage range is required and hard-coding for a specific stage is not desirable. For information, the user may read back the open-loop range and the required rebalancing offset found during the range sweep.

Where the system will be operating at a constant temperature and load, and/or where it is required that the stage provides a guaranteed "zero position" for the system, automatic rebalancing can of course be disabled.

11.2 Commanding rebalancing

11.2.1 stage.range.auto-balance.select-on-startup.get/set

11.2.1.1 Description

Stage range auto-balancing selected on startup

11.2.1.2 Command

```
stage.range.auto-balance.select-on-startup.get <channel>
stage.range.auto-balance.select-on-startup.set <channel> <value>
```

11.2.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.2.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	String	Enumeration “auto-balance-on-startup”			Rebalancing selected on startup “disabled” = no rebalancing is carried out on startup “range-sweep” = range sweep carried out on startup to rebalance stage

11.2.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “auto-balance-on-startup”			Rebalancing selected on startup “disabled” = no rebalancing is carried out on startup “range-sweep” = range sweep carried out on startup to rebalance stage

11.2.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel, or previous rebalancing is still in progress
Value out of range	Data value must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section).

11.2.1.7 Minimum security level

stage.range.auto-balance.select-on-startup.get	User level security required
stage.range.auto-balance.select-on-startup.set	Superuser level security required

11.2.1.8 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.2.2 stage.range.auto-balance.trigger

11.2.2.1 Description

Stage range auto-balancing manual trigger

11.2.2.2 Command

```
stage.range.auto-balance.trigger <channel> <value>
```

11.2.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	String	Enumeration “auto-balance-trigger”			Rebalancing method required “default” = carry out method as specified for the command “stage.range.auto-balance.select-on-startup.get/set”, see 11.2.1 “range-sweep” = carry out range sweep rebalancing

Note that if “default” is selected and “stage.range.auto-balance.select-on-startup.get/set” is set to “disabled”, this command will report success but no rebalancing will take place.

11.2.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Always returns “1” to indicate this was triggered successfully

11.2.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel, or previous rebalancing is still in progress
Value out of range	Data value must be within specified range

Error return “errcode” string reported	Description
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.2.2.6 Minimum security level

stage.range.auto-balance.trigger	User level security required
----------------------------------	------------------------------

11.2.2.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.2.3 stage.range.auto-balance.in-progress.get

11.2.3.1 Description

Stage range auto-balancing in progress

11.2.3.2 Command

```
stage.range.auto-balance.in-progress.get <channel>
```

11.2.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.2.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage range auto-balancing in progress

11.2.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.2.3.6 Minimum security level

stage.range.auto-balance.in-progress.get	User level security required
--	------------------------------

11.2.3.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.3 Stage range

11.3.1 stage.range.closed-loop.minimum.get

11.3.1.1 Description

Stage nominal closed-loop range minimum value

11.3.1.2 Command

```
stage.range.closed-loop.minimum.get <channel>
```

11.3.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.3.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage nominal closed-loop range minimum value

11.3.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.3.1.6 Minimum security level

stage.range.closed-loop.minimum.get	No security required
-------------------------------------	----------------------

11.3.1.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.3.2 stage.range.closed-loop.maximum.get

11.3.2.1 Description

Stage nominal closed-loop range maximum value

11.3.2.2 Command

```
stage.range.closed-loop.maximum.get <channel>
```

11.3.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.3.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage nominal closed-loop range maximum value

11.3.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.3.2.6 Minimum security level

stage.range.closed-loop.maximum.get	No security required
-------------------------------------	----------------------

11.3.2.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.3.3 stage.range.closed-loop.maximum.get

11.3.3.1 Description

Stage nominal closed-loop range maximum value

11.3.3.2 Command

```
stage.range.closed-loop.maximum.get <channel>
```

11.3.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.3.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage nominal closed-loop range maximum value

11.3.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.3.3.6 Minimum security level

stage.range.closed-loop.maximum.get	No security required
-------------------------------------	----------------------

11.3.3.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.3.4 stage.range.closed-loop.range.get

11.3.4.1 Description

Stage nominal closed-loop total range

11.3.4.2 Command

```
stage.range.closed-loop.range.get <channel>
```

11.3.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.3.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage nominal closed-loop total range

11.3.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.3.4.6 Minimum security level

stage.range.closed-loop.range.get	No security required
-----------------------------------	----------------------

11.3.4.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.3.5 stage.range.open-loop.range.get

11.3.5.1 Description

Stage measured open-loop range

11.3.5.2 Command

```
stage.range.open-loop.range.get <channel>
```

11.3.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.3.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage measured open-loop range

11.3.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.3.5.6 Minimum security level

stage.range.open-loop.range.get	Superuser level security required
---------------------------------	-----------------------------------

11.3.5.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

11.3.6 stage.range.auto-balance.offset.get

11.3.6.1 Description

Stage range auto-balancing offset

11.3.6.2 Command

```
stage.range.auto-balance.offset.get <channel>
```

11.3.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

11.3.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage range auto-balancing offset

11.3.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

11.3.6.6 Minimum security level

stage.range.auto-balance.offset.get	Superuser level security required
-------------------------------------	-----------------------------------

11.3.6.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

12 Analogue inputs, outputs and position command

12.1 Analogue inputs

12.1.1 Overview

Many OEM systems provide an analogue output which commands stage movement. Because the majority of nanopositioning systems have analogue inputs, this allows OEMs to use different nanopositioning systems as required for cost/performance/availability, without needing to change the rest of the OEM system.

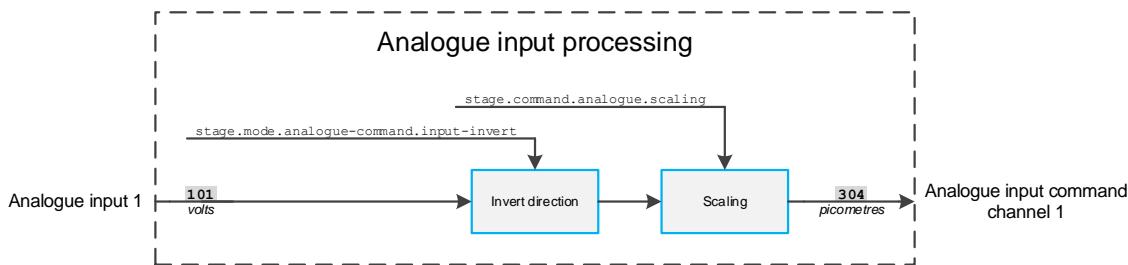
In common with other nanopositioning systems therefore, the NanoScan NPC-D-6xxx series has analogue inputs to command stage movement for each channel. The control loop diagram in 8.2.2 shows how this is summed with other command sources.

The scaling from analogue input voltage to commanded position will vary depending on the stage range. Systems for general microscopy applications typically use analogue signals where 0V commands the minimum position and +10V commands the maximum position. Other systems for applications such as beamlines may use analogue signals where -10V commands the minimum position and +10V commands the maximum position. The scaling used for a stage is specified on the stage datasheet, or for OEM customers this may be specified as part of the system requirements. The scaling is stored on the stage as part of the stage's factory calibration (see 5.2) and cannot be changed by users.

For general microscopy systems, stages may be mounted below the sample, so that increased position on the stage moves further away from the sample. The user has the ability to invert the scaling direction so that +10V commands the minimum position and 0V commands the maximum position. This can be changed by users and is stored in the stage presets.

12.1.2 Controllers with dedicated analogue inputs

Controllers such as the NPC-D-6110 and the NPC-D-6330 have one dedicated analogue input for each stage channel. On these controllers, each analogue input commands the stage position for the corresponding stage channel, if analogue commands are enabled for that channel.

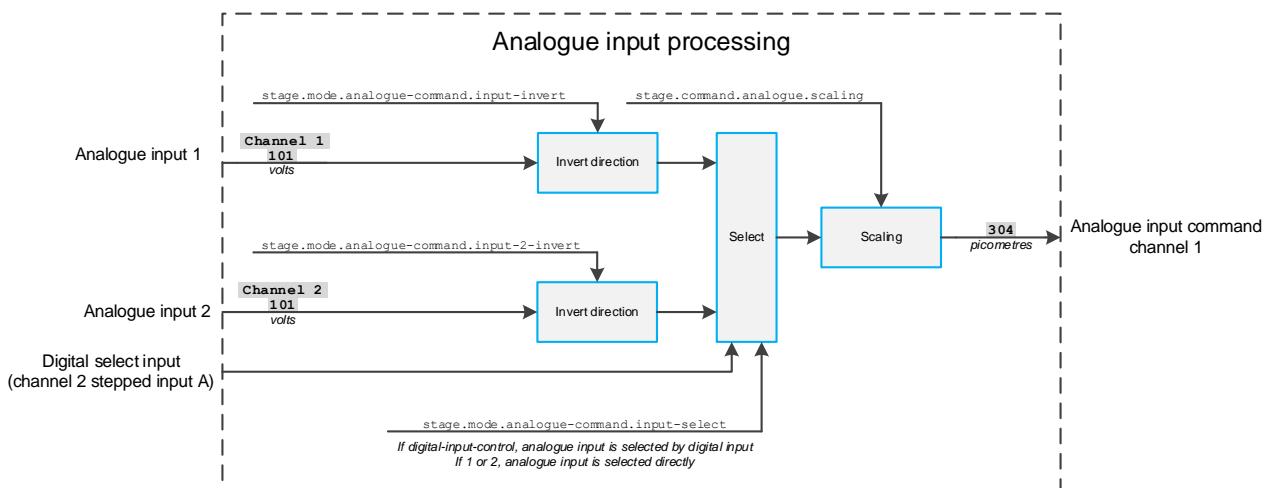


As the diagram shows, the voltage on analogue input 1 is scaled appropriately and used to command channel 1 position. Similarly on controllers with more than one channel, the voltage on analogue inputs 2 and 3 are scaled appropriately and used to command channels 2 and 3 positions. Each stage on a controller may have different ranges, so each stage has its own scaling from volts to position.

12.1.3 Controllers with switchable analogue inputs

Controllers such as the NPC-D-6111 have one stage channel but two analogue inputs. On these controllers, the analogue input to be used as the source of the analogue command may be selected from the host PC or from a digital input on the controller. On the NPC-D-6111, the digital input used to select each input uses the pin for channel 2 stepped input A (which otherwise would be unused on a single-channel controller). This allows the user to connect two possible command sources and switch between them as required without needing external equipment.

Both analogue inputs use the same scaling from voltage to position. However it is possible to invert the direction of travel for one input but not the other. For a general microscopy system which may be required to support both orientations of stage, one input may be configured as inverted and one as non-inverted. The user can then connect their equipment to the appropriate input to get the correct response with no reconfiguration required.



12.1.4 controller.analogue-inputs.get

12.1.4.1 Description

Number of analogue inputs on controller

12.1.4.2 Command

controller.analogue-inputs.get

12.1.4.3 Parameters

None

12.1.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		1		Number of analogue inputs

For controllers with dedicated analogue inputs, this will always be the same as the number of channels (see 5.1.11, command controller.channels.get). For controllers with switchable analogue inputs, this will always be 2.

12.1.4.5 Possible error reports

None

12.1.4.6 Minimum security level

controller.analogue-inputs.get	No security required
--------------------------------	----------------------

12.1.4.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

12.1.5 stage.mode.analogue-command.input-invert.get/set

12.1.5.1 Description

Invert analogue input used for channel position command

12.1.5.2 Command

stage.mode.analogue-command.input-invert.get <channel>

stage.mode.analogue-command.input-invert.set <channel> <value>

12.1.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

12.1.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Invert analogue input

12.1.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Invert analogue input

12.1.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

12.1.5.7 Minimum security level

stage.mode.analogue-command.input-invert.get	User level security required
stage.mode.analogue-command.input-invert.set	Superuser level security required

12.1.5.8 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

12.1.6 stage.mode.analogue-command.input-2-invert.get/set

12.1.6.1 Description

Invert second analogue input used for channel position command

12.1.6.2 Command

```
stage.mode.analogue-command.input-2-invert.get <channel>
```

```
stage.mode.analogue-command.input-2-invert.set <channel> <value>
```

12.1.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

12.1.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Invert analogue input 2

12.1.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Invert analogue input 2

12.1.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

12.1.6.7 Minimum security level

stage.mode.analogue-command.input-2-invert.get	User level security required
stage.mode.analogue-command.input-2-invert.set	Superuser level security required

12.1.6.8 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

12.1.7 stage.mode.analogue-command.input-select.get/set

12.1.7.1 Description

Analogue input used for channel position command

12.1.7.2 Command

```
stage.mode.analogue-command.input-select.get <channel>  
stage.mode.analogue-command.input-select.set <channel> <value>
```

12.1.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

12.1.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	String	Enumeration “analogue-command-input-select” or input number			1 = analogue input 1 2 = analogue input 2 3 = analogue input 3 “digital-input-control” = analogue input 1 or 2 selected by digital input

For controllers with dedicated analogue inputs, this will always be set to values 1-3 for channels 1-3 respectively. Attempting to set this to any other value will report an error “Value out of range”.

For controllers with switchable analogue inputs, attempting to set this to values greater than reported by controller.analogue-inputs.get (see 12.1.4) will report an error “Value out of range”.

12.1.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “analogue-command-input-select” or input number			1 = analogue input 1 2 = analogue input 2 3 = analogue input 3 “digital-input-control” = analogue input 1 or 2 selected by digital input

12.1.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

12.1.7.7 Minimum security level

stage.mode.analogue-command.input-select.get	User level security required
stage.mode.analogue-command.input-select.set	Superuser level security required

12.1.7.8 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

12.1.8 stage.command.analogue.scaling.get

12.1.8.1 Description

Analogue input/output scaling between position and voltage

12.1.8.2 Command

```
stage.command.analogue.scaling.get <channel>
```

12.1.8.3 Parameters

None

12.1.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	µm/V			Analogue input/output scaling between position and voltage

For angular stages, this will report a position scaling in µrad/V instead (see 3.1.4).

12.1.8.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

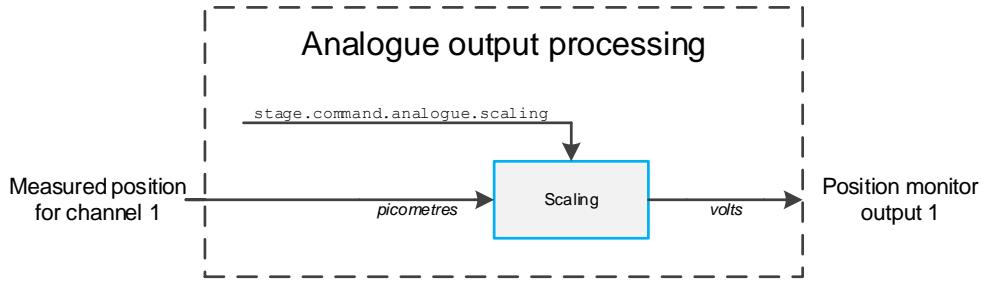
12.1.8.6 Minimum security level

stage.command.analogue.scaling.get	No security required
------------------------------------	----------------------

12.1.8.7 Supported in

Controller application firmware	6.6.14 onwards
Controller interface library	2.6.15 onwards

12.2 Analogue outputs



The position monitor analogue output uses the same scaling from position to voltage as the analogue inputs. Unlike the analogue inputs, there is no provision to invert the output direction, so increasing voltage always represents increasing position.

For controllers with dedicated analogue inputs, each channel has its own dedicated position monitor output.

For controllers with switchable analogue inputs and two analogue inputs, there are also two position monitor outputs. Both outputs are wired together, and report the position for the (single) channel.

Note that the position monitor analogue outputs may also be used for monitoring of internal scope points, instead of monitoring stage position. See 18 for details of scope monitoring.

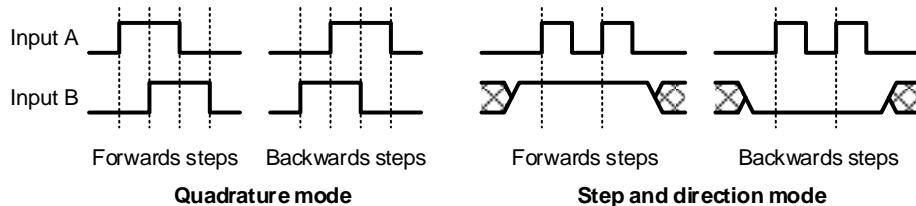
13 Stepped inputs, outputs and position command

13.1 Stepped inputs

Stepper motor controllers commonly use a 2-wire interface to command steps in either direction. This 2-wire interface is therefore also often available in motion control systems, and can provide a simple interfacing method for high-resolution, high-speed control. Since the interface only requires two digital outputs, this can be an extremely cost-effective solution for control from external equipment, compared to the cost and complexity of an equivalent analogue output. As a digital interface, it is also largely immune to electrical noise.

The stepped inputs may be configured as quadrature or step and direction interfaces. For each forwards or backwards step received, the commanded position moves by a configurable step in the relevant direction.

For quadrature mode, input A leads input B when commanding forwards movement on the stage; and input B leads input A when commanding backwards movement. With the standard Gray Code system as used for stepper motor controllers, each edge on each input indicates a step in the appropriate direction. For step and direction mode, input B controls whether a step is to be forwards or backwards, and a rising edge on input A triggers a step in the appropriate direction.



To ensure the controller can be easily integrated with other systems, this is also configurable to reverse direction, and for step and direction mode to trigger on rising or falling edge. Where control via stepped inputs is not required, of course it is possible to disable the stepped input.

The stepped input configuration is saved in the stage preset along with other system data (see 5.2). Different presets may therefore be configured with or without stepped inputs enabled, and with different step sizes, as required.

Debouncing is provided for stepped inputs if required, so that each input must be stable for the specified time before the new value is picked up. The same debounce time is applied for both inputs.

Debouncing is typically not required for quadrature mode, even if driven from a physical switch, because Gray Code ensures the stepped command will merely jitter one step forwards or backwards before settling on the final value. However it may be of value in step and direction mode, where spurious steps on input A will cause significant errors in the system. It may also be required for any system where significant noise on signal cables is expected. Debouncing will affect the rate at which the stepped command can be updated, so it should be used with caution.

Stepped inputs must not be changed faster than they can be read by the controller, otherwise steps may be missed. The following timings must be observed when debouncing is disabled. When debouncing is enabled, the limit will naturally be set by the debounce time instead.

Interface mode	Timing parameter	Absolute minimum	Recommended limit	Absolute maximum
Quadrature	Edge on input to edge on opposite input ("hold time")	20ns	40ns	
	Step rate for both inputs combined		25MHz	50MHz
Step and direction	Change of state on input B to trigger edge on input A ("settle time")	20ns	40ns	
	Trigger edge on input A to change of state on input B ("hold time")	20ns	40ns	
	Edge on input A to edge on input A ("hold time")	20ns	40ns	
	Step rate		12.5MHz	25MHz

Note that these limits relate to the controller electronics only. The cable type and length will cause some degradation of signals, as will other factors such as ferrite beads for EM emissions reduction. Achievable transmission rates in a customer's application will therefore typically be lower, and customers must carry out appropriate design investigations to ensure signals reach the controller reliably.

13.1.1 stage.stepped.input.enable.get/set**13.1.1.1 Description**

Stepped input enable

13.1.1.2 Command

```
stage.stepped.input.enable.get <channel>
stage.stepped.input.enable.set <channel> <value>
```

13.1.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.1.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Enable/disable stepped input

13.1.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Enable/disable stepped input

13.1.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.1.1.7 Minimum security level

stage.stepped.input.enable.get	No security required
stage.stepped.input.enable.set	Superuser level security required

13.1.1.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.1.2 stage.stepped.input.is-quadrature.get/set

13.1.2.1 Description

Stepped input interface mode

13.1.2.2 Command

```
stage.stepped.input.is-quadrature.get <channel>
```

```
stage.stepped.input.is-quadrature.set <channel> <value>
```

13.1.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.1.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Stepped input interface mode 1 = quadrature 0 = step and direction

13.1.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stepped input interface mode 1 = quadrature 0 = step and direction

13.1.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.1.2.7 Minimum security level

stage.stepped.input.is-quadrature.get	No security required
stage.stepped.input.is-quadrature.set	Superuser level security required

13.1.2.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.1.3 stage.stepped.input.reverse-direction.get/set**13.1.3.1 Description**

Stepped input reverse direction

13.1.3.2 Command

```
stage.stepped.input.reverse-direction.get <channel>
stage.stepped.input.reverse-direction.set <channel> <value>
```

13.1.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.1.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	<p>Stepped input reverse direction</p> <p>1 = reversed (opposite directions from diagram in 13.1)</p> <p>0 = normal</p>

13.1.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	<p>Stepped input reverse direction</p> <p>1 = reversed (opposite directions from diagram in 13.1)</p> <p>0 = normal</p>

13.1.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Value out of range	Data value must be within specified range (set only)

13.1.3.7 Minimum security level

stage-stepped.input.reverse-direction.get	No security required
stage-stepped.input.reverse-direction.set	Superuser level security required

13.1.3.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.1.4 stage.stepped.input.debounce-time.get/set**13.1.4.1 Description**

Stepped input debounce time

13.1.4.2 Command

```
stage.stepped.input.debounce-time.get <channel>
```

```
stage.stepped.input.debounce-time.set <channel> <value>
```

13.1.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.1.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	0	10	Stepped input debounce time

13.1.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0	10	Stepped input debounce time

13.1.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.1.4.7 Minimum security level

stage.stepped.input.debounce-time.get	No security required
stage.stepped.input.debounce-time.set	Superuser level security required

13.1.4.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.1.5 stage.steped.input.step-direction.is-rising-edge.get/set**13.1.5.1 Description**

Stepped input step and direction interface mode triggers a step on rising or falling edge

13.1.5.2 Command

```
stage.steped.input.step-direction.is-rising-edge.get <channel>
```

```
stage.steped.input.step-direction.is-rising-edge.set <channel>
<value>
```

13.1.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.1.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	<p>Stepped input step and direction trigger</p> <p>1 = triggered on rising edge of input A</p> <p>0 = triggered on falling edge of input A</p>

13.1.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	<p>Stepped input step and direction trigger</p> <p>1 = triggered on rising edge of input A</p> <p>0 = triggered on falling edge of input A</p>

13.1.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Value out of range	Data value must be within specified range (set only)

13.1.5.7 Minimum security level

stage-stepped.input.step-direction.is-rising-edge.get	No security required
stage-stepped.input.step-direction.is-rising-edge.set	Superuser level security required

13.1.5.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.2 Stepped position command

If the stepped input is enabled, every input step forwards or backwards will increase or decrease the stepped position command `stage.position.stepped-command` by one step. The stepped command may also be set directly by the user from the host PC, either as a position or in steps.

If a stepped position command is present and the step size is changed (either directly by the host PC or indirectly by the host PC selecting a new preset), then the stepped position command in steps will be recalculated such the command in picometres is unchanged, within limits of resolution.

The number of steps is a signed 32-bit value and therefore can count up to $\pm 2 \times 10^9$ steps. When choosing the step size, the stage resolution and the number of steps available must be considered. More usually though, the step size is limited by the speed of the stage (or the desired speed of response) and the maximum step rate, which is 25MHz for quadrature inputs (see 13.1).

For example, if the stage is to move end-to-end in 0.02s, then the step size must be at least 1/500,000th of the stage range. If the stage range is 100μm, this sets the minimum step size resolution at 200pm.

Because the stepped command may be set by the PC as well, it is also possible to implement a “coarse-fine” scheme where coarse changes are sent from the host PC over the comms link more slowly, and fine changes are sent from the stepped inputs more quickly. This may be of interest in systems with an external control loop, where the “central” position is often changed infrequently but fine correction is required to hold station accurately.

For example, suppose the step size is set at 1pm. Within 1ms, the stepped inputs can command movement of up to $\pm 25\text{nm}$ with very high precision, which may be desirable for accurate control. The host PC can still command coarse movement of up to $\pm 2\text{mm}$ range with the same step size resolution.

Since the stepped position command `stage.position.stepped-command` and digital command `stage.position.command` are separate entities, the same “coarse-fine” concept may also be used with the stepped position command solely coming from the stepped inputs. The digital command may then be adjusted for coarse positioning. Note also that the absolute position command `stage.position.absolute-command` may be used to adjust the digital position to a new “central” position without needing to allow for any current stepped position command.

With stepped position commands received from a source such as a thumbwheel, there is a risk that the command could be “spun” substantially past the end of the stage travel range. The stepped position command is therefore limited to the same range as the absolute position command, using the limits `stage.range.closed-loop-command.minimum` and `stage.range.closed-loop-command.maximum` (see 8.3.2 and 8.3.3).

13.2.1 stage.position.stepped-command.get/set**13.2.1.1 Description**

Stage stepped position command

13.2.1.2 Command

```
stage.position.stepped-command.get <channel>
stage.position.stepped-command.set <channel> <value>
```

13.2.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.2.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm			Stage stepped position command

13.2.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Stage stepped position command

13.2.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

13.2.1.7 Minimum security level

stage.position.stepped-command.get	No security required
stage.position.stepped-command.set	User level security required

13.2.1.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.2.2 stage.position.stepped-command.increment

13.2.2.1 Description

Stage stepped position command

13.2.2.2 Command

```
stage.position.stepped-command.increment <channel> <value>
```

13.2.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm			Increment value to add to stage stepped position command

13.2.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Increment value to add to stage stepped position command

13.2.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

13.2.2.6 Minimum security level

stage.position.stepped-command.increment	User level security required
--	------------------------------

13.2.2.7 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.2.3 stage.position.stepped-command.steps.get/set

13.2.3.1 Description

Stage stepped position command in steps

13.2.3.2 Command

```
stage.position.stepped-command.steps.get <channel>
```

```
stage.position.stepped-command.steps.set <channel> <value>
```

13.2.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.2.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit signed integer	steps			Stage stepped position command in steps

13.2.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit signed integer	steps			Stage stepped position command in steps

13.2.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

13.2.3.7 Minimum security level

stage.position.stepped-command.steps.get	No security required
stage.position.stepped-command.steps.set	User level security required

13.2.3.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.2.4 stage.position.stepped-command.steps.increment

13.2.4.1 Description

Stage stepped position command in steps

13.2.4.2 Command

stage.position.stepped-command.steps.increment <channel> <value>

13.2.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit signed integer	steps			Increment value to add to stage stepped position command in steps

13.2.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit signed integer	steps			Increment value to add to stage stepped position command in steps

13.2.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

13.2.4.6 Minimum security level

stage.position.stepped-command.steps.increment	User level security required
--	------------------------------

13.2.4.7 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.2.5 stage.position.stepped-command.step-size.get/set**13.2.5.1 Description**

Stage stepped position command step size

13.2.5.2 Command

```
stage.position.stepped-command.step-size.get <channel>
```

```
stage.position.stepped-command.step-size.set <channel> <value>
```

13.2.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.2.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm	0		Stage stepped position command step size

13.2.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		Stage stepped position command step size

13.2.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.2.5.7 Minimum security level

stage.position.stepped-command.step-size.get	No security required
stage.position.stepped-command.step-size.set	Superuser level security required

13.2.5.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3 Stepped outputs

As with stepped inputs, the stepped outputs can also be configured as quadrature or step and direction interfaces. Like the stepped inputs, they have the option to reverse direction, and to select for the step and direction interface whether steps are triggered on rising or falling edges.

A channel's stepped output can be derived from various sources.

- a simple echo of steps from the stepped input;
- reporting the stepped position command `stage.position.stepped-command` (see 13.2.1) or `stage.position.stepped-command.steps` (see 13.2.3);
- reporting the digital position command `stage.position.command` (see 8.3.4);
- reporting the absolute position command `stage.position.absolute-command` (see 8.3.1);
- reporting the measured position `stage.position.measured` (see 8.4.1).

The stepped output can also be disabled if not required.

The hold time for quadrature interface signals and the settle and hold times for step and direction interface signals are configurable. This allows step rates to be limited to what the receiving hardware or cables are able to handle. The following limits specify the maximum possible transmission speed for stepped outputs.

Interface mode	Timing parameter	Minimum	Maximum
Quadrature	Edge on output to edge on opposite output ("hold time")	20ns	
	Step rate for both outputs combined		50MHz
Step and direction	Change of state on output B to trigger edge on output A ("settle time")	40ns	
	Trigger edge on output A to change of state on output B ("hold time")	40ns	
	Edge on output A to edge on output A ("hold time")	40ns	
	Step rate		12.5MHz

Note that these limits relate to the controller electronics only. The cable type and length will cause some degradation of signals, as will other factors such as ferrite beads for EM emissions reduction. Achievable transmission rates in a customer's application will therefore typically be lower, and customers must carry out appropriate design investigations to ensure signals reach the customer's electronics reliably.

Regardless of the source of the stepped output signal, the output settle and hold times specified by the user will always be followed. In particular, the output settle and hold times will still be followed even if the stepped output is set to simply repeat steps received on the stepped input. It may therefore be possible for the input to change faster than the output can keep up. The output will still send the correct number of steps, but it will lag behind the input. The user must

be aware of this if using the stepped output for “handshaking” to verify that the controller has correctly received the stepped input command.

Where the stepped output reports a position, the step size can be set independently of the step size for the stepped input. If the step size is set too small and/or the step rate is set too slow, it is possible that position values reported on the stepped output may change slower than the slew rate of the stage. The user must ensure that the step size and step rate are configured appropriately for the system in which they will be used.

The nature of stepped I/O is that it transmits incremental changes to a value and not the actual value. For example, the initial stage position may be 5 μm and the step size configured to 10nm (0.01 μm) but the controller will not initially send any steps on the stepped output. When the stage position moves to 5.01um, the controller will only then send 1 “forwards” step. Any receiving electronics must therefore use other means to get this initial position, or must not care about the absolute position and only consider relative changes.

This may be inconvenient on startup if the system receiving the stepped output needs to know the actual value. The stepped output can therefore be configured to send steps corresponding to the full position value when the stage is connected (or if settings are changed). In the previous example, the controller would send 500 “forwards” steps on the stepped output when the controller starts. It would then send 1 “forwards” step when the stage position moves to 5.01um. Any receiving electronics must have previously reset its position measurement value, of course, otherwise these 500 steps will be incorrectly added to the previous value. Typically this would be handled by the host PC with some sequencing logic.

13.3.1 stage.stepped.output.select.get/set

13.3.1.1 Description

Stepped output source select

13.3.1.2 Command

```
stage.stepped.output.select.get <channel>  
stage.stepped.output.select.set <channel> <value>
```

13.3.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer				Stepped output source select 0 = stepped output disabled 1 = measured position, scaled by stage.stepped.output.step-size 2 = echo steps on stepped input 3 = stepped position command in steps 4 = stepped position command, scaled by stage.stepped.output.step-size 5 = digital position command, scaled by stage.stepped.output.step-size 6 = absolute position command, scaled by stage.stepped.output.step-size

13.3.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				<p>Stepped output source select</p> <p>0 = stepped output disabled</p> <p>1 = measured position, scaled by <code>stage.stepped.output.step-size</code></p> <p>2 = echo steps on stepped input</p> <p>3 = stepped position command in steps</p> <p>4 = stepped position command, scaled by <code>stage.stepped.output.step-size</code></p> <p>5 = digital position command, scaled by <code>stage.stepped.output.step-size</code></p> <p>6 = absolute position command, scaled by <code>stage.stepped.output.step-size</code></p>

13.3.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.1.7 Minimum security level

stage.stepped.output.select.get	No security required
stage.stepped.output.select.set	Superuser level security required

13.3.1.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.2 stage.stepped.output.is-quadrature.get/set**13.3.2.1 Description**

Stepped output interface mode

13.3.2.2 Command

```
stage.stepped.output.is-quadrature.get <channel>
```

```
stage.stepped.output.is-quadrature.set <channel> <value>
```

13.3.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Stepped output interface mode 1 = quadrature 0 = step and direction

13.3.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stepped output interface mode 1 = quadrature 0 = step and direction

13.3.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.2.7 Minimum security level

stage.stepped.output.is-quadrature.get	No security required
stage.stepped.output.is-quadrature.set	Superuser level security required

13.3.2.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.3 stage.stepped.output.reverse-direction.get/set**13.3.3.1 Description**

Stepped output reverse direction

13.3.3.2 Command

```
stage.stepped.output.reverse-direction.get <channel>
```

```
stage.stepped.output.reverse-direction.set <channel> <value>
```

13.3.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Stepped output reverse direction 1 = reversed (opposite directions from diagram in 13.1) 0 = normal

13.3.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stepped output reverse direction 1 = reversed (opposite directions from diagram in 13.1) 0 = normal

13.3.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Value out of range	Data value must be within specified range (set only)

13.3.3.7 Minimum security level

stage-stepped.output.reverse-direction.get	No security required
stage-stepped.output.reverse-direction.set	Superuser level security required

13.3.3.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.4 stage.stepped.output.step-direction.is-rising-edge.get/set**13.3.4.1 Description**

Stepped output step and direction interface mode triggers a step on rising or falling edge

13.3.4.2 Command

```
stage.stepped.output.step-direction.is-rising-edge.get <channel>
```

```
stage.stepped.output.step-direction.is-rising-edge.set <channel>
<value>
```

13.3.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Stepped output step and direction trigger 1 = triggered on rising edge of output A 0 = triggered on falling edge of output A

13.3.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stepped output step and direction trigger 1 = triggered on rising edge of output A 0 = triggered on falling edge of output A

13.3.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Value out of range	Data value must be within specified range (set only)

13.3.4.7 Minimum security level

stage-stepped.output.step-direction.is-rising-edge.get	No security required
stage-stepped.output.step-direction.is-rising-edge.set	Superuser level security required

13.3.4.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.5 stage.stepped.output.step-size.get/set**13.3.5.1 Description**

Stepped output position step size

13.3.5.2 Command

```
stage.stepped.output.step-size.get <channel>
stage.stepped.output.step-size.set <channel> <value>
```

13.3.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm	0		Stepped output position step size

13.3.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm	0		Stepped output position step size

13.3.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.5.7 Minimum security level

stage.stepped.output.step-size.get	No security required
stage.stepped.output.step-size.set	Superuser level security required

13.3.5.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.6 stage.stepped.output.send-full-value.get/set

13.3.6.1 Description

Stepped output send full value or incremental changes

13.3.6.2 Command

```
stage.stepped.output.send-full-value.get <channel>
```

```
stage.stepped.output.send-full-value.set <channel> <value>
```

13.3.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Stepped output send full value or incremental changes 1 = On stage connection or power-on, send steps corresponding to current value of parameter selected for stepped output. Send incremental changes thereafter. 0 = On stage connection or power-on, do nothing. Send incremental changes from initial position thereafter.

13.3.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stepped output send full value or incremental changes 1 = On stage connection or power-on, send steps corresponding to current value of parameter selected

Name	Type	Units	Minimum	Maximum	Description
					for stepped output. Send incremental changes thereafter. 0 = On stage connection or power-on, do nothing. Send incremental changes from initial position thereafter.

13.3.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.6.7 Minimum security level

stage.stepped.output.send-full-value.get	No security required
stage.stepped.output.send-full-value.set	Superuser level security required

13.3.6.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.7 stage.stepped.output.quadrature.hold-time.get/set

13.3.7.1 Description

Stepped output quadrature hold time

13.3.7.2 Command

```
stage.stepped.output.quadrature.hold-time.get <channel>  
stage.stepped.output.quadrature.hold-time.set <channel> <value>
```

13.3.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	0	10	Stepped output quadrature hold time

13.3.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0	10	Stepped output quadrature hold time

13.3.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.7.7 Minimum security level

stage.stepped.output.quadrature.hold-time.get	No security required
stage.stepped.output.quadrature.hold-time.set	Superuser level security required

13.3.7.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.8 stage.stepped.output.step-direction.settle-time.get/set

13.3.8.1 Description

Stepped output step and direction settle time

13.3.8.2 Command

stage.stepped.output.step-direction.settle-time.get <channel>

stage.stepped.output.step-direction.settle-time.set <channel> <value>

13.3.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	0	10	Stepped output step and direction settle time

13.3.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0	10	Stepped output step and direction settle time

13.3.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.8.7 Minimum security level

stage.stepped.output.step-direction.settle-time.get	No security required
stage.stepped.output.step-direction.settle-time.set	Superuser level security required

13.3.8.8 Supported in

Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

13.3.9 stage.stepped.output.step-direction.hold-time.get/set

13.3.9.1 Description

Stepped output step and direction hold time

13.3.9.2 Command

stage.stepped.output.step-direction.hold-time.get <channel>

stage.stepped.output.step-direction.hold-time.set <channel> <value>

13.3.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

13.3.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	0	10	Stepped output step and direction hold time

13.3.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0	10	Stepped output step and direction hold time

13.3.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data value must be within specified range (set only)

13.3.9.7 Minimum security level

stage.stepped.output.step-direction.hold-time.get	No security required
stage.stepped.output.step-direction.hold-time.set	Superuser level security required

13.3.9.8 Supported in

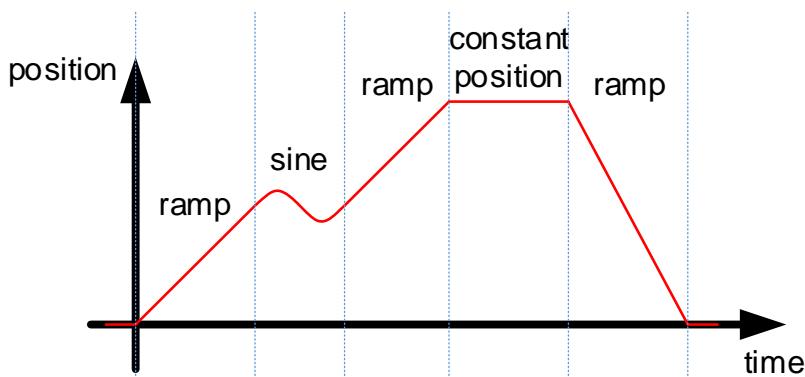
Controller application firmware	6.2.12 onwards
Controller interface library	2.2.7 onwards

14 Function playback

14.1 Overview

Function playback allows pre-programmed position command waveforms to be played back on one or more channels. This allows the user to construct complex steps or ramps, sine waves, raster scans, or any other waveform as required. For less sophisticated controllers, this would typically require the PC to be fitted with a high-resolution DAC interface to drive the controller's analogue input, and software to be written which would send the appropriate commands to the DAC. With function playback this can be achieved much more easily, with much greater accuracy, and without requiring expensive additional hardware or complex software.

Each channel controlling a stage has its own function playback and waveform setup. Function playback may be run independently for each channel, where stages control elements of a system which are not connected. Alternatively function playback can be run simultaneously for two or three channels on multi-axis systems, allowing complex 2D or 3D paths to be programmed.



Function playback at its most basic level allows a sampled waveform to control a channel's commanded position. Up to 10s of function playback can be programmed sample-accurately for each channel's waveform, updating the position command at the controller's 50kHz sample rate. Longer waveforms (of any duration) can be created by setting a greater time between samples, trading off function duration against accuracy. 14.2 describes how to configure the waveform data points. Any sampled waveform can be created to exactly match the user's requirements.

Calculating a waveform for complex profiles such as S-curve accelerations is often not straightforward. For this reason, the waveform generator allows a waveform to be constructed from segments which carry out steps, ramps, acceleration/deceleration and other move profiles. Section 15 describes the waveform generator in detail.

In general it is highly recommended that the waveform generator should be used where possible. It produces the required sampled waveform more easily, with all calculations handled by the controller; and it also generally allows the waveform to be set up faster, because there is no need to transmit points sample-by-sample to the controller. The user should normally only calculate the sampled waveform themselves if they have specific requirements which cannot be met by the waveform generator.

For even greater ease of use, the waveform builder provides pre-programmed waveform types for some frequently-used applications. Section 16 describes the waveform builder in detail.

When the waveform is ready for use, function playback can be started directly by the PC, or can be started by external electronics using digital trigger inputs. 14.4 and 14.5 describe how to start and stop function playback from the PC and from digital trigger inputs respectively. Function playback will normally run to completion and stop on its own when completed, but it may also be paused or stopped early by the user if required. For obvious reasons, function playback waveforms cannot be modified whilst playback is running for a channel, or whilst digital input triggers have been enabled to allow function playback to be started by external equipment.

Similarly to scope measurement and snapshot (see 17 and 18), the controller provides an additional “internal” function playback waveform which is accessed using channel number 0. This is not associated with a stage channel, and on its own will have no effect. However when routed to a scope output (see 18), it allows the user to turn an unused analogue output into a programmable signal generator, configured in the same way as any other function playback waveform. Since function playback can be run simultaneously for multiple channels, this waveform can be synchronised with stage movement waveforms, providing an integrated solution for controlling external electronics. Alternatively it could be simply used as a means of providing a separate signal to external electronics, saving cost and complexity in the system by removing the need for an additional PC-controlled DAC.

Note that whilst function playback may be used to configure any waveform within the limits available to the controller, the user must ensure that the stage is physically able to follow this position command waveform, as is the case for any other position command input. If the user specifies a ramp or step which is faster than the maximum speed possible for the stage with the current control loop settings, for example, the stage will only move as fast as it is able to go and will not track the function playback ramp correctly.

14.2 Waveform configuration using sampled data points

14.2.1 Waveform entry

The waveform for a channel allows a maximum of 10s of sample-accurate data, which on the 6000-series controller is 500,000 data points at a sample rate of 50kHz. The waveform is entered point by point, and the number of samples in the waveform is specified. For longer waveforms, the sample period may be extended. 500,000 points at a sample rate of 1ms would allow up to 500s of waveform playback, for instance. The sample period specified will be rounded to the nearest multiple of the DSP sample period (20 μ s for 50kHz).

As can be seen in 8.2.2, the function playback command is summed with all other command sources. This allows other complex behaviour to be set up which combines pre-programmed waveforms with other position commands. For example, a fine auto-focus could be implemented by programming a triangular waveform which automatically sweeps the stage over a $\pm 10\text{nm}$ offset from the current commanded position until the correct focus point is found.

Since the function playback command is summed with all other command sources, it is set to zero when function playback is not running. The first segment should therefore always start at zero, or may step from zero to another position if a smooth start is not required. A controlled end to the waveform is often required, in which case the last segment should also return to zero via a ramp or other smooth transition. If this is unimportant, the waveform may finish at any value and the controller will step back to the original position as the function playback command steps back to zero.

If the waveform is required to finish at a different position and hold that position after function playback has completed, it is possible to set a non-zero final position and set the waveform to “soft” stop at the end. In this case the digital position command is set to the combined position command at the end of the waveform (see the diagram in 8.2.2), so that the absolute commanded position holds its value when the function stops. See also 14.4 for details of commanding a “hard” or “soft” stop whilst the function is running, which behaves identically.

The last segment may also finish at a different position if the waveform is set to repeat forever (see 14.3), in which case any position after it stops does not need to be considered.

14.2.2 Ramped or stepped position command with longer sample periods

When the sample period is extended, by default the controller will ramp from the commanded position at the previous sample to the new commanded position in this sample, over the sample period time. This gives a smoother waveform as the sample rate is extended.

The user also has the option of selecting to step directly to a point and hold that value for the duration of the sample period, giving a “staircase” waveform as the sample period becomes longer. This may be selected for each point as required, using the command `function.waveform.command-transition` (see 14.2.6).

The majority of applications will not use this, because it greatly increases the inaccuracy of any waveform following a ramp or curve. However it can be useful for applications which require a hard-edged step between points, for example if setting up a square wave to test the system’s slew rate.

Note that this is only relevant for longer sample periods. At the default sample period (20 μ s) the commanded position is updated sample-accurately.

14.2.3 function.waveform.data.get/set

14.2.3.1 Description

Function playback waveform data.

14.2.3.2 Command

```
function.waveform.data.get <channel> <index>  
function.waveform.data.set <channel> <index> <value>
```

14.2.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point

14.2.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point
value	32-bit floating-point	pm			Data value at this point

14.2.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Data value at this point

14.2.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Sample index must be within specified range
Value out of range	Data value must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.2.3.7 Minimum security level

function.waveform.data.get	No security required
function.waveform.data.set	User level security required

14.2.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.2.4 function.waveform.count.get/set

14.2.4.1 Description

Function playback waveform data length

14.2.4.2 Command

```
function.waveform.count.get <channel>  
function.waveform.count.set <channel> <value>
```

14.2.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.2.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	1	See 14.2	Waveform data length

14.2.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	1	See 14.2	Waveform data length

14.2.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Data length must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.2.4.7 Minimum security level

function.waveform.count.get	No security required
function.waveform.count.set	User level security required

14.2.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.2.5 function.waveform.sample-period.get/set

14.2.5.1 Description

Period of samples for function playback

14.2.5.2 Command

```
function.waveform.sample-period.get <channel>  
function.waveform.sample-period.set <channel> <value>
```

14.2.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

14.2.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Period of samples for function playback

The sample period will be rounded to the nearest multiple of the controller's sample period (20µs for the controller's 50kHz sample rate). The result returned may therefore not be exactly the value set, if rounding is required.

14.2.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Period of samples for function playback

14.2.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Sample period must be greater than zero (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

14.2.5.7 Minimum security level

function.waveform.sample-period.get	No security required
function.waveform.sample-period.set	User level security required

14.2.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.2.6 function.waveform.command-transition.get/set

14.2.6.1 Description

Function playback waveform ramp or step to command

14.2.6.2 Command

```
function.waveform.command-transition.get <channel> <index>
```

```
function.waveform.command-transition.set <channel> <index> <value>
```

14.2.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point

14.2.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point
value	Boolean		0	1	Sample transition 0 = step to sample value 1 = ramp to sample value over function. waveform.command- transition period

14.2.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	Boolean		0	1	Sample transition 0 = step to sample value 1 = ramp to sample value over function. waveform.command- transition period

14.2.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Sample index must be within specified range
Value out of range	Number of steps must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.2.6.7 Minimum security level

function.waveform.command-transition.get	No security required
function.waveform.command-transition.set	User level security required

14.2.6.8 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

14.2.7 function.waveform.soft-stop-at-end.get/set**14.2.7.1 Description**

Function playback soft-stops at end (true) or steps to zero (false)

14.2.7.2 Command

```
function.waveform.soft-stop-at-end.get <channel>
```

```
function.waveform.soft-stop-at-end.set <channel> <value>
```

14.2.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.2.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Function playback soft-stops at end (true) or steps to zero (false)

14.2.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Function playback soft-stops at end (true) or steps to zero (false)

14.2.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Value must be as specified
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.2.7.7 Minimum security level

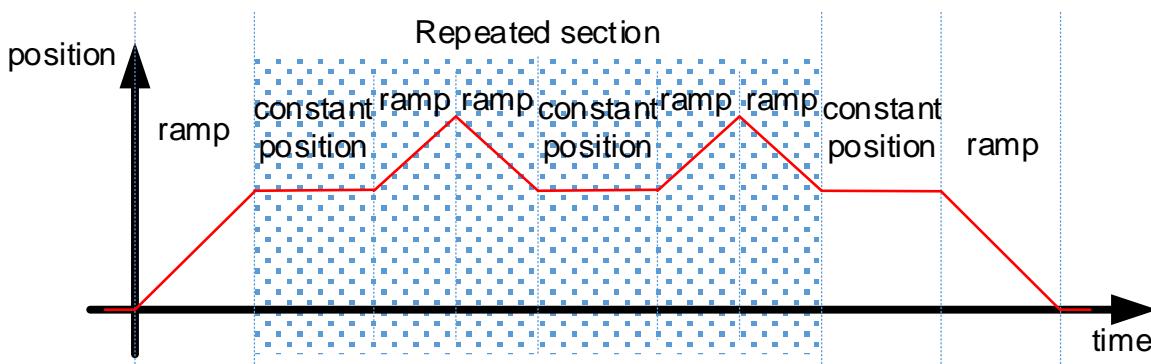
function.waveform.soft-stop-at-end.get	No security required
function.waveform.soft-stop-at-end.set	User level security required

14.2.7.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

14.3 Waveform design with repeated sections

It is often necessary to run some part of the waveform repeatedly. In the example below, the repeated section of the waveform holds a constant position for some time, then carries out a triangular ramp up and down.



Function playback allows a section of the waveform to be repeated a number of times as required (twice in the example above). The start and end sample indexes of the repeated section and the repeat count are specified. After the section has carried out the specified number of repeats, the remainder of the waveform is run. The waveform points at the start and end of the repeated section should usually be the same position, otherwise there will be a step change in the commanded position when repeating the section.

As the example above shows, repeated waveforms often require an initial section which moves the system to the required position, and a final section which returns the system to the initial point. Setting the start sample index after zero (i.e. not at the start of the entire waveform) gives an initial section which will be run once and not repeated; and setting the end sample index before the last point (i.e. not at the end of the entire waveform) gives a final section which will be run once after the repeated section.

If the entire waveform is to be repeated, initial or final sections need not be present. In this case the start sample index would be set to zero and the end sample index would be set to the index of the last sample in the waveform (one less than the sample count).

Repeated waveforms may also be required to repeat forever. This is selected by setting the repeat count to zero. If the waveform needs to be stopped, the user must stop it manually using an appropriate "stop" command (see 14.4 and 14.5). A final section for the waveform (after the repeated section) may exist but will never be carried out.

14.3.1 function.waveform.repeat-start.get/set

14.3.1.1 Description

Function playback waveform start point for repeating

14.3.1.2 Command

```
function.waveform.repeat-start.get <channel>
function.waveform.repeat-start.set <channel> <value>
```

14.3.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.3.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	0	See 14.2	Function playback waveform start point for repeating

14.3.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 14.2	Function playback waveform start point for repeating

14.3.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Start index must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.3.1.7 Minimum security level

function.waveform.repeat-start.get	No security required
function.waveform.repeat-start.set	User level security required

14.3.1.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

14.3.2 function.waveform.repeat-end.get/set

14.3.2.1 Description

Function playback waveform end point for repeating

14.3.2.2 Command

```
function.waveform.repeat-end.get <channel>  
function.waveform.repeat-end.set <channel> <value>
```

14.3.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.3.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	0	See 14.2	Function playback waveform end point for repeating

14.3.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 14.2	Function playback waveform end point for repeating

14.3.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	End index must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.3.2.7 Minimum security level

function.waveform.repeat-end.get	No security required
function.waveform.repeat-end.set	User level security required

14.3.2.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

14.3.3 function.waveform.repeat-count.get/set**14.3.3.1 Description**

Function playback number of repeats of waveform (0=repeat forever)

14.3.3.2 Command

```
function.waveform.repeat-count.get <channel>
function.waveform.repeat-count.set <channel> <value>
```

14.3.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.3.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer				Function playback number of repeats of waveform 0 = repeat forever 1 = run once (as normal) 2+ = repeat this many times

14.3.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Function playback number of repeats of waveform 0 = repeat forever 1 = run once (as normal) 2+ = repeat this many times

14.3.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.3.3.7 Minimum security level

function.waveform.repeat-count.get	No security required
function.waveform.repeat-count.set	User level security required

14.3.3.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

14.4 Function playback control from host PC

Function playback may be started, paused, unpause and stopped from the host PC. This may be carried out for a single channel individually, or for multiple channels synchronously. Snapshot capture may also be triggered synchronously with function playback.

Once function playback is started, a function with a fixed number of waveform iterations will run to completion and will automatically stop at that point, unless the user chooses to stop it before it has completed. A function which is set to run indefinitely will never complete, of course, and therefore must be stopped by the user.

When stopping function playback, the user has the option of a “hard” or “soft” stop in the same way as when playback stops naturally at the end (see 14.2). On a “soft” stop, the digital position command value is set to the combined position command (see the diagram in 8.2.2), so that the absolute commanded position holds its value at the time the function stops. A “hard” stop simply sets the function playback command (see the diagram in 8.2.2) to zero, so that the commanded position has a step-change back to the digital position command. The relevant stop command to use will depend on the user’s requirements.

Function playback may be paused at any time. The function playback commanded position will hold its value until function playback is unpause, and will then resume from that point.

Function playback may be stopped (see below) whilst the function is paused, if required. Note that pausing function playback does not affect other position command sources, which will still contribute to the absolute position command as shown in 8.2.2.

Commands allow all channels to be controlled simultaneously, setting flags for the channels to be controlled as required. This ensures function playback for multiple channels and/or snapshot capture is entirely synchronous. When only a subset of channels require function playback to be controlled, care must be taken to ensure the state of other channels is not accidentally affected.

Note that the command `function.waveform-generator.prepare-start` (see 15.7.8) may also be used to start function playback immediately after preparing the waveform. This command operates in the same way as commands specified here.

Repeated commands have no effect. In particular, if function playback is currently running then a start command will have no effect. There is no facility to retrigger a function, only to stop it and restart it as separate commands.

The number of parameters and return values for the following commands will vary depending on the number of controller channels. Where the controller only has 2 channels, the parameter for channel 3 is not required, and there will be no return value for channel 3. Similarly for a single-channel controller, parameters for channels 2 and 3 are not required, and there will be no return values for channels 2 and 3. The interface DLL ignores redundant parameters, so providing extra parameters for missing channels will not cause a problem. However calling code will need to be aware that the number of return values will be dependent on the number of controller channels, and must not assume that return values for channels 2 and 3 will always be present.

14.4.1 function.state.get

14.4.1.1 Description

Report the state of function playback and snapshot capture

14.4.1.2 Command

function.state.get

14.4.1.3 Parameters

None

14.4.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
running-snapshot	8-bit unsigned integer	Boolean	0	1	Running snapshot capture
running-internal-channel0	8-bit unsigned integer	Boolean	0	1	Running function playback for internal channel 0
running-channel1	8-bit unsigned integer	Boolean	0	1	Running function playback for channel 1
running-channel2	8-bit unsigned integer	Boolean	0	1	Running function playback for channel 2 (only returned for 2- or 3-channel controller)
running-channel3	8-bit unsigned integer	Boolean	0	1	Running function playback for channel 3 (only returned for 3-channel controller)
paused-internal-channel0	8-bit unsigned integer	Boolean	0	1	Paused function playback for internal channel 0
paused-channel1	8-bit unsigned integer	Boolean	0	1	Paused function playback for channel 1
paused-channel2	8-bit unsigned integer	Boolean	0	1	Paused function playback for channel 2 (only returned for 2- or 3-channel controller)
paused-channel3	8-bit unsigned integer	Boolean	0	1	Paused function playback for channel 3 (only returned for 3-channel controller)

14.4.1.5 Possible error reports

None

14.4.1.6 Minimum security level

function.state.get	No security required
--------------------	----------------------

14.4.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.4.2 function.command.start

14.4.2.1 Description

Start function playback and/or snapshot

14.4.2.2 Command

```
function.command.start <start-snapshot> <start-internal-channel0>
<start-channel1> [start-channel2] [start-channel3]
```

14.4.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
start-snapshot	8-bit unsigned integer	Boolean	0	1	Start snapshot capture
start-internal-channel0	8-bit unsigned integer	Boolean	0	1	Start function playback for internal channel 0
start-channel1	8-bit unsigned integer	Boolean	0	1	Start function playback for channel 1
start-channel2	8-bit unsigned integer	Boolean	0	1	Start function playback for channel 2 (only required for 2- or 3-channel controller)
start-channel3	8-bit unsigned integer	Boolean	0	1	Start function playback for channel 3 (only required for 3-channel controller)

14.4.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
start-snapshot	Boolean		0	1	Started snapshot capture
start-internal-channel0	Boolean		0	1	Started function playback for internal channel 0
start-channel1	Boolean		0	1	Started function playback for channel 1
start-channel2	Boolean		0	1	Started function playback for channel 2 (only returned for 2- or 3-channel controller)

Name	Type	Units	Minimum	Maximum	Description
start-channel3	Boolean		0	1	Started function playback for channel 3 (only returned for 3-channel controller)

14.4.2.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.4.2.6 Minimum security level

function.command.start	User level security required
------------------------	------------------------------

14.4.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.4.3 function.command.stop

14.4.3.1 Description

Hard-stop function playback and/or stop snapshot

14.4.3.2 Command

```
function.command.stop <stop-snapshot> <stop-internal-channel0> <stop-
channel1> [stop-channel2] [stop-channel3]
```

14.4.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
stop-snapshot	8-bit unsigned integer	Boolean	0	1	Stop snapshot
stop-internal-channel0	8-bit unsigned integer	Boolean	0	1	Stop function playback for internal channel 0
stop-channel1	8-bit unsigned integer	Boolean	0	1	Stop function playback for channel 1
stop-channel2	8-bit unsigned integer	Boolean	0	1	Stop function playback for channel 2 (only required for 2- or 3-channel controller)
stop-channel3	8-bit unsigned integer	Boolean	0	1	Stop function playback for channel 3 (only required for 3-channel controller)

14.4.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
stop-snapshot	8-bit unsigned integer	Boolean	0	1	Stopped snapshot capture
stop-internal-channel0	8-bit unsigned integer	Boolean	0	1	Stopped function playback for internal channel 0
stop-channel1	8-bit unsigned integer	Boolean	0	1	Stopped function playback for channel 1
stop-channel2	8-bit unsigned integer	Boolean	0	1	Stopped function playback for channel 2 (only returned for 2- or 3-channel controller)

Name	Type	Units	Minimum	Maximum	Description
stop-channel3	8-bit unsigned integer	Boolean	0	1	Stopped function playback for channel 3 (only returned for 3-channel controller)

14.4.3.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.4.3.6 Minimum security level

function.command.stop	User level security required
-----------------------	------------------------------

14.4.3.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.4.4 function.command.soft-stop

14.4.4.1 Description

Soft-stop function playback and/or snapshot

14.4.4.2 Command

```
function.command.soft-stop <stop-snapshot> <stop-internal-channel0>
<stop-channel1> [stop-channel2] [stop-channel3]
```

14.4.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
stop-snapshot	8-bit unsigned integer	Boolean	0	1	Stop snapshot
stop-internal-channel0	8-bit unsigned integer	Boolean	0	1	Stop function playback for internal channel 0
stop-channel1	8-bit unsigned integer	Boolean	0	1	Stop function playback for channel 1
stop-channel2	8-bit unsigned integer	Boolean	0	1	Stop function playback for channel 2 (only required for 2- or 3-channel controller)
stop-channel3	8-bit unsigned integer	Boolean	0	1	Stop function playback for channel 3 (only required for 3-channel controller)

14.4.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
stop-snapshot	8-bit unsigned integer	Boolean	0	1	Stopped snapshot capture
stop-internal-channel0	8-bit unsigned integer	Boolean	0	1	Stopped function playback for internal channel 0
stop-channel1	8-bit unsigned integer	Boolean	0	1	Stopped function playback for channel 1
stop-channel2	8-bit unsigned integer	Boolean	0	1	Stopped function playback for channel 2 (only returned for 2- or 3-channel controller)

Name	Type	Units	Minimum	Maximum	Description
stop-channel3	8-bit unsigned integer	Boolean	0	1	Stopped function playback for channel 3 (only returned for 3-channel controller)

14.4.4.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.4.4.6 Minimum security level

function.command.stop	User level security required
-----------------------	------------------------------

14.4.4.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.4.5 function.command.pause

14.4.5.1 Description

Pause function playback

14.4.5.2 Command

```
function.command.pause <pause-snapshot> <pause-internal-channel0>
<pause-channel1> [pause-channel2] [pause-channel3]
```

14.4.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
pause-internal-channel0	8-bit unsigned integer	Boolean	0	1	Pause function playback for internal channel 0
pause-channel1	8-bit unsigned integer	Boolean	0	1	Pause function playback for channel 1
pause-channel2	8-bit unsigned integer	Boolean	0	1	Pause function playback for channel 2 (only required for 2- or 3-channel controller)
pause-channel3	8-bit unsigned integer	Boolean	0	1	Pause function playback for channel 3 (only required for 3-channel controller)

14.4.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
pause-internal-channel0	8-bit unsigned integer	Boolean	0	1	Paused function playback for internal channel 0
pause-channel1	8-bit unsigned integer	Boolean	0	1	Paused function playback for channel 1
pause-channel2	8-bit unsigned integer	Boolean	0	1	Paused function playback for channel 2 (only returned for 2- or 3-channel controller)
pause-channel3	8-bit unsigned integer	Boolean	0	1	Paused function playback for channel 3 (only returned for 3-channel controller)

14.4.5.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.4.5.6 Minimum security level

function.command.pause	User level security required
------------------------	------------------------------

14.4.5.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.4.6 function.command.unpause

14.4.6.1 Description

Unpause function playback

14.4.6.2 Command

```
function.command.unpause <unpause-snapshot> <unpause-internal-
channel0> <unpause-channel1> [unpause-channel2] [unpause-channel3]
```

14.4.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
unpause-internal-channel0	8-bit unsigned integer	Boolean	0	1	Unpause function playback for internal channel 0
unpause-channel1	8-bit unsigned integer	Boolean	0	1	Unpause function playback for channel 1
unpause-channel2	8-bit unsigned integer	Boolean	0	1	Unpause function playback for channel 2 (only required for 2- or 3-channel controller)
unpause-channel3	8-bit unsigned integer	Boolean	0	1	Unpause function playback for channel 3 (only required for 3-channel controller)

14.4.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
unpause-internal-channel0	8-bit unsigned integer	Boolean	0	1	Unpaused function playback for internal channel 0
unpause-channel1	8-bit unsigned integer	Boolean	0	1	Unpaused function playback for channel 1
unpause-channel2	8-bit unsigned integer	Boolean	0	1	Unpaused function playback for channel 2 (only returned for 2- or 3-channel controller)
unpause-channel3	8-bit unsigned integer	Boolean	0	1	Unpaused function playback for channel 3 (only returned for 3-channel controller)

14.4.6.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.4.6.6 Minimum security level

function.command.unpause	User level security required
--------------------------	------------------------------

14.4.6.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.4.6.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5 Function playback control from digital trigger inputs

The controller has three digital inputs which may be used to trigger function playback and snapshot capture. The manual for each controller specifies digital input connectors and logic levels.

Digital trigger inputs may be used to control function playback for individual channels, or may be used to control function playback on multiple channels synchronously, as well as controlling snapshot capture. All control commands from the host PC as identified in section 14.4 may be configured to be triggered by digital inputs, which potentially allows function playback and snapshot capture to be controlled from external electronics without requiring the host PC. Host PC control commands may still be used as required, in addition to any digital trigger input actions.

Each control command (start, hard-stop, soft-stop, pause, unpause) for each function playback channel and snapshot capture may be linked to a digital input trigger. Available triggers are:-

- Input high
- Input low
- Input rising-edge
- Input falling-edge

To clarify the difference in behaviour between “high” and “rising-edge”, if the input is initially high and function playback is set to start on “high”, then function playback will be started as soon as “high” is selected. If the input is high when function playback completes, then function playback will immediately be restarted. Conversely, if the input is initially high and “rising-edge” is selected, the input must go low and high again before function playback is started; and after function playback completes, the input must again go low and high again before function playback is restarted. “Low” and “falling-edge” behave similarly.

As with control commands from the host PC, repeated commands have no effect. Once function playback is running, further start commands (whether from a “high” or “low” input held in that state, or from transitions on a “rising-edge” or “falling-edge” input) have no effect until the function completes or stops.

Using a trigger input for one command does not prevent it also being used for other commands. An obvious example would be configuring “high” on one input to trigger start, and “low” on the same input to trigger stop. The same input could even be used to trigger different behaviour on different channels; for example, if movement on one axis requires the other axis to be stationary, high and low could trigger start and stop on one channel, and could trigger pause and unpause on the second channel.

Once trigger inputs are configured, digital input triggers for that channel must be enabled, at which point the function playback waveform and configuration may not be changed. To make changes to function playback for this channel, digital input triggers for that channel must be disabled and any running function playback must have completed or stopped.

When capturing a snapshot, the area of interest may be some time after a function has started. When commanding function playback and snapshot capture from the host PC, the host PC can simply wait for some time before commanding a snapshot, but this may not be possible when commanding function playback and snapshot capture from digital inputs. Snapshot capture may therefore have a delay set between an input trigger and snapshot capture starting, so that both function playback and snapshot capture can be triggered from the same input.

14.5.1 function.trigger-inputs.start.get/set

14.5.1.1 Description

Function playback channel start trigger inputs

14.5.1.2 Command

```
function.trigger-inputs.start.get <channel>  
function.trigger-inputs.start.set <channel> <value>
```

14.5.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.5.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.5.1.7 Minimum security level

function.trigger-inputs.start.get	No security required
function.trigger-inputs.start.get	User level security required

14.5.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.2 function.trigger-inputs.stop.get/set

14.5.2.1 Description

Function playback channel hard-stop trigger inputs

14.5.2.2 Command

```
function.trigger-inputs.stop.get <channel>  
function.trigger-inputs.stop.set <channel> <value>
```

14.5.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.5.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.5.2.7 Minimum security level

function.trigger-inputs.stop.get	No security required
function.trigger-inputs.stop.get	User level security required

14.5.2.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.3 function.trigger-inputs.soft-stop.get/set**14.5.3.1 Description**

Function playback channel soft-stop trigger inputs

14.5.3.2 Command

```
function.trigger-inputs.soft-stop.get <channel>
```

```
function.trigger-inputs.soft-stop.set <channel> <value>
```

14.5.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.5.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.5.3.7 Minimum security level

function.trigger-inputs.soft-stop.get	No security required
function.trigger-inputs.soft-stop.get	User level security required

14.5.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.4 function.trigger-inputs.pause.get/set**14.5.4.1 Description**

Function playback channel pause trigger inputs

14.5.4.2 Command

```
function.trigger-inputs.pause.get <channel>
function.trigger-inputs.pause.set <channel> <value>
```

14.5.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.5.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.5.4.7 Minimum security level

function.trigger-inputs.pause.get	No security required
function.trigger-inputs.pause.get	User level security required

14.5.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.5 function.trigger-inputs.unpause.get/set

14.5.5.1 Description

Function playback channel unpause trigger inputs

14.5.5.2 Command

```
function.trigger-inputs.unpause.get <channel>
function.trigger-inputs.unpause.set <channel> <value>
```

14.5.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

14.5.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			Trigger inputs Bit 0: Input 1 high

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Input 2 high Bit 2: Input 3 high Bit 8: Input 1 low Bit 9: Input 2 low Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.5.5.7 Minimum security level

function.trigger-inputs.unpause.get	No security required
function.trigger-inputs.unpause.get	User level security required

14.5.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.6 snapshot.trigger-inputs.start.get/set

14.5.6.1 Description

Snapshot capture start trigger inputs

14.5.6.2 Command

snapshot.trigger-inputs.start.get

snapshot.trigger-inputs.start.set <value>

14.5.6.3 Parameters (get)

None

14.5.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			<p>Trigger inputs</p> <p>Bit 0: Input 1 high</p> <p>Bit 1: Input 2 high</p> <p>Bit 2: Input 3 high</p> <p>Bit 8: Input 1 low</p> <p>Bit 9: Input 2 low</p> <p>Bit 10: Input 3 low</p> <p>Bit 16: Input 1 rising-edge</p> <p>Bit 17: Input 2 rising-edge</p> <p>Bit 18: Input 3 rising-edge</p> <p>Bit 24: Input 1 falling-edge</p> <p>Bit 25: Input 2 falling-edge</p> <p>Bit 26: Input 3 falling-edge</p>

14.5.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			<p>Trigger inputs</p> <p>Bit 0: Input 1 high</p> <p>Bit 1: Input 2 high</p> <p>Bit 2: Input 3 high</p> <p>Bit 8: Input 1 low</p> <p>Bit 9: Input 2 low</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.6.6 Possible error reports

Error return “errcode” string reported	Description
Command could not be carried out	Snapshot capture is running or waiting for digital input trigger, so configuration cannot be changed

14.5.6.7 Minimum security level

snapshot.trigger-inputs.start.get	No security required
snapshot.trigger-inputs.start.get	User level security required

14.5.6.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.7 snapshot.trigger-inputs.stop.get/set

14.5.7.1 Description

Snapshot capture stop trigger inputs

14.5.7.2 Command

snapshot.trigger-inputs.stop.get

snapshot.trigger-inputs.stop.set <value>

14.5.7.3 Parameters (get)

None

14.5.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			<p>Trigger inputs</p> <p>Bit 0: Input 1 high</p> <p>Bit 1: Input 2 high</p> <p>Bit 2: Input 3 high</p> <p>Bit 8: Input 1 low</p> <p>Bit 9: Input 2 low</p> <p>Bit 10: Input 3 low</p> <p>Bit 16: Input 1 rising-edge</p> <p>Bit 17: Input 2 rising-edge</p> <p>Bit 18: Input 3 rising-edge</p> <p>Bit 24: Input 1 falling-edge</p> <p>Bit 25: Input 2 falling-edge</p> <p>Bit 26: Input 3 falling-edge</p>

14.5.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	bitfield			<p>Trigger inputs</p> <p>Bit 0: Input 1 high</p> <p>Bit 1: Input 2 high</p> <p>Bit 2: Input 3 high</p> <p>Bit 8: Input 1 low</p> <p>Bit 9: Input 2 low</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 10: Input 3 low Bit 16: Input 1 rising-edge Bit 17: Input 2 rising-edge Bit 18: Input 3 rising-edge Bit 24: Input 1 falling-edge Bit 25: Input 2 falling-edge Bit 26: Input 3 falling-edge

14.5.7.6 Possible error reports

Error return “errcode” string reported	Description
Command could not be carried out	Snapshot capture is running or waiting for digital input trigger, so configuration cannot be changed

14.5.7.7 Minimum security level

snapshot.trigger-inputs.stop.get	No security required
snapshot.trigger-inputs.stop.get	User level security required

14.5.7.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.8 snapshot.trigger-inputs.start-delay.get/set

14.5.8.1 Description

Snapshot start trigger delay

14.5.8.2 Command

snapshot.trigger-inputs.start-delay.get

snapshot.trigger-inputs.start-delay.set <value>

14.5.8.3 Parameters (get)

None

14.5.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples			Snapshot start trigger delay

14.5.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples			Snapshot start trigger delay

14.5.8.6 Possible error reports

Error return “errcode” string reported	Description
Command could not be carried out	Snapshot capture is running or waiting for digital input trigger, so configuration cannot be changed

14.5.8.7 Minimum security level

snapshot.trigger-inputs.start-delay.get	No security required
snapshot.trigger-inputs.start-delay.get	User level security required

14.5.8.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.9 function.trigger-inputs.enabled.get

14.5.9.1 Description

Trigger inputs enabled for function playback and/or snapshot

14.5.9.2 Command

function.trigger-inputs.enabled.get

14.5.9.3 Parameters

None

14.5.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
enabled-snapshot	8-bit unsigned integer	Boolean	0	1	Enabled trigger inputs for snapshot capture
enabled-internal-channel0	8-bit unsigned integer	Boolean	0	1	Enabled trigger inputs for function playback for internal channel 0
enabled-channel1	8-bit unsigned integer	Boolean	0	1	Enabled trigger inputs for function playback for channel 1
enabled-channel2	8-bit unsigned integer	Boolean	0	1	Enabled trigger inputs for function playback for channel 2 (only returned for 2- or 3-channel controller)
enabled-channel3	8-bit unsigned integer	Boolean	0	1	Enabled trigger inputs for function playback for channel 3 (only returned for 3-channel controller)

14.5.9.5 Possible error reports

None

14.5.9.6 Minimum security level

function.trigger-inputs.enabled.get	No security required
-------------------------------------	----------------------

14.5.9.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.10 function.trigger-inputs.enabled.enable-inputs

14.5.10.1 Description

Enable trigger inputs for function playback and/or snapshot

Note that after trigger inputs are enabled for function playback or for snapshot capture, the relevant configuration may not be changed until trigger inputs are disabled.

14.5.10.2 Command

```
function.trigger-inputs.enabled.enable-inputs <enable-snapshot>
<enable-internal-channel0> <enable-channel1> [enable-channel2]
[enable-channel3]
```

14.5.10.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
enable-snapshot	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for snapshot capture
enable-internal-channel0	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for internal channel 0
enable-channel1	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for channel 1
enable-channel2	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for channel 2 (only required for 2- or 3-channel controller)
enable-channel3	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for channel 3 (only required for 3-channel controller)

14.5.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
enable-snapshot	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for snapshot capture
enable-internal-channel0	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for internal channel 0

Name	Type	Units	Minimum	Maximum	Description
enable-channel1	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for channel 1
enable-channel2	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for channel 2 (only returned for 2- or 3-channel controller)
enable-channel3	8-bit unsigned integer	Boolean	0	1	Enable trigger inputs for function playback for channel 3 (only returned for 3-channel controller)

14.5.10.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.5.10.6 Minimum security level

function.trigger-inputs.enabled.enable-inputs	User level security required
---	------------------------------

14.5.10.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.5.11 function.trigger-inputs.enabled.disable-inputs

14.5.11.1 Description

Disable trigger inputs for function playback and/or snapshot

14.5.11.2 Command

```
function.trigger-inputs.enabled.disable-inputs <disable-snapshot>
<disable-internal-channel0> <disable-channel1> [disable-channel2]
[disable-channel3]
```

14.5.11.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
disable-snapshot	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for snapshot capture
disable-internal-channel0	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for internal channel 0
disable-channel1	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for channel 1
disable-channel2	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for channel 2 (only required for 2- or 3-channel controller)
disable-channel3	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for channel 3 (only required for 3-channel controller)

14.5.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
disable-snapshot	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for snapshot capture
disable-internal-channel0	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for internal channel 0

Name	Type	Units	Minimum	Maximum	Description
disable-channel1	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for channel 1
disable-channel2	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for channel 2 (only returned for 2- or 3-channel controller)
disable-channel3	8-bit unsigned integer	Boolean	0	1	Disable trigger inputs for function playback for channel 3 (only returned for 3-channel controller)

14.5.11.5 Possible error reports

None

Note that if a flag is set for a channel which has no stage present, the command for that channel will be ignored. Since this command controls multiple channels, this is preferable to reporting an error when one channel cannot be controlled but other channels can. The return value for that channel will return zero, to indicate that the command could not be carried out.

14.5.11.6 Minimum security level

function.trigger-inputs.enabled.disable-inputs	User level security required
--	------------------------------

14.5.11.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.6 Control of external equipment with digital trigger outputs

14.6.1 Overview

The controller has one digital trigger output per channel which may be pulsed synchronously with the waveform during function playback to activate external equipment. The controller manual (EN-014635-UM for the 6000 series) specifies digital output connectors and logic levels, which will not be repeated here. Note that each channel only has one trigger output; there is no facility for one channel to trigger multiple outputs.

For each sample point in the function playback waveform, a trigger output event can be specified. This allows accurate synchronisation of trigger output pulses with the function playback waveform.

14.6.2 Trigger output event types

Older controllers typically only provided simple time-based triggering, where trigger outputs were pulsed at specified times during the waveform. Since the stage movement lags behind the commanded position, the user had to estimate how long this lag would be and allow for it in their trigger times. This was generally a source of inaccuracy in systems, causing issues such as imperfect image focusing.

The Queensgate controller can also trigger based on position. The controller does not activate the trigger output immediately. Instead it notes the commanded position at the trigger point, and activates the trigger output when the measured position reaches that commanded position. The system no longer needs to consider the lag between commanded position and stage movement, and accuracy is greatly improved. Various triggering options are available.

Trigger output event type	Description
none	No trigger is set at this trigger position.
pulse	Trigger output is activated immediately. This is the “traditional” time-based triggering method.
above-command	Trigger output is activated when measured position is greater than commanded position at trigger point. If measured position is already greater than commanded position, trigger output is activated immediately.
below-command	Trigger output is activated when measured position is lower than commanded position at trigger point. If measured position is already lower than commanded position, trigger output is activated immediately.
rising-edge	Trigger output is activated when measured position becomes greater than commanded position at trigger point.

Trigger output event type	Description
	If measured position is already greater than commanded position, measured position must fall below this commanded position value and then increase above this commanded position value again for the trigger output to be activated.
falling-edge	Trigger output is activated when measured position becomes lower than commanded position at trigger point. If measured position is already lower than commanded position, measured position must increase above this commanded position value and fall below this commanded position value again for the trigger output to be activated.
either-edge	Trigger output is activated when measured position crosses commanded position at trigger point in either direction.
in-position	Trigger output is activated when measured position is within the in-position error threshold (<code>stage.in-position.error-threshold</code> , see 8.5.1) of commanded position at trigger point. If measured position is already within this range, trigger output is activated immediately. Supported in firmware 6.4.1 onwards.

Whilst position-based triggering improves trigger accuracy, it is important to select the correct trigger event for the waveform trajectory, otherwise the trigger output may be pulsed too soon or may never be activated at all.

For a positive-going trajectory, the trigger events “above-command” or “rising-edge” would typically be used. Similarly the trigger events “below-command” or “falling-edge” would generally be used for a negative-going trajectory.

14.6.3 Factors to consider with position-based triggering

Whilst edge-based triggers may generally seem desirable, the user should use these with caution. If the system has some ringing which has been excited by a step or other acceleration/deceleration, it is possible that the stage may have overshot the commanded position at the trigger point, and the output will not be triggered until after the measured position returns past this commanded position value again. This is particularly an issue for triggers during ramp and acceleration segments. If triggers are being missed, especially during or after acceleration/deceleration where ringing often occurs, it may be more appropriate to use level-based triggers (“above-command” or “below-command”) so that the trigger is activated immediately if the stage overshoots.

The threshold-based and edge-based trigger events all require that the measured position crosses the command position. When stepping to a position with an underdamped system it may take a long time for this to occur; and at the peaks of a sine or sawtooth wave (such as the “start” trigger for segment 3 in the example above) it is possible that the stage may never fully

reach that peak before the command changes to pull the stage back down again. As an alternative to threshold-based or edge-based triggering, the “in-position” trigger event activates the trigger output when the measured position is close enough to the command position for the current application’s purposes. This uses the same “in position” error threshold as the control loop (see 8.5 and diagram 8.2.4). In optical applications for example, focusing only needs to be accurate to half a wavelength of the relevant light source.

Position-based triggering is highly accurate for slower-moving systems, but it does require the update rate to be considered. It takes at least one 50kHz update cycle for the controller to read the stage position and a second update cycle to update the trigger output, and timing tolerances can potentially add another update cycle to this time. If low-pass filtering is used on the position input (see 8.2.3 and 8.4), this can further delay the position signal. In extremely fast-moving systems, this delay could result in a significant position error – for example, if the stage is moving at 5mm/s then it will travel 100nm in each update cycle, meaning that the trigger will be at least 200-300nm late. More consistent and accurate triggering may be achieved in this case by using a simple “pulse” trigger. In particular, if velocity control is active and the stage is commanded to travel at a constant velocity, it is valid to assume that velocity control will maintain this velocity with high accuracy.

14.6.4 Trigger events at start or end of sample period

When the sample period is extended so that each sampled data point covers multiple 50kHz update cycles, the user can select for each point whether the trigger event should occur at the start or end of the sample period using the command `function.waveform.trigger-at-sample-end` (see 14.6.9). No provision is made for triggers during the sample period; if a more accurate trigger is required then the sample period must be reduced.

For `pulse` trigger event types, the trigger output will be pulsed at this time.

For level-based trigger events where the commanded position is set to ramp linearly to the next point during this sample period (see 14.2.2), the trigger event is prepared at this time, and also the threshold for the level-based trigger is set to the commanded position at the start or end of the sample period ramp as required.

There is generally little difference between firing at the end of one sample or the start of the next sample. However it may be significant if the next sample is set to step to the new position (see 14.2.2). It is also relevant when triggering on the last sample of the waveform, of course.

Note that this is only relevant for longer sample periods. At the default sample rate of 50kHz, trigger events are updated sample-accurately.

14.6.5 Trigger output pulse duration

To ensure the trigger output pulse can be received correctly by external equipment, the duration of the trigger output pulse can be set.

14.6.6 Factors to consider with trigger timings

It is possible that the conditions for a trigger may not have been met before a second trigger event is set later in the waveform. This typically happens if the waveform does not allow enough time for the stage to reach a commanded position. In that case the controller discards the first trigger event and switches to using the next trigger event. This is usually undesirable since it generally leads to missed data captures.

If the trigger output pulse time is set too long and two or more trigger points are close together, there is a risk that a second trigger will occur whilst the trigger output is already active. In that case the trigger output will be retriggered and will remain active for the specified time from the second trigger, effectively giving a single longer pulse. This is usually undesirable for external equipment since it generally leads to missed data captures or incorrect operation.

The user should attempt to construct the waveform so that these issues cannot occur. It is also recommended that external equipment or data acquisition/processing should be able to detect missed triggers and take appropriate action as required.

14.6.7 Manual control of trigger output

The user may wish to manually command a trigger output pulse, instead of triggering as part of function playback. A command is provided to allow this.

This is particularly useful when testing that other equipment is being correctly triggered by the controller. It also allows the trigger output to be used in applications where function playback is not used to control stage movement.

14.6.8 function.waveform.trigger-out-event.get/set**14.6.8.1 Description**

Function playback waveform trigger output event

14.6.8.2 Command

```
function.waveform.trigger-out-event.get <channel> <index>
function.waveform.trigger-out-event.set <channel> <index> <value>
```

14.6.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point

14.6.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point
value	String	Enumeration “trigger-output-event-type”			Trigger output event See 14.6.215.6 for available trigger event types

14.6.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			Trigger output event See 14.6.2 for available trigger event types

14.6.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Sample index must be within specified range
Value out of range	Value must be as specified

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.6.8.7 Minimum security level

function.waveform.trigger-out-event.get	No security required
function.waveform.trigger-out-event.set	User level security required

14.6.8.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

14.6.9 function.waveform.trigger-at-sample-end.get/set**14.6.9.1 Description**

Function playback waveform trigger output event set at start (false) or end (true) of sample period

14.6.9.2 Command

```
function.waveform.trigger-at-sample-end.get <channel> <index>
function.waveform.trigger-at-sample-end.set <channel> <index> <value>
```

14.6.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point

14.6.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
index	32-bit unsigned integer	samples	0	See 14.2	Sample index for data point
value	32-bit unsigned integer	Boolean	0	1	Function playback waveform trigger output event set at start (false) or end (true) of sample period

14.6.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Function playback waveform trigger output event set at start (false) or end (true) of sample period

14.6.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Sample index must be within specified range

Error return “errcode” string reported	Description
Value out of range	Value must be as specified
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

14.6.9.7 Minimum security level

function.waveform.trigger-at-sample-end.get	No security required
function.waveform.trigger-at-sample-end.set	User level security required

14.6.9.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

14.6.10 function.trigger-output.pulse-time.get/set**14.6.10.1 Description**

Trigger output pulse length

14.6.10.2 Command

```
function.trigger-output.pulse-time.get <channel>
```

```
function.trigger-output.pulse-time.set <channel> <value>
```

14.6.10.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

14.6.10.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Function playback trigger output pulse length Setting this to exactly zero disables trigger output pulses

14.6.10.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Function playback trigger output pulse length Setting this to exactly zero disables trigger output pulses

14.6.10.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Pulse length must be within specified range (set only)

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

14.6.10.7 Minimum security level

function.trigger-output.pulse-time.get	No security required
function.trigger-output.pulse-time.set	User level security required

14.6.10.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

14.6.11 function.trigger-output.trigger**14.6.11.1 Description**

Manually command pulse on trigger output

14.6.11.2 Command

function.trigger-output.trigger <channel>

14.6.11.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

14.6.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Always returns 1

14.6.11.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

14.6.11.6 Minimum security level

function.trigger-output.trigger	User level security required
---------------------------------	------------------------------

14.6.11.7 Supported in

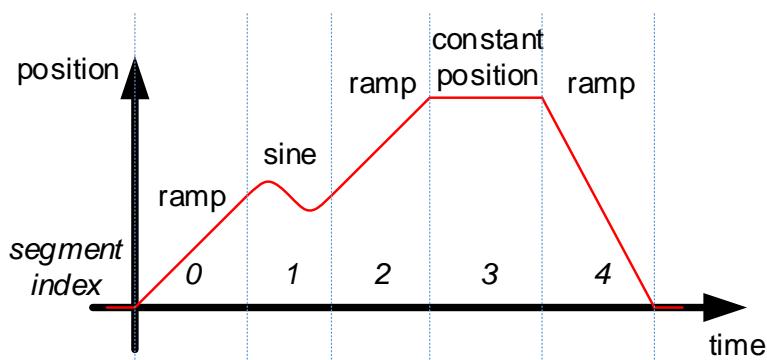
Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15 Waveform generator

15.1 Overview

Entering waveforms for sample-based function playback (see section 14) can be difficult, especially for more complex waveform such as S-curve accelerations. The waveform generator provides an alternative way to create waveforms for function playback, allowing complex waveforms to be constructed with relative ease in a similar way to third-party motion controllers. This is normally preferable to sample-based waveform entry.

For the waveform generator, waveforms are constructed from “segments”, where each segment of a waveform carries out some action. There are numerous different segment types, covering ramping, stepping to a position, various acceleration and deceleration curves, sine waves, and so on. For example, a waveform which ramps to a position, runs one cycle of a sine wave, ramps to a second position, holds position there for a fixed time and then ramps back to the start might be formed of 5 waveform segments with appropriate configurations.



15.2 specifies the commands used to configure waveform segments. 15.3 describes the segments types available and the parameters to configure each type. To clarify the waveform design process, 15.4 provides an example of how this can be used in a realistic application.

Part or all of the waveform can also be set up to run repeatedly, as described in 15.5. This is particularly useful for applications such as raster scanning.

In the user's system, external electronics may be required to be triggered during function playback, for example where a camera must be triggered at successive points whilst moving focus through a specimen. Each stage channel has a dedicated digital trigger output which may be used for this. 15.6 describes the trigger output configuration.

Once segments have been entered, the waveform generator must check that segment configurations are valid, and construct the waveform as a series of sampled data points which can be played by the controller. 15.7 describes how to generate the waveform from the segments entered.

All the normal function playback features apply once the waveform has been generated. The waveform points may be read back (or even edited) using the standard sample-based commands from function playback, as described in 14.2; and waveform playback may be started or stopped by the PC or by external electronics using digital trigger inputs, as described in 14.4 and 14.5.

15.2 Waveform generator segment configuration commands

Each segment type has a number of configuration parameters relating to that segment type. Some will be common to all (for example setting the starting position), and some will be specific to the segment type (for example setting the frequency for a sine wave). In general, all segment types which carry out a common operation (for example an S-curve acceleration) will use the same parameters, so that it is possible to easily compare the results of using similar segment types.

For each segment, the user has the option of whether to continue from the previous segment's final commanded position (and commanded velocity for some segment types), making it easy to construct smooth waveforms, or whether to step to a new position. 15.3 specifies the segment types available, and the parameters available for each. A waveform may have up to 1000 segments.

As can be seen in 8.2.2, the function playback command is summed with all other command sources. This allows other complex behaviour to be set up which combines pre-programmed waveforms with other position commands. For example, a fine auto-focus could be implemented by programming a triangular waveform which automatically sweeps the stage over a $\pm 100\text{nm}$ offset from the current commanded position until the correct focus point is found.

Since the function playback command is summed with all other command sources, it is set to zero when function playback is not running. The first segment should therefore always start at zero, or may step from zero to another position if a smooth start is not required. A controlled end to the waveform is often required, in which case the last segment should also return to zero via a ramp or other smooth transition. If this is unimportant, the waveform may finish at any value and the controller will step back to the original position as the function playback command steps back to zero.

If the waveform is required to finish at a different position and hold that position after function playback has completed, it is possible to set a non-zero final position and set the waveform to "soft" stop at the end. In this case the digital position command is set to the combined position command at the end of the waveform (see the diagram in 8.2.2), so that the absolute commanded position holds its value when the function stops. See also 14.4 for details of commanding a "hard" or "soft" stop whilst the function is running, which behaves identically.

The last segment may also finish at a different position if the waveform is set to repeat forever (see 15.5), in which case any position after it stops does not need to be considered.

15.2.1 function.waveform-generator.clear

15.2.1.1 Description

Clear waveform generator settings

15.2.1.2 Command

```
function.waveform-generator.clear <channel>
```

15.2.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.2.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	32-bit unsigned integer				Always returns 1 if clear succeeded

15.2.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

15.2.1.6 Minimum security level

function.waveform-generator.clear	User level security required
-----------------------------------	------------------------------

15.2.1.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.2.2 function.waveform-generator.segment.type.get/set**15.2.2.1 Description**

Function type for waveform generator segment

15.2.2.2 Command

```
function.waveform-generator.segment.type.get <channel> <segment>
```

```
function.waveform-generator.segment.type.set <channel> <segment>
<type>
```

15.2.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.2.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
type	String	Enumeration “waveform-type”			Function type for segment See 15.3 for available values

15.2.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
type	String	Enumeration “waveform-type”			Function type for segment See 15.3 for available values

15.2.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range

Error return “errcode” string reported	Description
Value out of range	Segment type is incorrect (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.2.2.7 Minimum security level

function.waveform-generator.segment.type.get	No security required
function.waveform-generator.segment.type.set	User level security required

15.2.2.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.2.3 function.waveform-generator.segment.parameter.get/set**15.2.3.1 Description**

Function parameter for waveform generator segment

15.2.3.2 Command

```
function.waveform-generator.segment.parameter.get <channel> <segment>
```

```
function.waveform-generator.segment.parameter.set <channel> <segment>
<value>
```

15.2.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
parameter	32-bit unsigned integer		0	7	Parameter index See 15.3 for parameters for each function type, and the index for each parameter

15.2.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
parameter	32-bit unsigned integer		0	7	Parameter index See 15.3 for parameters for each function type, and the index for each parameter
value	32-bit floating-point				Function parameter for segment See 15.3 for details of each segment type's parameters

15.2.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point				Function parameter for segment

Name	Type	Units	Minimum	Maximum	Description
					See 15.3 for details of each segment type's parameters

15.2.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index or parameter index must be within specified range
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.2.3.7 Minimum security level

function.waveform-generator.segment.parameter.get	No security required
function.waveform-generator.segment.parameter.set	User level security required

15.2.3.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.2.4 function.waveform-generator.segment.continue-position-velocity.get/set**15.2.4.1 Description**

Continue previous position and velocity for waveform generator segment.

15.2.4.2 Command

```
function.waveform-generator.segment.continue-position-velocity.get  
<channel> <segment>
```

```
function.waveform-generator.segment.continue-position-velocity.set  
<channel> <segment> <continue-position> <continue-velocity>
```

15.2.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.2.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
segment	32-bit unsigned integer		0	999	Segment index
continue-position	32-bit unsigned integer	Boolean	0	1	If set to 1, start position for this segment is taken from the end position for the previous segment, and start position function parameter for function type does not need to be provided. If set to 0, start position for this segment is set by the relevant function parameter for this segment. See 15.3 for details of function types and function parameters.
continue-velocity	32-bit unsigned integer	Boolean	0	1	If set to 1, and segment has a function parameter setting start velocity, then start velocity for this segment is taken from the end velocity for the previous segment.

Name	Type	Units	Minimum	Maximum	Description
					<p>This may only be set if continue-position is also set to 1.</p> <p>If set to 0, start velocity for this segment is set explicitly by the relevant function parameter for this segment, if the segment type uses it.</p> <p>If function type does not have a function parameter to set start position, then this is unused.</p> <p>See 15.3 for details of function types and function parameters.</p>

15.2.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
continue-position	32-bit unsigned integer	Boolean	0	1	<p>If set to 1, start position for this segment is taken from the end position for the previous segment, and start position function parameter for function type does not need to be provided.</p> <p>If set to 0, start position for this segment is set by the relevant function parameter for this segment.</p> <p>See 15.3 for details of function types and function parameters.</p>
continue-velocity	32-bit unsigned integer	Boolean	0	1	<p>If set to 1, and segment has a function parameter setting start velocity, then start velocity for this segment is taken from the end velocity for the previous segment.</p> <p>This may only be set if continue-position is also set to 1.</p>

Name	Type	Units	Minimum	Maximum	Description
					If set to 0, start velocity for this segment is set explicitly by the relevant function parameter for this segment, if the segment type uses it. If function type does not have a function parameter to set start position, then this is unused. See 15.3 for details of function types and function parameters.

15.2.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Value out of range	Continue-velocity must not be set to 1 if continue-position is not set to 1 (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.2.4.7 Minimum security level

function.waveform-generator.segment.continue-position-velocity.get	No security required
function.waveform-generator.segment.continue-position-velocity.set	User level security required

15.2.4.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.2.5 function.waveform-generator.count.get/set

15.2.5.1 Description

Number of waveform generator segments

15.2.5.2 Command

```
function.waveform-generator.count.get <channel>
```

```
function.waveform-generator.count.set <channel> <value>
```

15.2.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.2.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer		1	1000	Number of waveform generator segments

15.2.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		1	1000	Number of waveform generator segments

15.2.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Number of segments must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.2.5.7 Minimum security level

function.waveform-generator.count.get	No security required
function.waveform-generator.count.set	User level security required

15.2.5.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.2.6 function.waveform-generator.soft-stop-at-end.get/set

15.2.6.1 Description

Waveform soft-stops at end (true) or steps to zero (false)

15.2.6.2 Command

```
function.waveform-generator.soft-stop-at-end.get <channel>
```

```
function.waveform-generator.soft-stop-at-end.set <channel> <value>
```

15.2.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.2.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer	Boolean	0	1	If set to 1, waveform soft-stops at end. If set to 0, waveform stops normally at end.

15.2.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	If set to 1, waveform soft-stops at end. If set to 0, waveform stops normally at end.

15.2.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.2.6.7 Minimum security level

function.waveform-generator.soft-stop-at-end.get	No security required
function.waveform-generator.soft-stop-at-end.set	User level security required

15.2.6.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.3 Waveform generator segment function types

The waveform generator has a variety of segment function types, set by `function.waveform-generator.segment.type` (see 15.2.2). Every function type is configured with different function parameters depending on the waveform to be generated, set by `function.waveform-generator.segment.parameter` (see 15.2.3).

The function types available are intended to allow all frequently-used movement profiles to be used. The functions available can be grouped as:

- Basic functions: Constant position, constant-velocity ramps, and sine-waves.
- Acceleration and deceleration: Various degrees of S-curve implemented with linear characteristics or with sine-wave sections.
- Steps: Movement between positions using S-curve acceleration/deceleration profiles to reduce system excitation.
- Staircases: Repeated regular steps, which may use S-curve steps to reduce system excitation.

Additional functions may be added in future as applications are encountered which require them.

15.3.1 Function configuration

The majority of function types have three variants of the same function.

- The “duration” type carries out the function (for example, a constant-velocity ramp) over a fixed time period. It starts at the start position and continues for the specified time. The distance travelled depends on the time taken, and other parameters for the segment.
- The “position” type carries out the function from the start position to a fixed end position. The time taken depends on the distance and other parameters for the segment.
- The “distance” type carries out the function from the start position to some distance relative to the start position. This is important for more complex waveforms where the user may not be able to easily calculate the distance travelled during a previous function segment. As with the “position” type, the time taken depends on the distance and other parameters for the segment.

These make it relatively easy to construct a waveform with fixed intervals of time or distance, depending on the application requirements. All three variants will not always be applicable for every function, but it is common to see two or three variants of the same function. Other variants may also exist, where other parameters may be used to control the function.

Parameters for functions must be set to valid values, which will depend on the function. Durations must not be zero or negative, for example; and similarly distances must generally not be zero. The possible errors are listed for each function type, with errors reported using the standard `function.waveform-generator.failure-cause.get` command (see 15.7.11).

Note that many functions perform similar roles. For example, the various S-curve acceleration functions all perform the same basic requirement of accelerating to a velocity within the specified time/distance, and vary only in how they achieve this. The same parameter configuration is used for all related functions, making it easy to compare the effects of different functions simply by changing the function type, without having to also change the parameter configuration.

15.3.2 Rounding errors with large sample periods

As described in 15.7, durations are rounded to fit the waveform sample period. For smaller sample periods this is normally not significant. However if a relatively large sample period is set, rounding errors may become significant. This is particularly relevant for non-linear function types such as sine waves or S-curve accelerations.

- “Duration” function types will prioritise maintaining the specified velocities or accelerations. The stage may travel a slightly longer or shorter distance than expected during the segment, depending on rounding.
- “Position” and “distance” function types will prioritise hitting the specified end position. The velocity or acceleration during the segment may be slightly greater or less than expected or specified, depending on rounding.

Each function type specifies how it is affected by rounding. If the waveform sample period does have an adverse effect on the waveform, it is normally straightforward to use a shorter sample period. If this is not possible, consider using a different function type where the waveform generator will prioritise the correct parameters for your application.

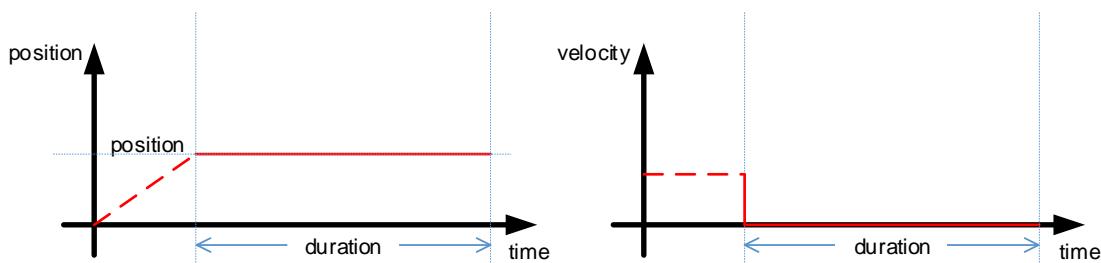
15.3.3 Function types supported in firmware versions

New function types will continue to be added as features are added to the controller. The “Supported in” section states which versions of controller firmware and interface DLL are required to use each function type.

15.3.4 Basic function types

15.3.4.1 Constant position

Segment function type	Description
constant-position	Hold a fixed command position for the specified time.



This provides basic constant position behaviour, holding position for a set time.

A square wave can be constructed from two “constant-position” segments at the relevant positions.

15.3.4.1.1 constant-position

15.3.4.1.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Position	pm			Position to hold. If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Duration	s	> 0		Time to hold fixed position

When preparing the waveform, duration is rounded to the nearest sample period (see 15.7).

15.3.4.1.1.2 Possible errors

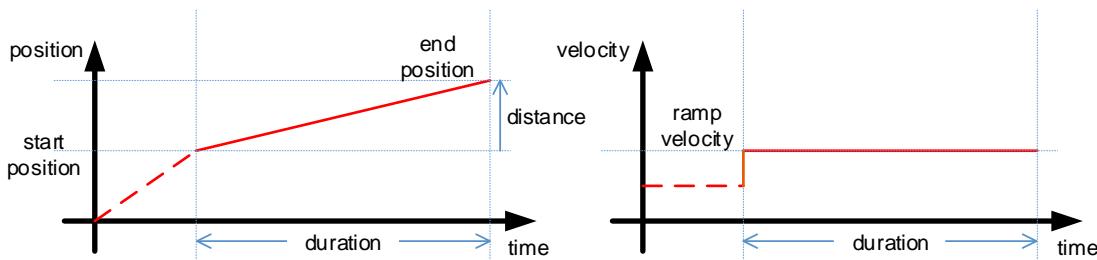
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.4.1.1.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.3.4.2 Constant velocity

Segment function type	Description
constant-velocity-duration	Ramp command position at a constant velocity for the specified time.
constant-velocity-position	Ramp command position at a constant velocity from start position to end position.
constant-velocity-distance	Ramp command position at a constant velocity from start position for a relative distance.



This provides a simple ramp. The user has the choice of whether to ramp over a fixed time or a fixed distance.

A sawtooth wave can be constructed from successive “constant-velocity” segments, stepping back to the initial position (i.e. “continue-position” is not set).

A triangle wave can be constructed from successive “constant-velocity” segments, ramping up and then ramping down.

15.3.4.2.1 constant-velocity-duration

15.3.4.2.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Ramp velocity	nm/ms	Non-zero		Velocity for ramp. If “continue-velocity” is set, the end velocity for the previous segment is used, and this parameter is unused.
2	Duration	s	> 0		Duration of ramp

When preparing the waveform, duration is rounded to the nearest sample period (see 15.7).

If duration is not an exact multiple of the sample period and must be rounded, ramping at the specified velocity for a slightly different duration will cause the end position to be slightly different to that expected.

15.3.4.2.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
velocity-zero	For function types which specify a velocity, the velocity must not be zero.

15.3.4.2.1.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

NOTE: In 6.4.x and 6.5.x firmware, the ramp velocity parameter units were pm/s. For 6.6.1 onwards, the velocity units were changed to nm/ms as stated.

15.3.4.2.2 constant-velocity-position

15.3.4.2.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Ramp velocity	nm/ms	Non-zero		Velocity for ramp. If “continue-velocity” is set, the end velocity for the previous segment is used, and this parameter is unused.
2	End position	pm			End position for ramp

When preparing the waveform, the ramp duration for the specified velocity is rounded to the nearest sample period (see 15.7).

If the ramp duration is not an exact multiple of the sample period and must be rounded, ramping over the specified distance for a slightly different duration will cause the velocity to be slightly different to that specified. If it is critical that the velocity is exactly correct, then the “constant-velocity-duration” segment type (see 15.3.4.2.1) should be used instead, maintaining the specified velocity at the cost of a slight change to end position. Alternatively the sample period

(see 15.7) may be adjusted so that the ramp duration becomes an exact fit, or a close enough fit for the velocity tolerance.

15.3.4.2.2.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
velocity-zero	For function types which specify a velocity, the velocity must not be zero.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.4.2.2.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

NOTE: In 6.4.x and 6.5.x firmware, the ramp velocity parameter units were pm/s. For 6.6.1 onwards, the velocity units were changed to nm/ms as stated.

15.3.4.2.3 constant-velocity-distance

15.3.4.2.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Ramp velocity	nm/ms	Non-zero		Velocity for ramp. If “continue-velocity” is set, the end velocity for the previous segment is used, and this parameter is unused.
2	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the ramp duration for the specified velocity is rounded to the nearest sample period (see 15.7).

If the ramp duration is not an exact multiple of the sample period and must be rounded, ramping over the specified distance for a slightly different duration will cause the velocity to be slightly different to that specified. If it is critical that the velocity is exactly correct, then the “constant-

“velocity-duration” segment type (see 15.3.4.2.1) should be used instead, maintaining the specified velocity at the cost of a slight change to end position. Alternatively the sample period (see 15.7) may be adjusted so that the ramp duration becomes an exact fit, or a close enough fit for the velocity tolerance.

15.3.4.2.3.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
velocity-zero	For function types which specify a velocity, the velocity must not be zero.

15.3.4.2.3.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

NOTE: In 6.4.x and 6.5.x firmware, the ramp velocity parameter units were pm/s. For 6.6.1 onwards, the velocity units were changed to nm/ms as stated.

15.3.4.2.4 constant-velocity-position-duration

15.3.4.2.4.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			End position for ramp
2	Duration	s	> 0		Duration of ramp

When preparing the waveform, the ramp duration is rounded to the nearest sample period (see 15.7).

If the ramp duration is not an exact multiple of the sample period and must be rounded, ramping over the specified distance for a slightly different duration will cause the velocity to be slightly different to that expected. If it is critical that the velocity is exactly correct, then the “constant-velocity-duration” segment type (see 15.3.4.2.1) should be used instead, maintaining the specified velocity at the cost of a slight change to end position. Alternatively the sample period (see 15.7) may be adjusted so that the ramp duration becomes an exact fit, or a close enough fit for the velocity tolerance.

15.3.4.2.4.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.4.2.4.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.4.2.5 constant-velocity-distance-duration**15.3.4.2.5.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm	Non-zero		Relative distance of end position from start position.
2	Duration	s	> 0		Duration of ramp

When preparing the waveform, the ramp duration is rounded to the nearest sample period (see 15.7).

If the ramp duration is not an exact multiple of the sample period and must be rounded, ramping over the specified distance for a slightly different duration will cause the velocity to be slightly different to that expected. If it is critical that the velocity is exactly correct, then the “constant-velocity-duration” segment type (see 15.3.4.2.1) should be used instead, maintaining the specified velocity at the cost of a slight change to end position. Alternatively the sample period (see 15.7) may be adjusted so that the ramp duration becomes an exact fit, or a close enough fit for the velocity tolerance.

15.3.4.2.5.2 Possible errors

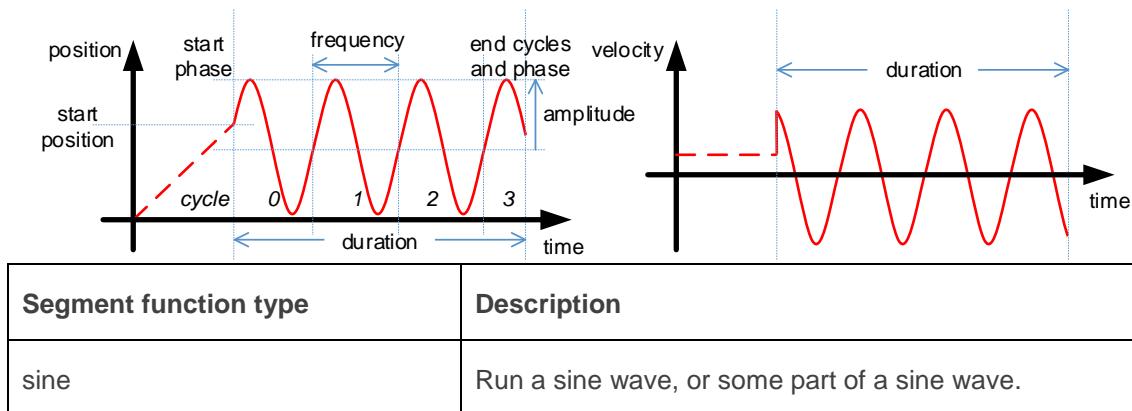
Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.4.2.5.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.4.3 Sine wave



A sine wave can be run between any phase positions in a cycle, and for multiple cycles. The phase is expressed as a fraction of a complete cycle. For example, starting at a phase of 90 degrees requires a start phase value of 0.25; and finishing at a phase of 270 degrees during the 4th cycle requires an end phase value of 4.75.

The start phase value must be greater than or equal to zero, and less than 1 (360 degrees). Negative values for phase/cycles are not permitted. For example, running part of a sine wave from -45 degrees to +45 degrees must be represented by running from a start phase of 0.875 (315 degrees) to an end phase of 1.125 (405 degrees). The end phase value must of course be greater than the start phase.

The “centre” of the sine wave is calculated based on the start phase and on the start position at that phase. If the start phase is zero, for example, then the sine wave is starting at the centre and so will be centred around the start position. If the start phase is 0.25 (90 degrees) instead, then the sine wave is starting at a peak and so will be centred around the start position minus the amplitude.

15.3.4.3.1 sine

15.3.4.3.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start phase	Fraction of cycle	0	< 1	Starting phase in cycle, scaled to a fraction of a cycle (0 = 0 degrees, 0.5 = 180 degrees, 1 = 360 degrees)
2	End cycles and phase	Fraction of cycle	0		End cycles and end phase, scaled to a fraction of a cycle (0 = 0 degrees, 0.5 = 180

Index	Parameter	Units	Minimum	Maximum	Description
					degrees, 1 = 360 degrees, 2 = 720 degrees i.e. 2 cycles)
3	Amplitude	pm	0		Sine wave amplitude, from centre to peak
4	Frequency	Hz	> 0		Sine wave frequency

When preparing the waveform, the duration of the sine wave is rounded to the nearest sample period (see 15.7).

If the sine wave duration is not an exact multiple of the sample period and must be rounded, the frequency will be adjusted slightly so that the rounded duration fits the specified start phase, end phase and cycles.

15.3.4.3.1.2 Possible errors

Error reported	Description
start-not-first-cycle	For periodic functions such as sine waves, the start phase must be within the first cycle. Since phase values are set as a fraction of a cycle, the start phase must be between 0 and 1.
end-before-start	For periodic functions such as sine waves, the end must be set after the start.
amplitude-negative	For periodic functions such as sine waves, the amplitude is set as a magnitude and must not be negative.
frequency-zero	For periodic functions such as sine waves, the frequency cannot be zero (or negative).
frequency-above-half-sample-rate	For periodic functions such as sine waves, the frequency cannot be greater than half the waveform sample rate (the Nyquist limit). Either reduce the sample period to allow waveforms with higher frequencies to be constructed, or change the frequency required.

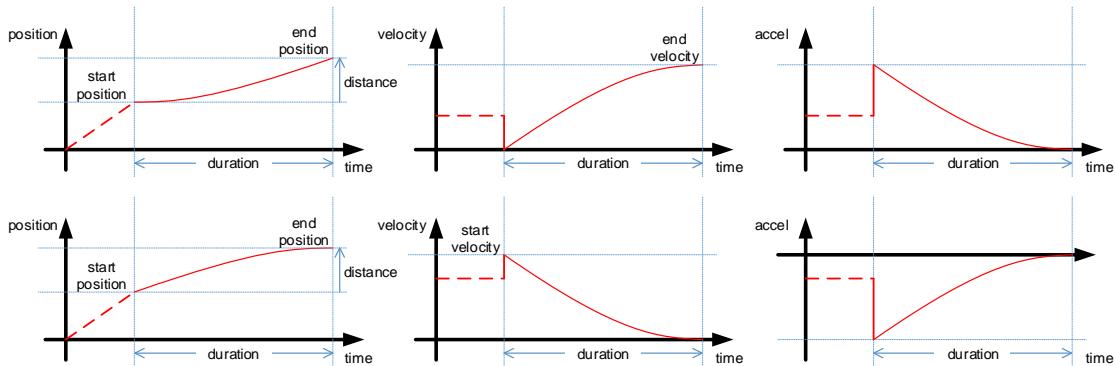
15.3.4.3.1.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.3.5 Sine-wave profile S-curves for acceleration and deceleration

15.3.5.1 Quarter-sine acceleration and deceleration profiles

Segment function type	Description
quarter-sine-accel-duration	Using a quarter-sine acceleration profile, accelerate from rest at to specified velocity in specified time.
quarter-sine-accel-position	Using a quarter-sine acceleration profile, accelerate from rest at start position to specified velocity at end position.
quarter-sine-accel-distance	Using a quarter-sine acceleration profile, accelerate from rest at start position to specified velocity over relative distance.
quarter-sine-decel-duration	Using a quarter-sine deceleration profile, decelerate from specified velocity to rest in specified time.
quarter-sine-decel-position	Using a quarter-sine deceleration profile, decelerate from specified velocity to rest at end position .
quarter-sine-decel-distance	Using a quarter-sine deceleration profile, decelerate from specified velocity to rest over relative distance.



A common technique for minimising resonance during acceleration and deceleration is for the trajectory to follow a portion of a sine wave.

As can be seen from the above diagrams, the use of a quarter-sine-wave trajectory ensures that the final acceleration/deceleration is zero, giving a smooth transition into a constant-velocity ramp (for acceleration) or a steady-state position (for deceleration). Note however that there is an initial step-change in acceleration, with maximum acceleration/deceleration applied at the start of the segment. For a mechanically-stiff system this can give a more rapid acceleration/deceleration, with any resonances excited by the step-change in acceleration damped by the system. For a less stiff system, it may be more appropriate to use an S-curve acceleration profile which does not excite system resonances as much.

The acceleration types always start from rest, so will ignore the "continue-velocity" setting. The deceleration types may use the "continue-velocity" setting, and will always finish with zero velocity.

Acceleration and deceleration require some time to complete, and some starting/stopping distance. The user has the choice of whether to fix the time taken or the distance travelled, as required for the user's waveform.

15.3.5.1.1 quarter-sine-accel-duration

15.3.5.1.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End velocity	nm/ms	Non-zero		Final velocity after acceleration
2	Duration	s	> 0	Duration of acceleration	

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.5.1.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.

15.3.5.1.1.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.5.1.2 quarter-sine-accel-position

15.3.5.1.2.1 Function parameters

Index	Parameter	Units	Minimu m	Maximu m	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
1	End position	pm			Final position after acceleration
2	End velocity	nm/ms	Non-zero		Final velocity after acceleration

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.5.1.2.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.1.2.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.5.1.3 quarter-sine-accel-distance

15.3.5.1.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End velocity	nm/ms	Non-zero		Final velocity after acceleration
2	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.5.1.3.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.1.3.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.5.1.4 quarter-sine-decel-duration

15.3.5.1.4.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms	Non-zero		If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	Duration	s	> 0		Duration of deceleration

When preparing the waveform, the duration of the deceleration is rounded to the nearest sample period (see 15.7).

15.3.5.1.4.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

Error reported	Description
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.

15.3.5.1.4.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.5.1.5 quarter-sine-decel-position

15.3.5.1.5.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms	Non-zero		If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End position	pm			Final position after deceleration

When preparing the waveform, the duration of the deceleration is rounded to the nearest sample period (see 15.7).

15.3.5.1.5.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.1.5.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.5.1.6 quarter-sine-decel-distance**15.3.5.1.6.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms	Non-zero		If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the deceleration is rounded to the nearest sample period (see 15.7).

15.3.5.1.6.2 Possible errors

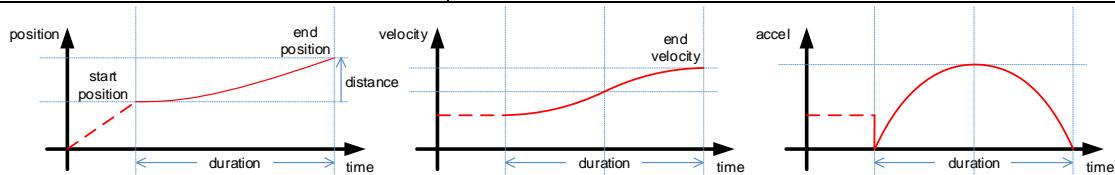
Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.1.6.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.5.2 S-curve acceleration to velocity – half-sine profile

Segment function type	Description
accel-to-velocity-s-curve-half-sine-accel-duration	Using a half-sine acceleration profile, accelerate or decelerate from starting velocity to desired velocity in specified time.
accel-to-velocity-s-curve-half-sine-accel-position	Using a half-sine acceleration profile, accelerate or decelerate from starting velocity to desired velocity at end position.
accel-to-velocity-s-curve-half-sine-accel-distance	Using a half-sine acceleration profile, accelerate or decelerate from starting velocity to desired velocity over relative distance.



Building on the quarter-sine acceleration and deceleration profiles, the half-sine acceleration profile allows acceleration or deceleration from the initial velocity to a new velocity. This may start or finish at rest, but (unlike the quarter-sine profiles) does not require it.

The half-sine profile ensures the segment starts and finishes with zero acceleration. This excites resonances less than a quarter-sine profile, which has an initial step-change in acceleration. However the profile requires slightly greater peak acceleration to reach a target position in a specified time.

There is still an instantaneous jerk at the start and finish of the segment as acceleration changes. If this excites resonances in the system, the full-sine profile may be more appropriate.

15.3.5.2.1 accel-to-velocity-s-curve-half-sine-accel-duration

15.3.5.2.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration

Index	Parameter	Units	Minimum	Maximum	Description
3	Duration	s	> 0		Duration of acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity.

15.3.5.2.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.

15.3.5.2.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.5.2.2 accel-to-velocity-s-curve-half-sine-accel-position

15.3.5.2.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
2	End position	pm			Final position after acceleration
3	End velocity	nm/ms			Final velocity after acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.5.2.2.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.2.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.5.2.3 accel-to-velocity-s-curve-half-sine-accel-distance

15.3.5.2.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous

Index	Parameter	Units	Minimum	Maximum	Description
					segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.5.2.3.2 Possible errors

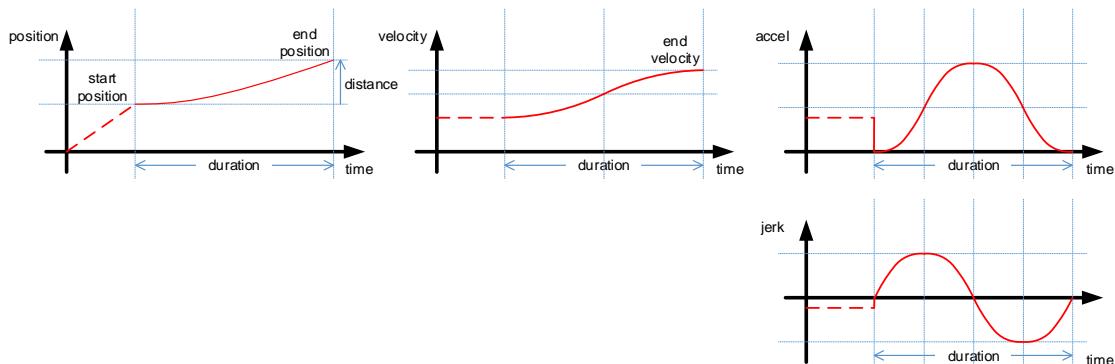
Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.2.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.5.3 S-curve acceleration to velocity – full-sine profile

Segment function type	Description
accel-to-velocity-s-curve-full-sine-accel-duration	Using a full-sine acceleration profile, accelerate or decelerate from starting velocity to desired velocity in specified time.
accel-to-velocity-s-curve-full-sine-accel-position	Using a full-sine acceleration profile, accelerate or decelerate from starting velocity to desired velocity at end position.
accel-to-velocity-s-curve-full-sine-accel-distance	Using a full-sine acceleration profile, accelerate or decelerate from starting velocity to desired velocity over relative distance.



Like the half-sine acceleration, the full-sine acceleration profile allows acceleration or deceleration from the initial velocity to a new velocity. This may start or finish at rest, but (unlike the quarter-sine profiles) does not require it.

The full-sine profile ensures the segment starts and finishes with zero acceleration and with zero jerk. This should reduce system excitation to a minimum. However the profile requires slightly greater peak acceleration to reach a target position in a specified time.

15.3.5.3.1 accel-to-velocity-s-curve-full-sine-accel-duration

15.3.5.3.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
2	End velocity	nm/ms			Final velocity after acceleration
3	Duration	s	> 0		Duration of acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity.

15.3.5.3.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.

15.3.5.3.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.5.3.2 accel-to-velocity-s-curve-full-sine-accel-position

15.3.5.3.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous

Index	Parameter	Units	Minimum	Maximum	Description
					segment is held, and this parameter is unused.
2	End position	pm			Final position after acceleration
3	End velocity	nm/ms			Final velocity after acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.5.3.2.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.5.3.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.5.3.3 accel-to-velocity-s-curve-full-sine-accel-distance**15.3.5.3.3.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.5.3.3.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

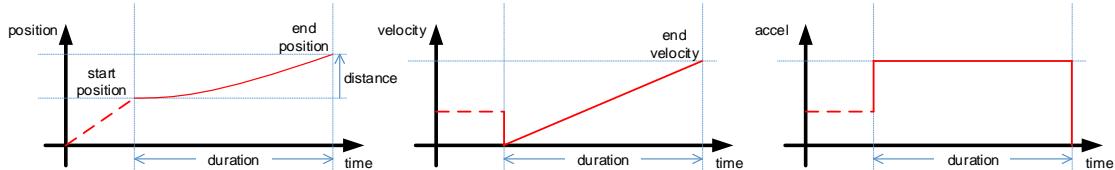
15.3.5.3.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.6 Linear profile S-curves for acceleration and deceleration

15.3.6.1 Constant acceleration to velocity – triangular velocity profile

Segment function type	Description
accel-to-velocity-constant-accel	Accelerate (or decelerate) from current velocity to new velocity with specified acceleration rate.
accel-to-velocity-constant-accel-duration	Accelerate (or decelerate) from current velocity to new velocity in specified time.
accel-to-velocity-constant-accel-position	Accelerate (or decelerate) from current velocity to new velocity at end position.
accel-to-velocity-constant-accel-distance	Accelerate (or decelerate) from current velocity to new velocity to rest over relative distance.



A common acceleration profile for accelerating or decelerating is to apply a constant acceleration/deceleration until the desired change in velocity is achieved. The constant acceleration results in a triangular velocity profile. Acceleration may be specified directly, or the required acceleration may be calculated from the acceleration duration or distance travelled, as required for the user's waveform.

Unlike the quarter-sine acceleration and deceleration functions specified in 15.3.4.3.1.3, this allows changes between non-zero velocities, whereas the quarter-sine functions only handle acceleration from rest and deceleration to rest.

A constant acceleration profile inherently results in step-changes in acceleration at both the start and end of the segment. For a mechanically-stiff system this can give rapid acceleration/deceleration, with any resonances excited by the step-change in acceleration damped by the system. For a less stiff system, it may be more appropriate to use an S-curve acceleration profile which does not excite system resonances as much.

15.3.6.1.1 accel-to-velocity-constant-accel

15.3.6.1.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Acceleration	nm/ms/ms	Non-zero		Acceleration rate

Note that both velocity and acceleration are signed. For positive velocity, positive acceleration equates to increased speed in the direction of travel and negative acceleration equates to decreased speed (braking). For negative velocity, negative acceleration equates to increased speed in the direction of travel and positive acceleration equates to decreased speed (braking).

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity.

15.3.6.1.1.2 Possible errors

Error reported	Description
accel-zero	For function types which specify an acceleration, the acceleration must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-same	For acceleration/deceleration function types, the start and end velocities cannot both be the same.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.

15.3.6.1.1.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.6.1.2 accel-to-velocity-constant-accel-duration

15.3.6.1.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Duration	s	> 0		Duration of acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the expected acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity.

15.3.6.1.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.1.2.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.6.1.3 accel-to-velocity-constant-accel-position**15.3.6.1.3.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End position	pm			Final position after acceleration
3	End velocity	nm/ms			Final velocity after acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the expected acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.6.1.3.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.

Error reported	Description
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.1.3.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.6.1.4 accel-to-velocity-constant-accel-distance

15.3.6.1.4.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the expected acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.6.1.4.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.

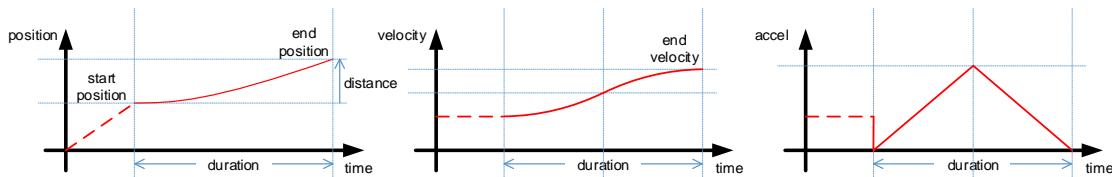
Error reported	Description
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.1.4.3 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

15.3.6.2 S-curve acceleration to velocity – triangular acceleration profile

Segment function type	Description
accel-to-velocity-s-curve-triangular-accel-duration	Using a triangular acceleration profile, accelerate or decelerate from starting velocity to desired velocity in specified time.
accel-to-velocity-s-curve-triangular-accel-position	Using a triangular acceleration profile, accelerate or decelerate from starting velocity to desired velocity at end position.
accel-to-velocity-s-curve-triangular-accel-distance	Using a triangular acceleration profile, accelerate or decelerate from starting velocity to desired velocity over relative distance.



The triangular acceleration profile ensures the segment starts and finishes with zero acceleration. This excites resonances less than the triangular velocity profile, which has step-changes in acceleration at start and end. However the profile requires slightly greater peak acceleration to reach a target position in a specified time compared to the triangular velocity profile.

There is still an instantaneous jerk at the start and finish of the segment as acceleration changes. If this excites resonances in the system, the triangular jerk profile may be more appropriate.

15.3.6.2.1 accel-to-velocity-s-curve-triangular-accel-duration

15.3.6.2.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration

Index	Parameter	Units	Minimum	Maximum	Description
3	Duration	s	> 0		Duration of acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity.

15.3.6.2.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.

15.3.6.2.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.6.2.2 accel-to-velocity-s-curve-triangular-accel-position

15.3.6.2.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
2	End position	pm			Final position after acceleration
3	End velocity	nm/ms			Final velocity after acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.6.2.2.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.2.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.6.2.3 accel-to-velocity-s-curve-triangular-accel-distance

15.3.6.2.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous

Index	Parameter	Units	Minimum	Maximum	Description
					segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.6.2.3.2 Possible errors

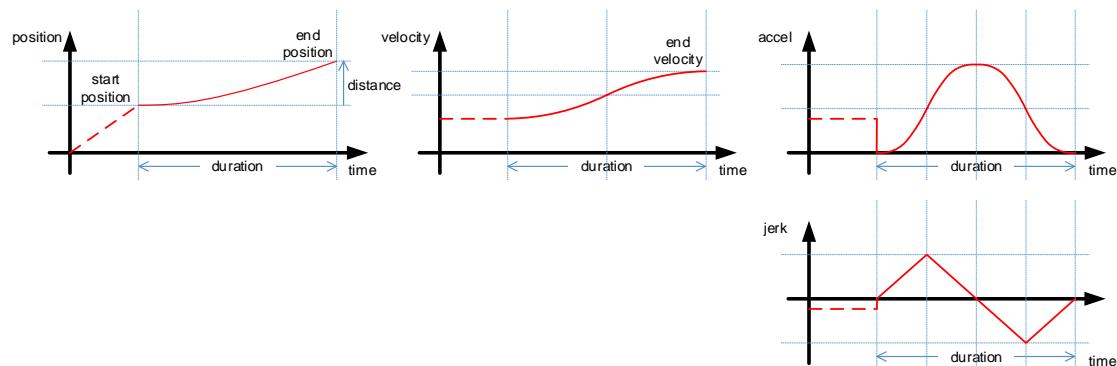
Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.2.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.6.3 S-curve acceleration to velocity – triangular jerk profile

Segment function type	Description
accel-to-velocity-s-curve-triangular-jerk-duration	Using a triangular jerk acceleration profile, accelerate or decelerate from starting velocity to desired velocity in specified time.
accel-to-velocity-s-curve-triangular-jerk-position	Using a triangular jerk acceleration profile, accelerate or decelerate from starting velocity to desired velocity at end position.
accel-to-velocity-s-curve-triangular-jerk-distance	Using a triangular jerk acceleration profile, accelerate or decelerate from starting velocity to desired velocity over relative distance.



The triangular jerk profile ensures the segment starts and finishes with zero acceleration and also with zero jerk. This should reduce system excitation to a minimum, compared to the triangular acceleration profile. However the profile requires slightly greater peak acceleration to reach a target position in a specified time compared to the triangular acceleration profile.

15.3.6.3.1 accel-to-velocity-s-curve-triangular-jerk-duration

15.3.6.3.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration

Index	Parameter	Units	Minimum	Maximum	Description
3	Duration	s	> 0		Duration of acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity.

15.3.6.3.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.

15.3.6.3.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.6.3.2 accel-to-velocity-s-curve-triangular-jerk-position

15.3.6.3.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
2	End position	pm			Final position after acceleration
3	End velocity	nm/ms			Final velocity after acceleration

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.6.3.2.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.3.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.6.3.3 accel-to-velocity-s-curve-triangular-jerk-distance

15.3.6.3.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous

Index	Parameter	Units	Minimum	Maximum	Description
					segment is held, and this parameter is unused.
1	Start velocity	nm/ms			If “continue-velocity” is set, the end velocity for the previous segment is held, and this parameter is unused.
2	End velocity	nm/ms			Final velocity after acceleration
3	Distance	pm	Non-zero		Relative distance of end position from start position.

When preparing the waveform, the duration of the acceleration/deceleration is rounded to the nearest sample period (see 15.7).

If the duration is not an exact multiple of the sample period and must be rounded, the acceleration will be adjusted slightly so that the rounded duration gives the specified end velocity at the specified end position.

15.3.6.3.3.2 Possible errors

Error reported	Description
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.

15.3.6.3.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

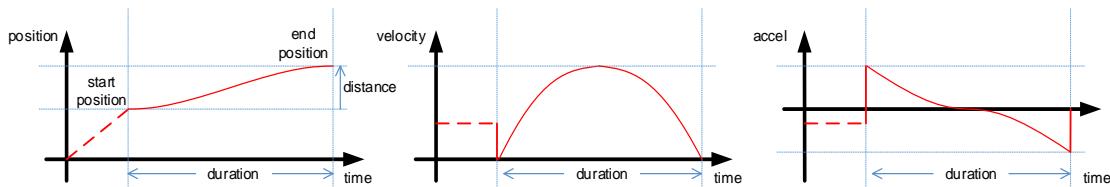
15.3.7 Steps using sine-wave profile S-curves

Using the sine-wave profile S-curves for acceleration and deceleration as described in 15.3.5, the following functions create corresponding steps. The S-curve command can help to reduce excitation of system resonance.

All step functions start and finish at rest (with zero velocity), so will ignore the “continue-velocity” setting. Acceleration and deceleration phases are symmetrical, so maximum velocity occurs at the midpoint of the step. Maximum velocity and acceleration are calculated based on the distance and duration specified.

15.3.7.1 Step with quarter-sine acceleration/deceleration profiles

Segment function type	Description
step-quarter-sine-accel-position	Using quarter-sine acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at end position.
step-quarter-sine-accel-distance	Using quarter-sine acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at relative distance.



Quarter-sine acceleration and deceleration profiles as described in 15.3.5.1 give a smooth step response. Note however that there is an initial step-change in acceleration and a corresponding step-change in deceleration at the end, with maximum acceleration/deceleration applied at the start and end of the segment. For a mechanically-stiff system this can give a more rapid acceleration/deceleration, with any resonances excited by the step-change in acceleration damped by the system. For a less stiff system, it may be more appropriate to use an S-curve acceleration profile which does not excite system resonances as much.

15.3.7.1.1 step-quarter-sine-accel-position

15.3.7.1.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after step
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.7.1.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.7.1.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.7.1.2 step-quarter-sine-accel-distance

15.3.7.1.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm			Relative distance of end position from start position.
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.7.1.2.2 Possible errors

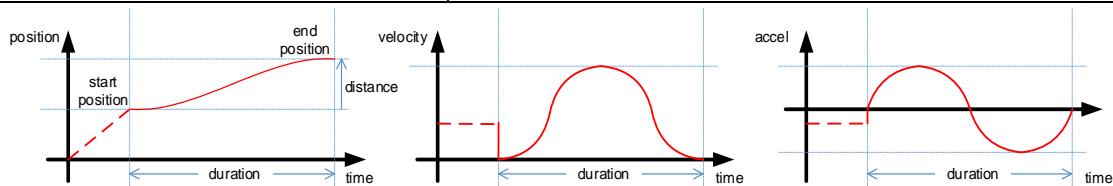
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.7.1.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.7.2 Step with half-sine acceleration/deceleration profiles

Segment function type	Description
step-half-sine-accel-position	Using half-sine acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at end position.
step-half-sine-accel-distance	Using half-sine acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at relative distance.



Half-sine acceleration and deceleration profiles as described in 15.3.5.2 ensure the segment starts and finishes with zero acceleration. This excites resonances less than a quarter-sine profile, which has an initial step-change in acceleration. However the profile requires slightly greater peak acceleration to reach a target position in a specified time.

There is still an instantaneous jerk at the start and finish of the segment as acceleration changes. If this excites resonances in the system, the full-sine profile may be more appropriate.

15.3.7.2.1 step-half-sine-accel-position

15.3.7.2.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after step
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.7.2.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.7.2.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.7.2.2 step-half-sine-accel-distance**15.3.7.2.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm			Relative distance of end position from start position.
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.7.2.2.2 Possible errors

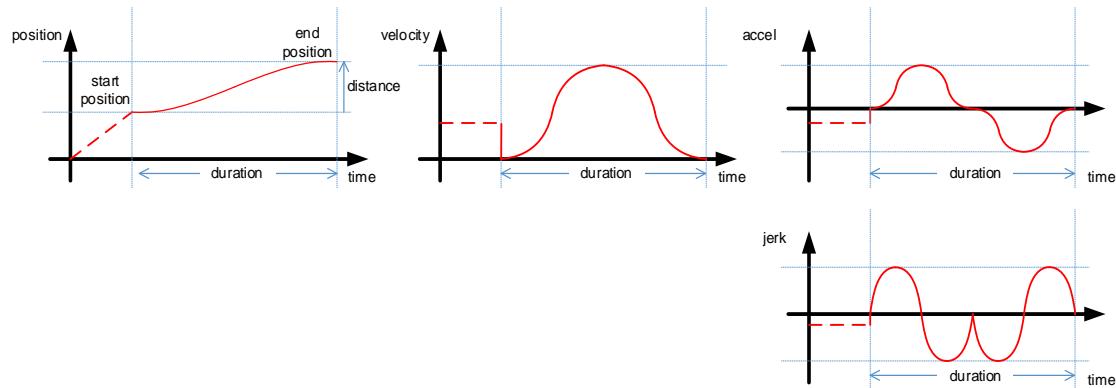
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.7.2.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.7.3 Step with full-sine acceleration/deceleration profiles

Segment function type	Description
step-half-sine-accel-position	Using full-sine acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at end position.
step-half-sine-accel-distance	Using full-sine acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at relative distance.



Full-sine acceleration and deceleration profiles as described in 15.3.5.3 ensure the segment starts and finishes with zero acceleration and zero jerk. This should reduce system excitation to a minimum. However the profile requires slightly greater peak acceleration to reach a target position in a specified time.

15.3.7.3.1 step-full-sine-accel-position

15.3.7.3.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after step
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.7.3.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.7.3.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.7.3.2 step-full-sine-accel-distance**15.3.7.3.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm			Relative distance of end position from start position.
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.7.3.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.7.3.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

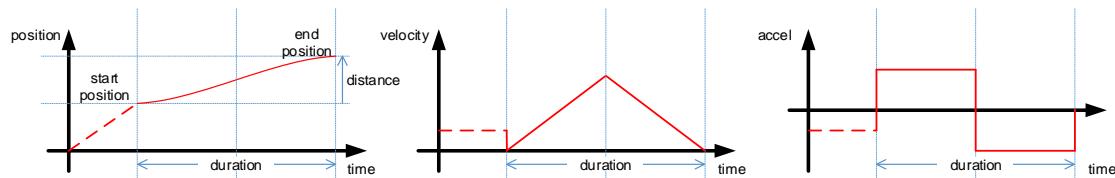
15.3.8 Steps using linear profile S-curves

Using the linear profile S-curves for acceleration and deceleration as described in 15.3.5.3.3.3, the following functions create corresponding steps. The S-curve command can help to reduce excitation of system resonance.

All step functions start and finish at rest (with zero velocity), so will ignore the “continue-velocity” setting. Acceleration and deceleration phases are symmetrical, so maximum velocity occurs at the midpoint of the step. Maximum velocity and acceleration are calculated based on the distance and duration specified.

15.3.8.1 Step with constant acceleration/deceleration – triangular velocity profile

Segment function type	Description
step-triangular-velocity-position	Move from rest at start position to rest at end position in specified time, with constant acceleration and deceleration.
step-triangular-velocity-distance	Move from rest at start position to rest at a relative distance in specified time, with constant acceleration and deceleration.



This uses constant symmetrical acceleration and deceleration to round off the edges of a step, which can reduce system resonances. This is an extremely simple profile, but can give reasonable results where more complex trajectory profiles are not required.

15.3.8.1.1 step-triangular-velocity-position

15.3.8.1.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after step
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.8.1.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.8.1.1.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.3.8.1.2 step-triangular-velocity-distance**15.3.8.1.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm	Non-zero		Relative distance of end position from start position.
2	Duration	s	> 0		Duration of acceleration

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.8.1.2.2 Possible errors

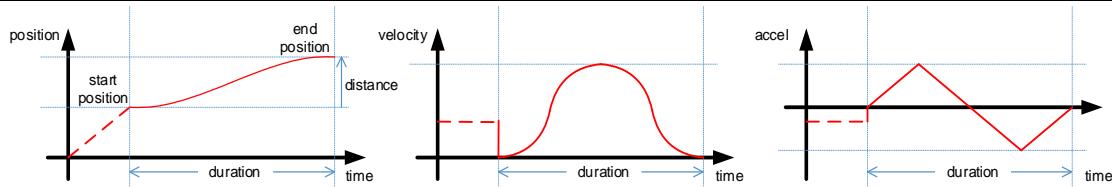
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.8.1.2.3 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.3.8.2 Step with triangular-acceleration acceleration/deceleration profiles

Segment function type	Description
step-triangular-accel-position	Using triangular-acceleration acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at end position.
step- triangular -accel-distance	Using triangular-acceleration acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at relative distance.



Triangular acceleration and deceleration profiles as described in 15.3.6.1.4.3 ensure the segment starts and finishes with zero acceleration. This excites resonances less than a triangular-velocity profile, which has an initial step-change in acceleration. However the profile requires slightly greater peak acceleration to reach a target position in a specified time.

There is still an instantaneous jerk at the start and finish of the segment as acceleration changes. If this excites resonances in the system, the triangular-jerk profile may be more appropriate.

15.3.8.2.1 step-triangular-accel-position

15.3.8.2.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after step
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.8.2.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.8.2.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.8.2.2 step-triangular-accel-distance**15.3.8.2.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm			Relative distance of end position from start position.
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.8.2.2.2 Possible errors

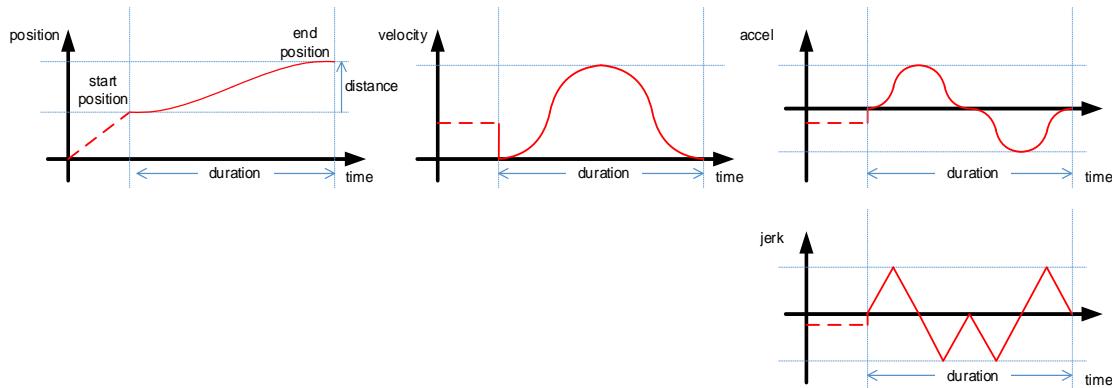
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.8.2.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.8.3 Step with triangular-jerk acceleration/deceleration profiles

Segment function type	Description
step-triangular-jerk-position	Using triangular-jerk acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at end position.
step-triangular-jerk-distance	Using triangular-jerk acceleration and deceleration profiles, accelerate from rest at start position and decelerate to rest at relative distance.



Triangular-jerk acceleration and deceleration profiles as described in 15.3.6.2.3.3 ensure the segment starts and finishes with zero acceleration and zero jerk. This should reduce system excitation to a minimum. However the profile requires slightly greater peak acceleration to reach a target position in a specified time.

15.3.8.3.1 step-triangular-jerk-position

15.3.8.3.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after step
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the step is rounded to the nearest sample period (see 15.7).

15.3.8.3.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.8.3.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.8.3.2 step-triangular-jerk-distance**15.3.8.3.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Distance	pm			Relative distance of end position from start position.
2	Duration	s	> 0		Duration of step

When preparing the waveform, the duration of the acceleration is rounded to the nearest sample period (see 15.7).

15.3.8.3.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero (or negative).

15.3.8.3.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9 “Staircase” repeated steps

A number of applications require repeated “staircase” moves over a range, stepping to successive positions and holding for some period before moving to the next position. This is particularly relevant for Z-stacking, focus-stacking, or for the “slow” axis of an X-Y raster scan. Steps can use S-curve profiles to reduce system excitation and improve settling time.

Staircase moves can be specified with a desired end position or total distance, dividing this distance into the required number of steps. Alternatively the distance per step can be specified.

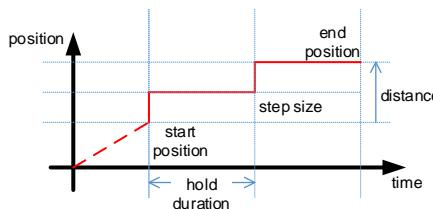
Staircase moves always start with an initial step. If the application requires some operation at the current position before the first step, add a constant-position segment before the staircase. This may need to be longer or shorter than the staircase hold durations, depending on previous movement and settling times, so it is preferable for this to be configured separately.

Steps are always assumed to start with zero velocity; and each staircase step is followed by holding position for some time. By their nature therefore, staircase moves always start and finish with zero velocity.

When using trigger outputs with staircase moves (see 15.6), typically the “start” and “during” triggers are used to trigger after the step, with the “start” trigger providing a trigger pulse after the first step, and the “during” trigger providing trigger pulses after all subsequent steps. To trigger after the step, the trigger offset time for both “start” and “during” triggers is set to the step duration. The number of “during” triggers is set to one less than the number of steps.

15.3.9.1 Basic stepped staircase

Segment function type	Description
staircase-step-position	Move from start position to end position with specified number of steps.
staircase-step-distance	Move from start position to a relative total distance with specified number of steps.
staircase-step-step-size	Move from start position for a specified number of steps, each with the specified relative distance.



The stepped staircase moves to each successive position in turn without applying any measures to smooth the edges of the steps. This is the most basic form of a staircase move. Command trajectory control (see 8.7) may be sufficient to reduce system excitation; if not, it may be better to use a staircase type with less abrupt steps.

15.3.9.1.1 staircase-step-position**15.3.9.1.1.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after staircase
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.1.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.1.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.1.2 staircase-step-position**15.3.9.1.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.

Index	Parameter	Units	Minimum	Maximum	Description
1	End position	pm			Relative distance of end position from start position.
2	Steps		> 0	Number of steps	
3	Hold duration	s	> 0	Hold time for each step	

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.1.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.1.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.1.3 staircase-step-step-size

15.3.9.1.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Step size	pm			Relative distance for each step. May be negative for a negative-going staircase.
2	Steps		> 0	Number of steps	

Index	Parameter	Units	Minimum	Maximum	Description
3	Hold duration	s	> 0	Hold time for each step	

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.1.3.2 Possible errors

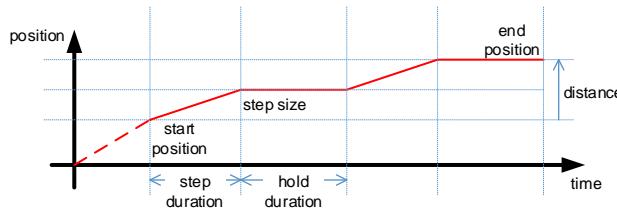
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.1.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.2 Ramped-step staircase

Segment function type	Description
staircase-ramp-position	Move from start position to end position with specified number of steps, using a constant-velocity ramp to move between positions.
staircase-ramp-distance	Move from start position to a relative total distance with specified number of steps, using a constant-velocity ramp to move between positions.
staircase-ramp-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using a constant-velocity ramp to move between positions.



The ramped-step staircase uses a constant-velocity ramp to move between positions, instead of applying a hard step as with the basic staircase. This is the simplest way of smoothing transitions between steps. If this is not sufficient to reduce system excitation, a staircase move using S-curve steps will give better results.

Note that the step duration can be set to zero, at which point this becomes identical to the basic stepped staircase (see 15.3.9.1). The hold duration must always be non-zero.

15.3.9.2.1 staircase-ramp-position

15.3.9.2.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after staircase
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.2.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.2.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.2.2 staircase-ramp-position

15.3.9.2.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Relative distance of end position from start position.
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.2.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.2.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.2.3 staircase-ramp-step-size**15.3.9.2.3.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Step size	pm			Relative distance for each step
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.2.3.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.

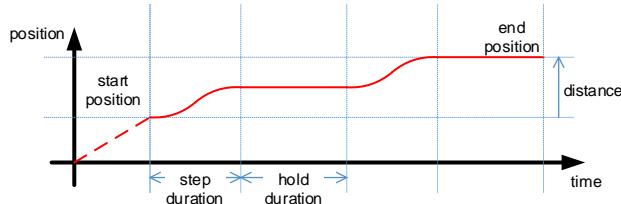
Error reported	Description
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.2.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.3 Staircase using sine-wave profile S-curve steps

Segment function type	Description
staircase-quarter-sine-position	Move from start position to end position with specified number of steps, using quarter-sine S-curve steps.
staircase-quarter-sine-distance	Move from start position to a relative total distance with specified number of steps, using quarter-sine S-curve steps.
staircase-quarter-sine-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using quarter-sine S-curve steps.
staircase-half-sine-position	Move from start position to end position with specified number of steps, using half-sine S-curve steps.
staircase-half-sine-distance	Move from start position to a relative total distance with specified number of steps, using half-sine S-curve steps.
staircase-half-sine-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using half-sine S-curve steps.
staircase-full-sine-position	Move from start position to end position with specified number of steps, using full-sine S-curve steps.
staircase-full-sine-distance	Move from start position to a relative total distance with specified number of steps, using full-sine S-curve steps.
staircase-full-sine-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using full-sine S-curve steps.



To reduce system excitation, sine-wave profile S-curve steps as described in 15.3.7 can be used for the steps. See 15.3.7.1, 15.3.7.2 and 15.3.7.3 for the position, velocity, acceleration and jerk profiles for each step type.

Note that the step duration can be set to zero, at which point this becomes identical to the basic stepped staircase (see 15.3.9.1). The hold duration must always be non-zero.

15.3.9.3.1 staircase-quarter-sine-position, staircase-half-sine-position, staircase-full-sine-position

15.3.9.3.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after staircase
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.3.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.3.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.3.2 staircase-quarter-sine-distance, staircase-half-sine-distance, staircase-full-sine-distance

15.3.9.3.2.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Relative distance of end position from start position.
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.3.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.3.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.3.3 staircase-quarter-sine-step-size, staircase-half-sine-step-size, staircase-full-sine-step-size

15.3.9.3.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Step size	pm			Relative distance for each step
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.3.3.2 Possible errors

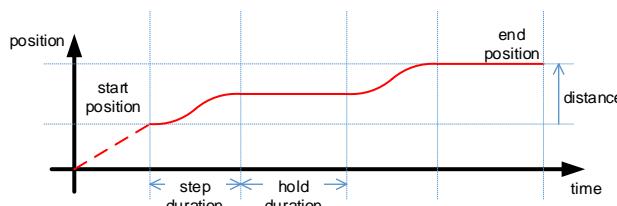
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.3.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.4 Staircase using linear profile S-curve steps

Segment function type	Description
staircase-triangular-velocity-position	Move from start position to end position with specified number of steps, using triangular-velocity S-curve steps.
staircase-triangular-velocity-distance	Move from start position to a relative total distance with specified number of steps, using triangular-velocity S-curve steps.
staircase-triangular-velocity-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using triangular-velocity S-curve steps.
staircase-triangular-acceleration-position	Move from start position to end position with specified number of steps, using triangular-acceleration S-curve steps.
staircase-triangular-acceleration-distance	Move from start position to a relative total distance with specified number of steps, using triangular-acceleration S-curve steps.
staircase-triangular-acceleration-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using triangular-acceleration S-curve steps.
staircase-triangular-jerk-position	Move from start position to end position with specified number of steps, using triangular-jerk S-curve steps.
staircase-triangular-jerk-distance	Move from start position to a relative total distance with specified number of steps, using triangular-jerk S-curve steps.
staircase-triangular-jerk-step-size	Move from start position for a specified number of steps, each with the specified relative distance, using triangular-jerk S-curve steps.



To reduce system excitation, linear profile S-curve steps as described in 15.3.8 can be used for the steps. See 15.3.8.1, 15.3.8.2 and 15.3.8.3 for the position, velocity, acceleration and jerk profiles for each step type.

Note that the step duration can be set to zero, at which point this becomes identical to the basic stepped staircase (see 15.3.9.1). The hold duration must always be non-zero.

15.3.9.4.1 staircase-triangular-velocity-position, staircase-triangular-acceleration-position, staircase-triangular-jerk-position

15.3.9.4.1.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Final position after staircase
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.4.1.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.4.1.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.4.2 staircase-triangular-velocity-distance, staircase-triangular-acceleration-distance, staircase-triangular-jerk-distance**15.3.9.4.2.1 Function parameters**

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	End position	pm			Relative distance of end position from start position.
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.4.2.2 Possible errors

Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.4.2.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.3.9.4.3 staircase-triangular-velocity-step-size, staircase-triangular-acceleration-step-size, staircase-triangular-jerk-step-size

15.3.9.4.3.1 Function parameters

Index	Parameter	Units	Minimum	Maximum	Description
0	Start position	pm			If “continue-position” is set, the end position for the previous segment is held, and this parameter is unused.
1	Step size	pm			Relative distance for each step
2	Steps		> 0		Number of steps
3	Hold duration	s	> 0		Hold time for each step
4	Step duration	s	≥ 0		Step transition time for each step

When preparing the waveform, the hold duration of the step is rounded to the nearest sample period (see 15.7).

15.3.9.4.3.2 Possible errors

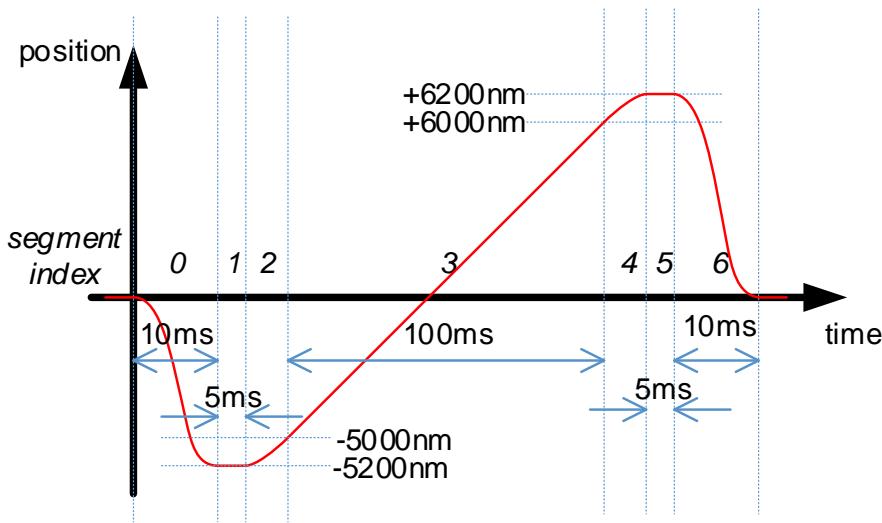
Error reported	Description
duration-zero	For function types which specify a duration, the duration must not be zero.
duration-negative	For function types which specify a duration, the duration must not be negative.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).

15.3.9.4.3.3 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.4 Waveform design example

It may not be clear how to construct a waveform from the component segment types. This section provides a simple worked example as a demonstration, using the commands specified in 15.2 and the function types specified in 15.3.



The example above shows a unidirectional linear ramp from -5000nm to +6000nm over a 100ms duration, with start and end considerations around the ramp. This is a typical operation for many applications, for example focussing a microscope. The waveform carries out the following operations:-

- Travel from the quiescent position to some position before the start of the ramp, using a step with “rounded-off” edges to reduce resonances. Whilst we do not want this to excite system resonances, we are not concerned about the trajectory during this step being particularly accurate, so long as it reaches the position correctly.
- Hold position for a short time to allow any resonances to settle.
- Accelerate to the required ramp rate by the start position for the ramp.
- Ramp at constant velocity, during which time the system will generally be taking measurements or capturing images, depending on the application.
- At the end of the ramp, decelerate to rest.
- Hold position for a short time to allow any resonances to settle.
- Travel back to the quiescent position and stop.

This clearly divides into seven waveform segments.

Segment	Segment type and operation
0	Step from 0nm to -5200nm in 10ms.
1	Hold position for 5ms.
2	Accelerate to required velocity with 200nm “run-up” distance. The ramp covers 11,000nm in 100ms, so the velocity required is 110nm/ms.
3	Ramp from -5000nm to +6000nm in 100ms.
4	Decelerate to rest with 200nm stopping distance.

Segment	Segment type and operation
5	Hold position for 5ms.
6	Step back to 0nm in 10ms.

The example uses constant acceleration profiles for segments 2 and 4, so that the waveform follows a simple trapezoidal trajectory. If this turns out to excite system resonances, the waveform generator provides alternative acceleration profiles which can be used instead. Alternatively it would be possible to set longer “run-up” and stopping distances, allowing acceleration and deceleration to be more progressive.

Similarly, if testing discovered that the “rounded-off” steps in segments 0 and 6 at the start and end did excite system resonances, it would be easy to replace that single-segment step with three segments comprising an alternative acceleration type, a linear ramp and an alternative deceleration type, in the same way as segments 2-4 do. This would give finer control of the waveform trajectory. Alternatively the step could be allowed to take place over a longer time, allowing acceleration and deceleration to be more progressive.

Setting this up in the waveform generator requires the following commands. Note that the controller requires positions to be specified in pm, times to be specified in seconds and velocities to be specified in pm/s, so values are scaled accordingly. For this example, we will assume the stage is connected to channel 1.

Segment	Setting	Command
	Clear waveform	function.waveform-generator.clear 1
0	Step to position	function.waveform-generator.segment.type.set 1 0 step-triangular-velocity-position
	Start from 0nm	function.waveform-generator.segment.parameter.set 1 0 0 0
	Finish at -5200nm	function.waveform-generator.segment.parameter.set 1 0 1 -5200e+3
	Duration 10ms	function.waveform-generator.segment.parameter.set 1 0 2 10e-3
1	Hold position	function.waveform-generator.segment.type.set 1 1 constant-position
	Hold at current position	function.waveform-generator.segment.continue- position-velocity.set 1 1 1 0
	Hold for 5ms	function.waveform-generator.segment.parameter.set 1 1 1 5e-3

Segment	Setting	Command
2	Accelerate from rest to velocity	function.waveform-generator.segment.type.set 1 2 accel-to-velocity-constant-accel-position
	Start at current position and velocity	function.waveform-generator.segment.continue-position-velocity.set 1 2 1 1
	Complete acceleration at -5000nm	function.waveform-generator.segment.parameter.set 1 2 2 -5000e+3
	Accelerate to 110nm/ms	function.waveform-generator.segment.parameter.set 1 2 3 110
3	Linear ramp	function.waveform-generator.segment.type.set 1 3 constant-velocity-position
	Continue at this velocity	function.waveform-generator.segment.continue-position-velocity.set 1 3 1 1
	Ramp to +6000nm	function.waveform-generator.segment.parameter.set 1 3 2 +6000e+3
4	Decelerate to rest	function.waveform-generator.segment.type.set 1 4 accel-to-velocity-constant-accel-position
	Start at current position and velocity	function.waveform-generator.segment.continue-position-velocity.set 1 4 1 1
	Complete deceleration at +6200nm	function.waveform-generator.segment.parameter.set 1 4 2 +6200e+3
	Decelerate to rest (0nm/ms)	function.waveform-generator.segment.parameter.set 1 4 3 0
5	Hold position	function.waveform-generator.segment.type.set 1 5 constant-position
	Hold at current position	function.waveform-generator.segment.continue-position-velocity.set 1 5 1 0
	Hold for 5ms	function.waveform-generator.segment.parameter.set 1 5 1 5e-3

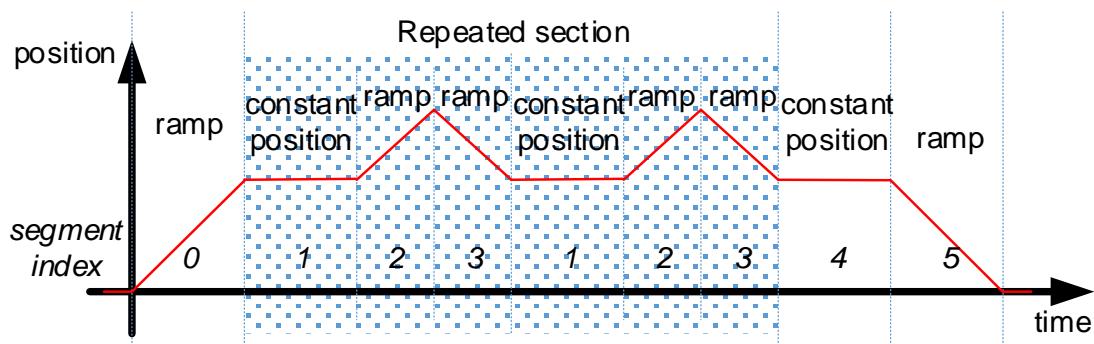
Segment	Setting	Command
6	Step to position	function.waveform-generator.segment.type.set 1 6 step-triangular-velocity-position
	Start from current position	function.waveform-generator.segment.continue-position-velocity.set 1 6 1 0
	Finish at 0nm	function.waveform-generator.segment.parameter.set 1 6 1 0
	Duration 10ms	function.waveform-generator.segment.parameter.set 1 6 2 10e-3
	Use 7 segments for waveform	function.waveform-generator.count.set 1 7
	Check waveform settings (optional)	function.waveform-generator.check-waveform 1
	Prepare waveform for playback	function.waveform-generator.prepare-waveform 1
	Wait for prepare to complete	function.waveform-generator.prepare-waveform-status.get 1
	If check/prepare failed, report faulty segment	function.waveform-generator.failed-at-segment-index.get 1
	If check/prepare failed, report fault	function.waveform-generator.failure-cause.get 1
	Start playback for channel 1	function.command.start 0 0 1 0 0

After the waveform has been set up, the user may also wish to set up parts of the waveform to repeat (see 15.5), or have function playback control the digital trigger output (see 15.6).

Once all elements of the waveform have been configured, it must be prepared for playback (see 15.7). It can then be started and stopped with PC commands (see 14.4) or digital trigger inputs (see 14.5).

15.5 Waveform design with repeating waveforms

It is often necessary to run some part of the waveform repeatedly. In the example below, the repeated section of the waveform holds a constant position for some time, then carries out a triangular ramp up and down.



The waveform generator allows a section of the waveform to be repeated a number of times as required (twice in the example above). The start and end segments of the repeated section and the repeat count are specified. After the section has carried out the specified number of repeats, the remainder of the waveform is run. The positions at the start of the first repeated segment and the end of the last repeated segment must usually be the same, otherwise there will be a step back to the start position on the first repeated segment. Setting “continue-position” (see 15.2.4) for the first repeated segment is only effective from the segment immediately before, and not when repeating.

As the example above shows, repeated waveforms often require an initial section which moves the system to the required position, and a final section which returns the system to the initial point. Setting the start segment greater than zero (i.e. not at the start of the entire waveform) gives an initial section which will be run once and not repeated; and setting the end segment before the last segment (i.e. not at the end of the entire waveform) gives a final section which will be run once after the repeated section.

The example above would be configured with:

- Segment count set to 6
- Start segment set to 1
- End segment set to 3
- Repeat count set to 2

If the entire waveform is to be repeated, initial or final sections need not be present. In this case the start segment would be set to zero and the end segment would be set to the last segment in the waveform (one less than the segment count).

Repeated waveforms may also be required to repeat forever. This is selected by setting the repeat count to zero. If the waveform needs to be stopped, the user must stop it manually using an appropriate “stop” command (see 14.4 and 14.5). A final section for the waveform (after the repeated section) may exist but will never be carried out.

15.5.1 function.waveform-generator.repeat-start.get/set

15.5.1.1 Description

Waveform generator start segment for repeating

15.5.1.2 Command

```
function.waveform-generator.repeat-start.get <channel>
function.waveform-generator.repeat-start.set <channel> <value>
```

15.5.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.5.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer		0	999	Waveform generator start segment for repeating

15.5.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		0	999	Waveform generator start segment for repeating

15.5.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Segment index must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.5.1.7 Minimum security level

function.waveform-generator.repeat-start.get	No security required
function.waveform-generator.repeat-start.set	User level security required

15.5.1.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.5.2 function.waveform-generator.repeat-end.get/set

15.5.2.1 Description

Waveform generator end segment for repeating

15.5.2.2 Command

```
function.waveform-generator.repeat-end.get <channel>
function.waveform-generator.repeat-end.set <channel> <value>
```

15.5.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.5.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer		0	999	Waveform generator end segment for repeating

15.5.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		0	999	Waveform generator end segment for repeating

15.5.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Segment index must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.5.2.7 Minimum security level

function.waveform-generator.repeat-end.get	No security required
function.waveform-generator.repeat-end.set	User level security required

15.5.2.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.5.3 function.waveform-generator.repeat-count.get/set**15.5.3.1 Description**

Waveform generator number of repeats of waveform (0=repeat forever)

15.5.3.2 Command

```
function.waveform-generator.repeat-count.get <channel>
function.waveform-generator.repeat-count.set <channel> <value>
```

15.5.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.5.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer		0		Waveform generator number of repeats of waveform 0 = repeat forever 1 = play once (no repeat) 2+ = repeat specified number of times

15.5.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		0		Waveform generator number of repeats of waveform 0 = repeat forever 1 = play once (no repeat) 2+ = repeat specified number of times

15.5.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.5.3.7 Minimum security level

function.waveform-generator.repeat-count.get	No security required
function.waveform-generator.repeat-count.get	User level security required

15.5.3.8 Supported in

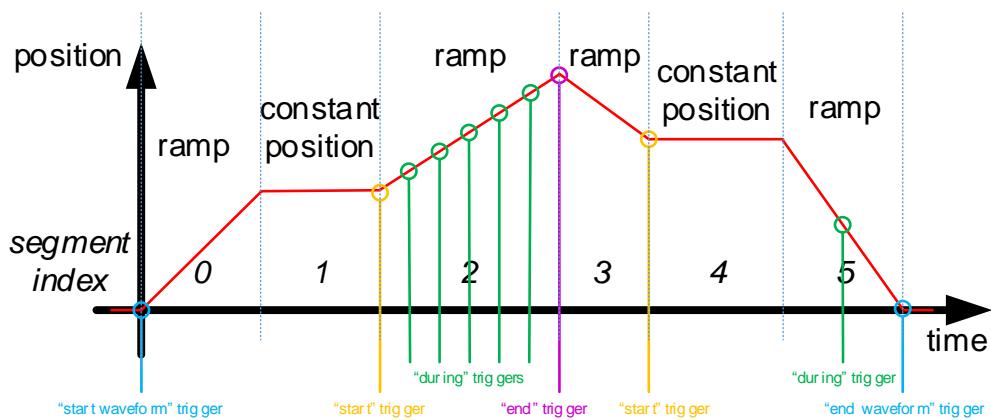
Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.6 Waveforms with synchronised digital trigger outputs

15.6.1 Trigger points within waveform

As described in 14.6, the controller has one digital trigger output per channel which may be pulsed synchronously with the waveform during function playback to activate external equipment.

From the waveform generator, the trigger output for each channel may be activated at the start or end of a segment, or at evenly-spaced intervals during a segment. The number of “during” trigger points can be set for each segment, dividing the segment into “ $n+1$ ” intervals and triggering at each interval (not including the start and end points). Triggers can also be set at the start and end of the entire waveform.



The example above demonstrates the various trigger points available within the function playback waveform. Triggers are set at the start and end of the entire waveform. Segment 2 has “start” and “end” triggers set, and also has 5 “during” triggers set. The “during” triggers are evenly spaced, so setting 5 triggers as in the example above will divide the segment into 6 equal parts. Segments 4 and 5 show that it is possible to set just “start”, “end” or “during” triggers (with 1 “during” trigger occurring at the midpoint of the segment). Using the trigger output is optional, and segment 1 shows that it is also possible to have no triggers set at all.

Mechanical systems often require some time for resonances to settle before external equipment is triggered. Offset times may be specified for triggers to allow for this.

15.6.2 Trigger output event types

Trigger output event types are described in 14.6.2. For convenience, they are repeated here.

Trigger output event type	Description
none	No trigger is set at this trigger position.
pulse	Trigger output is activated immediately. This is the “traditional” time-based triggering method.
above-command	Trigger output is activated when measured position is greater than commanded position at trigger point.

Trigger output event type	Description
	If measured position is already greater than commanded position, trigger output is activated immediately.
below-command	<p>Trigger output is activated when measured position is lower than commanded position at trigger point.</p> <p>If measured position is already lower than commanded position, trigger output is activated immediately.</p>
rising-edge	<p>Trigger output is activated when measured position becomes greater than commanded position at trigger point.</p> <p>If measured position is already greater than commanded position, measured position must fall below this commanded position value and then increase above this commanded position value again for the trigger output to be activated.</p>
falling-edge	<p>Trigger output is activated when measured position becomes lower than commanded position at trigger point.</p> <p>If measured position is already lower than commanded position, measured position must increase above this commanded position value and fall below this commanded position value again for the trigger output to be activated.</p>
either-edge	Trigger output is activated when measured position crosses commanded position at trigger point in either direction.
in-position	<p>Trigger output is activated when measured position is within the in-position error threshold (<code>stage.in-position.error-threshold</code>, see 8.5.1) of commanded position at trigger point.</p> <p>If measured position is already within this range, trigger output is activated immediately.</p>

15.6.3 Trigger output offset times

To allow for system movement, mechanical resonances settling or any other reasons, the “start”, “during” and “end” trigger points for each segment may be offset by some time earlier or later within the segment, relative to their nominal positions. Separate offset times are provided for “start”, “during” and “end” trigger points. If multiple “during” triggers are specified, all “during” triggers are delayed by the same amount relative to the nominal position within the segment.

Offset times may not place the trigger points before the start of the segment or after the end of the segment. This gives us the following limits.

- For “start” trigger points,

$$0 \leq \text{"start" offset time} \leq \text{segment duration}$$

- For “during” trigger points,

$$-\frac{\text{segment duration}}{\text{number of "during" triggers} + 1} \leq \text{"during" offset time} \leq \frac{\text{segment duration}}{\text{number of "during" triggers} + 1}$$

- For “end” trigger points,

$$-\text{segment duration} \leq \text{"end" offset time} \leq 0$$

If triggering before the start of the segment or after the end of the segment is necessary, the triggers must be set up for the previous or following segment respectively.

In rare cases where the exact duration of a segment may not be known in advance and a trigger may need to be configured in a different segment, the user (or application) should carry out a waveform check (`function.waveform-generator.check-waveform`, see 15.7.6) so that the duration of each segment can be reported. Triggers can then be configured in the appropriate segment before preparing the waveform points and triggers.

Note that threshold-based triggers take the threshold value from the commanded position at the time of the trigger, not at any nominal point within the segment. For example, if a segment ramping linearly from 10 μm to 30 μm in 1s was given a “start” trigger with an offset of 0.1s, the commanded position at that offset time would be 12 μm and the trigger would use that threshold value.

There is no requirement for “start”, “during” and “end” trigger points to occur in that order. Offset times may be used to place the trigger points anywhere within the segment, relative to their nominal positions. Note though that if trigger points are coincident then only one trigger can occur. Also note that if triggers overlap then the result will be a single extended trigger pulse, as described in 14.6.5.

15.6.4 Multiple triggers at irregular intervals

Some applications may require triggers at irregular intervals during the waveform. As described, the “start”, “during” and “end” triggers may be used together with offset times to place up to 3 triggers at any point within a segment.

If more triggers are required, the typical solution is to divide a single large segment into two or more segments with the same configuration, continuing the waveform steplessly (in position and velocity, see 15.2.4) from one segment to the next. Each additional segment allows a further 3 triggers to be freely placed within that segment. This is particularly effective for ramp, sine-wave and staircase-step function profiles, where segment transitions can easily be made fully stepless.

15.6.5 function.waveform-generator.trigger-output.start-trigger.get/set**15.6.5.1 Description**

Waveform generator trigger output event at start, if required

15.6.5.2 Command

```
function.waveform-generator.trigger-output.start-trigger.get <channel>
function.waveform-generator.trigger-output.start-trigger.set <channel>
<value>
```

15.6.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.6.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at start See 15.6.2 for available trigger events

15.6.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at start See 15.6.2 for available trigger events

15.6.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Trigger event must be as specified (set only)

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.5.7 Minimum security level

function.waveform-generator.trigger-output.start-trigger.get	No security required
function.waveform-generator.trigger-output.start-trigger.set	User level security required

15.6.5.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.6.6 function.waveform-generator.trigger-output.end-trigger.get/set**15.6.6.1 Description**

Waveform generator trigger output event at end, if required

15.6.6.2 Command

```
function.waveform-generator.trigger-output.end-trigger.get <channel>
function.waveform-generator.trigger-output.end-trigger.set <channel>
<value>
```

15.6.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.6.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at end See 15.6.2 for available trigger events

15.6.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at end See 15.6.2 for available trigger events

15.6.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Trigger event must be as specified (set only)

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.6.7 Minimum security level

function.waveform-generator.trigger-output.end-trigger.get	No security required
function.waveform-generator.trigger-output.end-trigger.set	User level security required

15.6.6.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.6.7 function.waveform-generator.segment.trigger-output.start-trigger.get/set**15.6.7.1 Description**

Waveform generator trigger output event at start of segment, if required

15.6.7.2 Command

```
function.waveform-generator.trigger-output.segment.start-trigger.get  
<channel> <segment>
```

```
function.waveform-generator.trigger-output.segment.start-trigger.set  
<channel> <segment> <value>
```

15.6.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at start of segment See 15.6.2 for available trigger events

15.6.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at start of segment See 15.6.2 for available trigger events

15.6.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Value out of range	Trigger event must be as specified (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.7.7 Minimum security level

function.waveform-generator.segment.trigger-output.start-trigger.get	No security required
function.waveform-generator.segment.trigger-output.start-trigger.set	User level security required

15.6.7.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.6.8 function.waveform-generator.segment.trigger-output.start-trigger-offset.get/set

15.6.8.1 Description

Offset time of trigger output event at start of segment

15.6.8.2 Command

```
function.waveform-generator.segment.trigger-output.start-trigger-
offset.get <channel> <segment>
```

```
function.waveform-generator.segment.trigger-output.start-trigger-
offset.set <channel> <segment> <value>
```

15.6.8.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.8.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	32-bit floating-point	s	0		Offset time of trigger output event at start of segment

15.6.8.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Offset time of trigger output event at start of segment

15.6.8.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range

Error return “errcode” string reported	Description
Value out of range	Offset time must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.8.7 Minimum security level

function.waveform-generator.segment.trigger-output.start-trigger-offset.get	No security required
function.waveform-generator.segment.trigger-output.start-trigger-offset.set	User level security required

15.6.8.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.6.9 function.waveform-generator.segment.trigger-output.end-trigger.get/set**15.6.9.1 Description**

Waveform generator trigger output event at end of segment, if required

15.6.9.2 Command

```
function.waveform-generator.trigger-output.segment.end-trigger.get  
<channel> <segment>
```

```
function.waveform-generator.trigger-output.segment.end-trigger.set  
<channel> <segment> <value>
```

15.6.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at end of segment See 15.6.2 for available trigger events

15.6.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output event at end of segment See 15.6.2 for available trigger events

15.6.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Value out of range	Trigger event must be as specified (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.9.7 Minimum security level

function.waveform-generator.segment.trigger-output.end-trigger.get	No security required
function.waveform-generator.segment.trigger-output.end-trigger.set	User level security required

15.6.9.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.6.10 function.waveform-generator.segment.trigger-output.end-trigger-offset.get/set**15.6.10.1 Description**

Offset time of trigger output event at end of segment

15.6.10.2 Command

```
function.waveform-generator.segment.trigger-output.end-trigger-
offset.get <channel> <segment>
```

```
function.waveform-generator.segment.trigger-output.end-trigger-
offset.set <channel> <segment> <value>
```

15.6.10.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.10.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	32-bit floating-point	s		0	Offset time of trigger output event at end of segment

15.6.10.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s		0	Offset time of trigger output event at end of segment

15.6.10.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range

Error return “errcode” string reported	Description
Value out of range	Offset time must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.10.7 Minimum security level

function.waveform-generator.segment.trigger-output.end-trigger-offset.get	No security required
function.waveform-generator.segment.trigger-output.end-trigger-offset.set	User level security required

15.6.10.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.6.11 function.waveform-generator.segment.trigger-output.during-trigger.get/set

15.6.11.1 Description

Waveform generator trigger output events during segment, if required

15.6.11.2 Command

```
function.waveform-generator.trigger-output.segment.during-trigger.get
<channel> <segment>
```

```
function.waveform-generator.trigger-output.segment.during-trigger.set
<channel> <segment> <value>
```

15.6.11.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.11.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output events during segment See 15.6.2 for available trigger events

15.6.11.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			Waveform generator trigger output events during segment See 15.6.2 for available trigger events

15.6.11.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Value out of range	Trigger event must be as specified (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.11.7 Minimum security level

function.waveform-generator.segment.trigger-output.during-trigger.get	No security required
function.waveform-generator.segment.trigger-output.during-trigger.set	User level security required

15.6.11.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.6.12 function.waveform-generator.segment.trigger-output.during-trigger-offset.get/set

15.6.12.1 Description

Offset time of trigger output events during segment

15.6.12.2 Command

```
function.waveform-generator.segment.trigger-output.during-trigger-
offset.get <channel> <segment>
```

```
function.waveform-generator.segment.trigger-output.during-trigger-
offset.set <channel> <segment> <value>
```

15.6.12.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.12.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	32-bit floating-point	s			Offset time of trigger output events during segment

15.6.12.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s			Offset time of trigger output events during segment

15.6.12.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.12.7 Minimum security level

function.waveform-generator.segment.trigger-output.during-trigger-offset.get	No security required
function.waveform-generator.segment.trigger-output.during-trigger-offset.set	User level security required

15.6.12.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.6.13 function.waveform-generator.segment.trigger-output.during-trigger-count.get/set

15.6.13.1 Description

Waveform generator number of trigger output events during segment

15.6.13.2 Command

```
function.waveform-generator.trigger-output.segment.during-trigger-count.get <channel> <segment>
```

```
function.waveform-generator.trigger-output.segment.during-trigger-count.set <channel> <segment> <value>
```

15.6.13.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.6.13.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index
value	32-bit unsigned integer		0		Waveform generator number of trigger output events during segment

15.6.13.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		0		Waveform generator number of trigger output events during segment

15.6.13.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range

Error return “errcode” string reported	Description
Value out of range	Number of trigger output events cannot be greater than the maximum number of samples for function playback (see 15.7)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.6.13.7 Minimum security level

function.waveform-generator.segment.trigger-output.during-trigger-count.get	No security required
function.waveform-generator.segment.trigger-output.during-trigger-count.set	User level security required

15.6.13.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7 Waveform check, waveform preparation and sampled data points

15.7.1 Waveform generation and sample rates

Waveform segments are not played directly. Instead, the waveform generator must generate a series of sampled data points for the function playback waveform (see 14.2). The controller allows up to 500,000 points of sampled data in a function generator waveform, which equates to 10s of sample-accurate waveform playback at the controller's 50kHz sample rate.

If a longer waveform is required, the sample rate for function generator waveform playback can be set slower using the command `function.waveform-generator.sample-period.get/set` (see 15.7.5). 500,000 points at a sample rate of 1ms would allow up to 500s of waveform playback, for instance. The sample period specified will be rounded to the nearest multiple of the DSP sample period (20 μ s for 50kHz).

A slower sample rate may also be chosen to allow the waveform generator to prepare data points more quickly. The more data points the waveform generator is required to calculate, the longer the preparation process takes. Typically a waveform using all 500,000 points can be prepared within 2-3s, but if the user requires a faster response time between configuring the waveform generator and being able to start function playback, it may be necessary to use a slower sample rate and reduce the number of points in the waveform. At slower sample rates, the controller ramps linearly between data points to reduce the impact of the slower sample rate.

Each waveform generator segment must be fitted to the sample rate. As described in 15.3, this may cause some rounding of segment durations to fit the sample rate. If the sample rate is reduced to a point where rounding can affect the waveform then different function types may be more appropriate depending on the application.

The effects of reduced sample rate will be most significant in nonlinear waveform segments such as sine waves or S-curve accelerations, where substantially reducing the number of data points will affect how accurately the controller can plot the trajectory. In this case the sample rate should be set faster than the bandwidth of the system, so that errors due to the slower sample rate are naturally "smoothed out" by the system.

In practise the typical bandwidth of stages and connected systems means that sample periods up to 200 μ s (5kHz) will normally have little effect on the waveform, providing up to 100s of waveform playback at this rate.

15.7.2 Waveform generator check

The first step in waveform generation is to check that the waveform is valid. Parameters for segments must be validated, the time to carry out each segment must be calculated, and the entire waveform must be checked to ensure it fits within the number of sample points available. This check may be done separately using the command `function.waveform-generator.check-waveform` (see 15.7.6) or may be carried out at the same time as preparing data points (see 15.7.4).

When the check is successful, the waveform generator allows the user to check each segment's duration, start and end positions, and start and end velocities, as well as the duration for the entire waveform. This may be of use when reporting back details of the waveform on a user interface.

If the check fails, the waveform generator reports which segment has a fault using the command function `waveform-generator.failed-at-segment-index.get` (see 15.7.10), and what the fault is using command function `waveform-generator.failure-cause.get` (see 15.7.11). This allows a user interface to provide guidance on how the user can solve the fault.

15.7.3 Waveform generator error reporting

Command function `waveform-generator.failure-cause.get` (see 15.7.11) reports a standard set of faults if the waveform generator check fails. Some faults (e.g. “no-segments-set”) are applicable for the waveform configuration generally. Most faults relate to misconfiguration of the function type, and section 15.3 specifies which faults may be applicable for each function type.

15.7.3.1 General errors

Fault reported	Description
none	Waveform check/prepare succeeded without errors.
no-segments-set	No waveform segments have been set up (number of segments is zero).
segment-type-not-set	The function type for this waveform segment has not been set.
waveform-too-long	As described in 15.7.1, the waveform can only be up to 500,000 samples long. If the waveform exceeds this, this fault will be reported for the segment where the waveform runs out of samples. Either increase the sample period, or construct a waveform which requires less data.
too-few-samples-for-triggers	As described in 15.6.1, trigger output pulses can be set up for a segment. As for the commanded position, trigger pulses are synchronised on sample periods. If the sample period is set too long or the segment is very short, it is possible that a segment may require more trigger pulses than there are samples in the segment. Either change the sample period, change the segment configuration, or change the number of trigger pulses required.
trigger-offset-too-large-positive	As described in 15.6.1, trigger output pulses can be set up for a segment, and a time offset can be applied for the trigger output relative to the trigger's position in the segment. A positive time offset has been set which puts the trigger time beyond the end of the segment. Either reduce the time offset, or specify the trigger to occur in the following segment.
trigger-offset-too-large-negative	As described in 15.6.1, trigger output pulses can be set up for a segment, and a time offset can be applied for the trigger output relative to the trigger's position in the segment. A negative time offset has been set which puts the trigger time before the start of the

Fault reported	Description
	segment. Either reduce the time offset, or specify the trigger to occur in the previous segment.

15.7.3.2 Function type specific errors

Error reported	Description
duration-zero	For some function types which specify a duration, the duration must not be zero. Note that some function types allow zero-length durations.
duration-negative	For function types which specify a duration, the duration must not be negative.
distance-zero	For function types which specify a distance or start/end positions, the distance must not be zero.
velocity-zero	For function types which specify a velocity, the velocity must not be zero.
accel-zero	For function types which specify an acceleration, the acceleration must not be zero.
steps-zero	For function types which specify a number of steps, the number of steps must not be zero (or negative).
start-end-velocities-zero	For acceleration/deceleration function types, the start and end velocities cannot both be zero.
start-end-velocities-same	For acceleration/deceleration function types, the start and end velocities cannot both be the same.
start-end-velocities-change-direction	For acceleration/deceleration function types, the start and end velocities cannot be in different directions (different signs). Note that either can be zero. If the stage is required to change direction, use one segment for decelerating to zero velocity, and a second segment for accelerating to the required velocity in the opposite direction.
velocity-wrong-direction-for-distance	For function types which specify a velocity and a distance or start/end positions, the velocity must be in same direction (same sign) as the distance or in the correct direction to reach the end position.
start-not-first-cycle	For periodic functions such as sine waves, the start phase must be within the first cycle. Since phase values are set as a fraction of a cycle, the start phase must be between 0 and 1.

Error reported	Description
end-before-start	For periodic functions such as sine waves, the end must be set after the start.
amplitude-negative	For periodic functions such as sine waves, the amplitude is set as a magnitude and must not be negative.
frequency-zero	For periodic functions such as sine waves, the frequency cannot be zero (or negative).
frequency-above-half-sample-rate	For periodic functions such as sine waves, the frequency cannot be greater than half the waveform sample rate (the Nyquist limit). Either reduce the sample period to allow waveforms with higher frequencies to be constructed, or change the frequency required.

15.7.4 Waveform generator sampled data point preparation

The preparation step using the command `function.waveform-generator.prepare-waveform` (see 15.7.7) generates sampled data points for each segment to construct the waveform. If the waveform has not been checked (see 15.7.2), this is done automatically before the data points are generated. Data points cannot be generated if the check fails.

The preparation process can take up to 2-3s, which is too long for a single command to wait for results. Preparation is started with one command, and the user must then check the preparation status repeatedly until it completes.

Once the preparation process has been completed successfully, function playback can be started using the commands in 14.4 and 14.5.

Alternatively a waveform may be prepared and then playback automatically started after preparation using the command `function.waveform-generator.prepare-start` (see 15.7.8). This intentionally follows the same design as commands in 14.4 so that playback can be started synchronously for multiple channels.

15.7.5 function.waveform-generator.sample-period.get/set

15.7.5.1 Description

Period of samples in generated waveform

15.7.5.2 Command

```
function.waveform-generator.sample-period.get <channel>  
function.waveform-generator.sample-period.set <channel> <value>
```

15.7.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Period of samples in generated waveform

The sample period will be rounded to the nearest multiple of the controller's sample period (20µs for the controller's 50kHz sample rate). The result returned may therefore not be exactly the value set, if rounding is required.

15.7.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Period of samples in generated waveform

15.7.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Sample period must be greater than zero (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

15.7.5.7 Minimum security level

function.waveform-generator.sample-period.get	No security required
function.waveform-generator.sample-period.set	User level security required

15.7.5.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.6 function.waveform-generator.check-waveform

15.7.6.1 Description

Check waveform segments and calculate segment boundaries

15.7.6.2 Command

```
function.waveform-generator.check-waveform <channel>
```

15.7.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		1	1	Always returns 1 if check succeeds. If check fails, returns an error.

15.7.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	The waveform is not configured correctly. Use <code>function.waveform-generator.prepare-waveform-status.get</code> (see 15.7.10) to find which waveform segment is causing the problem
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

15.7.6.6 Minimum security level

function.waveform-generator.check-waveform	User level security required
--	------------------------------

15.7.6.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.7 function.waveform-generator.prepare-waveform

15.7.7.1 Description

Prepare waveform for playback, calculating sample data points

15.7.7.2 Command

```
function.waveform-generator.prepare-waveform <channel>
```

15.7.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	32-bit unsigned integer		1	1	Always returns 1 if prepare can be started

15.7.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

15.7.7.6 Minimum security level

function.waveform-generator.prepare-waveform	User level security required
--	------------------------------

15.7.7.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.8 function.waveform-generator.prepare-start

15.7.8.1 Description

Prepare waveform for playback for one or more channels, calculating sample data points, and start playback synchronously on channels after preparing

15.7.8.2 Command

```
function.waveform-generator.prepare-start <start-snapshot> <start-internal-channel0> <start-channel1> [start-channel2] [start-channel3]
```

15.7.8.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
start-snapshot	8-bit unsigned integer	Boolean	0	1	Start snapshot capture on starting playback
start-internal-channel0	8-bit unsigned integer	Boolean	0	1	Prepare waveform and start function playback for internal channel 0
start-channel1	8-bit unsigned integer	Boolean	0	1	Prepare waveform and start function playback for channel 1
start-channel2	8-bit unsigned integer	Boolean	0	1	Prepare waveform and start function playback for channel 2 (only required for 2- or 3-channel controller)
start-channel3	8-bit unsigned integer	Boolean	0	1	Prepare waveform and start function playback for channel 3 (only required for 3-channel controller)

15.7.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
start-snapshot	8-bit unsigned integer	Boolean	0	1	Snapshot capture will be started when function playback starts
start-internal-channel0	8-bit unsigned integer	Boolean	0	1	Preparing waveform and starting function playback for internal channel 0

Name	Type	Units	Minimum	Maximum	Description
start-channel1	8-bit unsigned integer	Boolean	0	1	Preparing waveform and starting function playback for channel 1
start-channel2	8-bit unsigned integer	Boolean	0	1	Preparing waveform and starting function playback for channel 2 (only returned for 2- or 3-channel controller)
start-channel3	8-bit unsigned integer	Boolean	0	1	Preparing waveform and starting function playback for channel 3 (only returned for 3-channel controller)

15.7.8.5 Possible error reports

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

15.7.8.6 Minimum security level

function.waveform-generator.prepare-start	User level security required
---	------------------------------

15.7.8.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.7.9 function.waveform-generator.prepare-waveform-status.get

15.7.9.1 Description

Status of waveform preparation

15.7.9.2 Command

```
function.waveform-generator.prepare-waveform-status.get <channel>
```

15.7.9.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “prepare-waveform-status”			<p>Status of waveform preparation</p> <p>“idle”: Not running, and generated waveform sample data points successfully</p> <p>“error”: Not running, and waveform has an error which caused waveform preparation to fail. Use function.waveform-generator.prepare-waveform-status.get (see 15.7.10) to find which waveform segment is causing the problem.</p> <p>“in-progress”: Waveform generator is still generating sample data points</p>

15.7.9.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

15.7.9.6 Minimum security level

function.waveform-generator.prepare-waveform-status.get	No security required
---	----------------------

15.7.9.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.10 function.waveform-generator.failed-at-segment-index.get**15.7.10.1 Description**

If waveform generation failed, reports the segment index at which an error occurred when last running waveform check or prepare

15.7.10.2 Command

```
function.waveform-generator.failed-at-segment-index.get <channel>
```

15.7.10.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit signed integer				Segment index at which an error occurred Returns -1 if last check or prepare completed successfully

15.7.10.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

15.7.10.6 Minimum security level

function.waveform-generator.failed-at-segment-index.get	No security required
---	----------------------

15.7.10.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.11 function.waveform-generator.failure-cause.get**15.7.11.1 Description**

If waveform generation failed, reports details of the cause of the error which occurred when last running waveform check or prepare

15.7.11.2 Command

```
function.waveform-generator.failure-cause.get <channel>
```

15.7.11.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “waveform-failure-cause”			Waveform generator failure cause See 15.7.3 for possible values.

15.7.11.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

15.7.11.6 Minimum security level

function.waveform-generator.failure-cause.get	No security required
---	----------------------

15.7.11.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

15.7.12 function.waveform-generator.waveform-duration.get**15.7.12.1 Description**

Total duration for generated waveform

15.7.12.2 Command

```
function.waveform-generator.waveform-duration.get <channel>
```

15.7.12.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

15.7.12.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Total duration for generated waveform

15.7.12.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Waveform check (see <code>function.waveform-generator.check-waveform</code> , 15.7.6) or prepare (see <code>function.waveform-generator.prepare-waveform</code> , 15.7.7) has not been run successfully since waveform configuration was last changed.

15.7.12.6 Minimum security level

function.waveform-generator.waveform-duration.get	No security required
---	----------------------

15.7.12.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.13 function.waveform-generator.segment.duration.get**15.7.13.1 Description**

Waveform generator segment duration

15.7.13.2 Command

```
function.waveform-generator.segment.duration.get <channel> <segment>
```

15.7.13.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.7.13.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Waveform generator segment duration

15.7.13.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Command could not be carried out	Waveform check (see function.waveform-generator.check-waveform, 15.7.6) or prepare (see function.waveform-generator.prepare-waveform, 15.7.7) has not been run successfully since waveform configuration was last changed.

15.7.13.6 Minimum security level

function.waveform-generator.segment.duration.get	No security required
--	----------------------

15.7.13.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.14 function.waveform-generator.segment.start-position.get**15.7.14.1 Description**

Waveform generator segment start position

15.7.14.2 Command

```
function.waveform-generator.segment.start-position.get <channel>
<segment>
```

15.7.14.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.7.14.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Waveform generator segment start position

15.7.14.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Command could not be carried out	Waveform check (see <code>function.waveform-generator.check-waveform</code> , 15.7.6) or prepare (see <code>function.waveform-generator.prepare-waveform</code> , 15.7.7) has not been run successfully since waveform configuration was last changed.

15.7.14.6 Minimum security level

function.waveform-generator.segment.start-position.get	No security required
--	----------------------

15.7.14.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.15 function.waveform-generator.segment.start-velocity.get**15.7.15.1 Description**

Waveform generator segment start velocity

15.7.15.2 Command

```
function.waveform-generator.segment.start-velocity.get <channel>
<segment>
```

15.7.15.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.7.15.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	nm/ms			Waveform generator segment start velocity

15.7.15.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Command could not be carried out	Waveform check (see <code>function.waveform-generator.check-waveform</code> , 15.7.6) or prepare (see <code>function.waveform-generator.prepare-waveform</code> , 15.7.7) has not been run successfully since waveform configuration was last changed.

15.7.15.6 Minimum security level

function.waveform-generator.segment.start-velocity.get	No security required
--	----------------------

15.7.15.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

NOTE: In 6.4.x and 6.5.x firmware, the velocity units for this return value were pm/s. For 6.6.1 onwards, the velocity units were changed to nm/ms as stated.

15.7.16 function.waveform-generator.segment.end-position.get**15.7.16.1 Description**

Waveform generator segment end position

15.7.16.2 Command

```
function.waveform-generator.segment.end-position.get <channel>
<segment>
```

15.7.16.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.7.16.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Waveform generator segment end position

15.7.16.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Command could not be carried out	Waveform check (see <code>function.waveform-generator.check-waveform</code> , 15.7.6) or prepare (see <code>function.waveform-generator.prepare-waveform</code> , 15.7.7) has not been run successfully since waveform configuration was last changed.

15.7.16.6 Minimum security level

function.waveform-generator.segment.end-position.get	No security required
--	----------------------

15.7.16.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

15.7.17 function.waveform-generator.segment.end-velocity.get**15.7.17.1 Description**

Waveform generator segment end velocity

15.7.17.2 Command

```
function.waveform-generator.segment.end-velocity.get <channel>
<segment>
```

15.7.17.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
segment	32-bit unsigned integer		0	999	Segment index

15.7.17.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	nm/ms			Waveform generator segment end velocity

15.7.17.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Index out of range	Segment index must be within specified range
Command could not be carried out	Waveform check (see <code>function.waveform-generator.check-waveform</code> , 15.7.6) or prepare (see <code>function.waveform-generator.prepare-waveform</code> , 15.7.7) has not been run successfully since waveform configuration was last changed.

15.7.17.6 Minimum security level

function.waveform-generator.segment.end-velocity.get	No security required
--	----------------------

15.7.17.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

NOTE: In 6.4.x and 6.5.x firmware, the velocity units for this return value were pm/s. For 6.6.1 onwards, the velocity units were changed to nm/ms as stated.

16 Waveform builder

16.1 Overview

The waveform generator (see section 15) provides a way for users to set up a waveform meeting their exact requirements. Where this level of control over the waveform may not be required, the waveform builder provides the ability to construct entire preprogrammed waveforms with only certain key parameters set by the user.

The waveform builder uses the waveform generator for this. The resulting waveform may be read back or edited through the waveform generator (see section 15) or function playback (see section 14) as required.

Waveforms constructed with the waveform builder may be started or stopped by the PC or by external electronics using digital trigger inputs, as described in 14.4 and 14.5. Alternatively the waveform builder also allows waveforms to be constructed and started with a single command, in a similar way to the same way as the waveform builder command `function.waveform-generator.prepare-start` (see 15.7 and 15.7.8).

16.2 Waveform builder and sample rates

When using the waveform generator to construct waveforms longer than 10s, the sample period must be set to allow the length of waveform required (see 15.7.1). The waveform generator will report the fault “waveform-too-long” when checking or preparing the waveform if this is not set appropriately (see 15.7.3.1).

The waveform builder does not require this. Since the length of waveform is known, the waveform builder automatically sets an appropriate sample period for longer waveforms. This is transparent for users, to fit the intent of the waveform builder that a user should be able to specify the waveform at a higher level and not need to consider the details of its implementation.

16.3 Staircase-step waveform builder

16.3.1 Overview

The staircase-step waveform builder provides a convenient way to generate a stepped waveform. Primary uses are in microscopy to carry out Z-stacking or focus-stacking, and for users carrying out raster scans.

The step size and number of steps must be set. A negative step size produces a negative-going stepped waveform. The stepped waveform may simply run in one direction and return to the start; or it may be bidirectional, running a stepped waveform in one direction, and then reversing direction to run the same number (and size) of steps returning to the start.

To reduce system excitation, steps use S-curve profiles with zero initial jerk (see 15.3.5.3). The step duration (to reduce system excitation), and hold duration, are all configurable. If an S-curve profile for steps is not required, the step duration may be set to zero. The hold duration must always be non-zero.

Since users will normally be conducting measurements or acquiring images at each stepped position, trigger outputs can be enabled after each step. The trigger type is configurable, as is the delay time between completion of the edge and triggering the output.

The stepped waveform may be set to repeat a number of times, or indefinitely. (See also 15.5 for details of the waveform generator's handling of repeated sections.) This is often required for time-lapse Z-stacking in microscopy, and for raster scans.

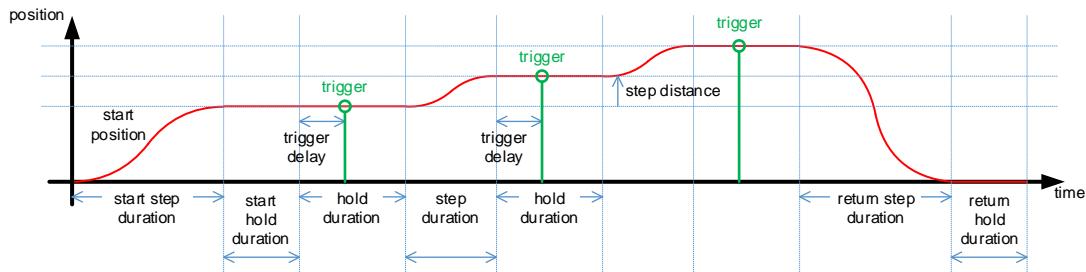
16.3.2 Waveform configurations

The following diagrams demonstrate the waveform configurations available through the staircase-step waveform builder. Configurable values within the waveform are shown in each case.

Before configuring the waveform builder, `function.waveform-builder.staircase.clear` (see 16.3.5) may be used to clear the waveform builder to the following default settings.

- Waveform is single-direction (“is bidirectional” is set false).
- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set to zero.
- Step distance is set to zero.
- Number of steps is set to 1.
- Hold duration is set to 0.1s.
- All other durations are set to zero.
- Trigger type is set to “none”.

16.3.2.1 Single-direction, not repeated



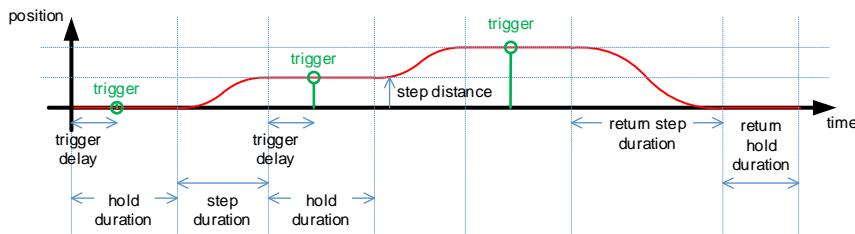
This shows a staircase waveform with the following configuration.

- Waveform is single-direction (“is bidirectional” is set false).
- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set non-zero.
- Number of steps is set to 3.

The first step takes place at the start position. Further steps move by the specified step distance for each step.

The number of steps must be at least 1. Attempting to set a value of zero will report “value out of range”.

16.3.2.2 Single-direction, start position zero

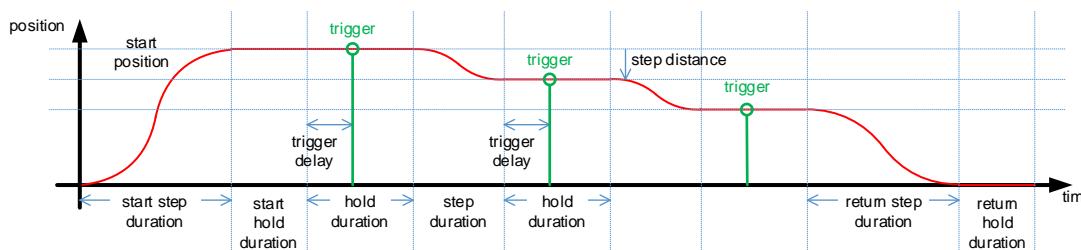


This shows a staircase waveform with the following configuration.

- Waveform is single-direction (“is bidirectional” is set false).
- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set to zero.
- Number of steps is set to 3.

When the start position is zero, the waveform builder omits the step and hold sections for moving to a start position because no initial movement is required.

16.3.2.3 Single-direction, negative-going staircase



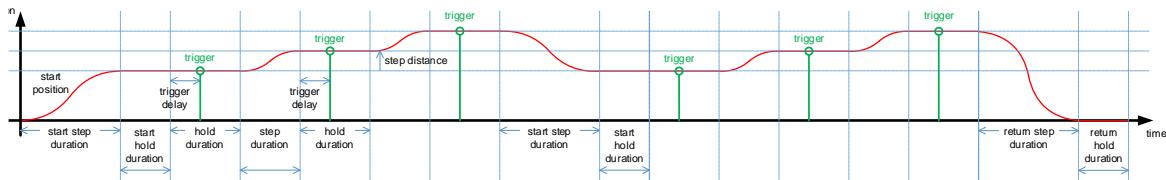
This shows a staircase waveform with the following configuration.

- Waveform is single-direction (“is bidirectional” is set false).

- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set to zero.
- Number of steps is set to 3.
- Step distance is set negative.

With a negative step distance, a negative-going staircase is produced.

16.3.2.4 Single-direction, repeated



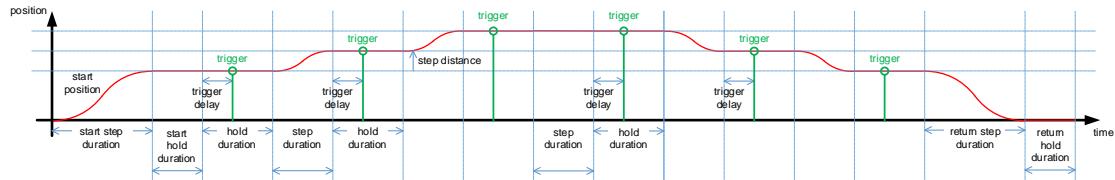
This shows a staircase waveform with the following configuration.

- Waveform is single-direction (“is bidirectional” set false).
- Waveform repeats (repeat count is set to 2).
- Start position is set non-zero.
- Number of steps is set to 3.

Note that the repeated waveform returns to the start position between repeats, not to zero.

As described in 16.3.2.2, if the start position is zero then the initial step and hold sections for moving to a start position are omitted. However the step and hold sections between repeats will always be present, to ensure the first acquisition on the next repeat is captured correctly.

16.3.2.5 Bidirectional, not repeated



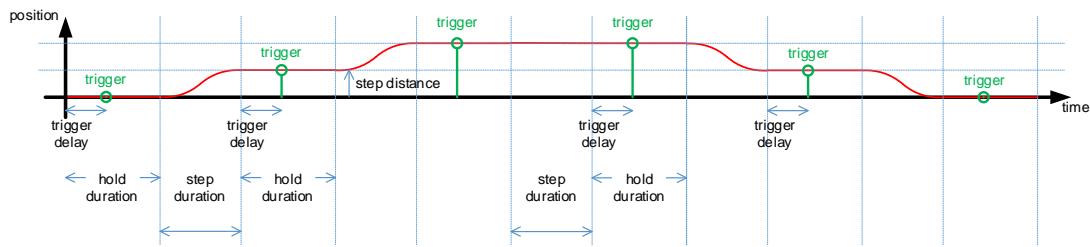
This shows a staircase waveform with the following configuration.

- Waveform is bidirectional (“is bidirectional” is set true).
- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set non-zero.
- Number of steps is set to 3.

This produces a symmetrical staircase in both directions. Two samples are taken at the top position to give the same number of sample acquisitions in both directions.

In order to give evenly-spaced acquisitions, the staircases in each direction are separated by an additional “step duration” time. This ensure the spacing between triggers is always the same, which is usually preferable when triggering connected electronics.

16.3.2.6 Bidirectional, start position zero

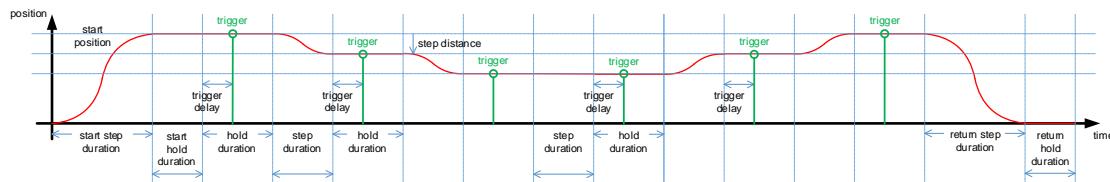


This shows a staircase waveform with the following configuration.

- Waveform is bidirectional (“is bidirectional” is set true).
- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set to zero.
- Number of steps is set to 3.

Similarly to the single-direction waveform (see 16.3.2.2), when the start position is zero, the waveform builder omits the step and hold sections for moving to a start position because no initial movement is required. For a bidirectional waveform, the waveform builder also omits the step and hold sections for returning to zero at the end for the same reason.

16.3.2.7 Bidirectional, negative-going first staircase

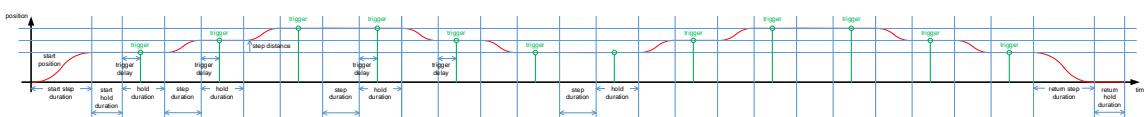


This shows a staircase waveform with the following configuration.

- Waveform is bidirectional (“is bidirectional” is set true).
- Waveform has a single pass without repeating (repeat count is set to 1).
- Start position is set non-zero.
- Number of steps is set to 3.
- Step distance is set negative.

With a negative step distance, the first staircase is negative-going. The symmetrical staircase in the opposite direction is therefore positive-going.

16.3.2.8 Bidirectional, repeated



This shows a staircase waveform with the following configuration.

- Waveform is bidirectional (“is bidirectional” is set true).
- Waveform repeats (repeat count is set to 2).
- Start position is set non-zero.
- Number of steps is set to 3.

As for a normal bidirectional staircase (see 16.3.2.5), the staircases in each direction are separated by an additional “step duration” time in order to give evenly-spaced acquisitions.

As described in 16.3.2.6, if the start position is zero then the step and hold sections for moving to a start position and returning to zero at the end are omitted.

16.3.3 Trigger outputs

For a staircase waveform, the direction of travel can naturally change at the start of each staircase. As a result, only the following trigger output event types described in 14.6.2 are permitted for the waveform builder.

Trigger output event type	Description
none	No trigger is set at this trigger position.
pulse	Trigger output is activated immediately. This is the “traditional” time-based triggering method.
in-position	Trigger output is activated when measured position is within the in-position error threshold (<code>stage.in-position.error-threshold</code> , see 8.5.1) of commanded position at trigger point. If measured position is already within this range, trigger output is activated immediately.

Level-triggered and edge-triggered types cannot be activated reliably here, so are not permitted. If there is a strong requirement for these to be used, consider using the waveform generator instead.

As shown in the waveforms in 16.3.2, the delay between the end of a step and triggering the output is configurable, to allow for system resonances to settle before acquiring data. The trigger delay may not be negative, and may not be greater than the hold duration.

16.3.4 Check and prepare waveform

The command `function.waveform-builder.staircase.prepare` (see 16.3.7) causes the waveform builder to generate the relevant waveform using the waveform generator (see 15) and then to prepare the sampled function playback waveform ready for use (see 14). If the waveform builder settings are valid, the command returns 1; if not, the command returns an error.

The following will result in an error.

- Trigger delay greater than hold duration.

All other invalid settings are reported immediately by the relevant command. See the limits and possible error codes for each command.

The waveform generator progress in preparing can be reported using `function.waveform-generator.prepare-waveform-status.get` (see 15.7.9) as normal. Note that if the “prepare” command does not report an error, preparing will always succeed. Once the waveform is prepared, the standard function playback features may be used to start and stop playback.

The command `function.waveform-builder.staircase.prepare-start` (see 16.3.8) may be used to prepare the waveform and immediately start after preparing, in the same way as the corresponding feature in the waveform generator.

The waveform builder settings can also be checked before preparing, using the command `function.waveform-builder.staircase.check` (see 16.3.6). This does not result in any change to the waveform generator or function playback waveforms.

16.3.5 function.waveform-builder.staircase.clear

16.3.5.1 Description

Clear staircase waveform builder settings

16.3.5.2 Command

```
function.waveform-builder.staircase.clear <channel>
```

16.3.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		1	1	Always returns 1 if clear can be carried out

16.3.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

16.3.5.6 Minimum security level

function.waveform-builder.staircase.clear	User level security required
---	------------------------------

16.3.5.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.6 function.waveform-builder.staircase.check

16.3.6.1 Description

Check staircase waveform builder settings

16.3.6.2 Command

```
function.waveform-builder.staircase.check <channel>
```

16.3.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	32-bit unsigned integer		1	1	Always returns 1 if check succeeds. If check fails, returns an error.

16.3.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	The waveform is not configured correctly. See 16.3.4 for details of configuration errors which may cause this.
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

16.3.6.6 Minimum security level

function.waveform-builder.staircase.check	User level security required
---	------------------------------

16.3.6.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.7 function.waveform-builder.staircase.prepare

16.3.7.1 Description

Prepare staircase waveform

16.3.7.2 Command

```
function.waveform-builder.staircase.prepare <channel>
```

16.3.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	32-bit unsigned integer		1	1	Always returns 1 if check succeeds. If check fails, returns an error.

16.3.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	The waveform is not configured correctly. See 16.3.4 for details of configuration errors which may cause this.
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

16.3.7.6 Minimum security level

function.waveform-builder.staircase.prepare	User level security required
---	------------------------------

16.3.7.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.8 function.waveform-builder.staircase.prepare-start

16.3.8.1 Description

Prepare staircase waveform and start after preparing

16.3.8.2 Command

```
function.waveform-builder.staircase.prepare-start <channel>
```

16.3.8.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	32-bit unsigned integer		1	1	Always returns 1 if check succeeds. If check fails, returns an error.

16.3.8.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	The waveform is not configured correctly. See 16.3.4 for details of configuration errors which may cause this.
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed

16.3.8.6 Minimum security level

function.waveform-builder.staircase.prepare-start	User level security required
---	------------------------------

16.3.8.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.9 function.waveform-builder.staircase.is-bidirectional.get/set**16.3.9.1 Description**

Staircase waveform builder whether staircase is bidirectional or single direction

16.3.9.2 Command

```
function.waveform-builder.staircase.is-bidirectional.get <channel>
function.waveform-builder.staircase.is-bidirectional.set <channel>
<value>
```

16.3.9.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.9.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer	Boolean	0	1	Whether staircase is bidirectional or single direction 0 = single direction 1 = bidirectional

16.3.9.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Whether staircase is bidirectional or single direction 0 = single direction 1 = bidirectional

16.3.9.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Boolean value must be 0 or 1 (set only)

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.9.7 Minimum security level

function.waveform-builder.staircase.is-bidirectional.get	No security required
function.waveform-builder.staircase.is-bidirectional.set	User level security required

16.3.9.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.10 function.waveform-builder.staircase.steps.get/set**16.3.10.1 Description**

Staircase waveform builder number of steps

16.3.10.2 Command

```
function.waveform-builder.staircase.steps.get <channel>  
function.waveform-builder.staircase.steps.set <channel> <value>
```

16.3.10.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.10.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer		1	10000	Number of steps

16.3.10.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer		1	10000	Number of steps

16.3.10.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Sample period must be greater than zero and less than maximum (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.10.7 Minimum security level

function.waveform-builder.staircase.steps.get	No security required
function.waveform-builder.staircase.steps.set	User level security required

16.3.10.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.11 function.waveform-builder.staircase.step-distance.get/set**16.3.11.1 Description**

Staircase waveform builder step distance

16.3.11.2 Command

```
function.waveform-builder.staircase.step-distance.get <channel>
```

```
function.waveform-builder.staircase.step-distance.set <channel>
<value>
```

16.3.11.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.11.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	pm			Step distance

16.3.11.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Step distance

16.3.11.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.11.7 Minimum security level

function.waveform-builder.staircase.step-distance.get	No security required
function.waveform-builder.staircase.step-distance.set	User level security required

16.3.11.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.12 function.waveform-builder.staircase.step-duration.get/set**16.3.12.1 Description**

Staircase waveform builder duration for step

16.3.12.2 Command

```
function.waveform-builder.staircase.step-duration.get <channel>
```

```
function.waveform-builder.staircase.step-duration.set <channel>  
<value>
```

16.3.12.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.12.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Duration for step

16.3.12.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Duration for step

16.3.12.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Duration must be zero or greater (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.12.7 Minimum security level

function.waveform-builder.staircase.step-duration.get	No security required
function.waveform-builder.staircase.step-duration.set	User level security required

16.3.12.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.13 function.waveform-builder.staircase.hold-duration.get/set**16.3.13.1 Description**

Staircase waveform builder duration to hold after each step

16.3.13.2 Command

```
function.waveform-builder.staircase.hold-duration.get <channel>
```

```
function.waveform-builder.staircase.hold-duration.set <channel>  
<value>
```

16.3.13.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.13.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	>0		Duration to hold after each step

16.3.13.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	>0		Duration to hold after each step

16.3.13.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Duration must be greater than zero (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.13.7 Minimum security level

function.waveform-builder.staircase.hold-duration.get	No security required
function.waveform-builder.staircase.hold-duration.set	User level security required

16.3.13.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.14 function.waveform-builder.staircase.start-position.get/set**16.3.14.1 Description**

Staircase waveform builder start position

16.3.14.2 Command

```
function.waveform-builder.staircase.start-position.get <channel>
```

```
function.waveform-builder.staircase.start-position.set <channel>
<value>
```

16.3.14.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.14.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	pm			Start position

16.3.14.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Start position

16.3.14.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.14.7 Minimum security level

function.waveform-builder.staircase.start-position.get	No security required
function.waveform-builder.staircase.start-position.set	User level security required

16.3.14.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.15 function.waveform-builder.staircase.start-step-duration.get/set**16.3.15.1 Description**

Staircase waveform builder duration to step to staircase start

16.3.15.2 Command

```
function.waveform-builder.staircase.start-step-duration.get <channel>
```

```
function.waveform-builder.staircase.start-step-duration.set <channel>
<value>
```

16.3.15.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.15.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Duration to step to staircase start

16.3.15.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Duration to step to staircase start

16.3.15.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Duration must be zero or greater (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.15.7 Minimum security level

function.waveform-builder.staircase.start-step-duration.get	No security required
function.waveform-builder.staircase.start-step-duration.set	User level security required

16.3.15.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.16 function.waveform-builder.staircase.start-hold-duration.get/set**16.3.16.1 Description**

Staircase waveform builder duration to hold after step to staircase start

16.3.16.2 Command

```
function.waveform-builder.staircase.start-hold-duration.get <channel>
```

```
function.waveform-builder.staircase.start-hold-duration.set <channel>
<value>
```

16.3.16.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.16.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Duration to hold after step to staircase start

16.3.16.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Duration to hold after step to staircase start

16.3.16.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Duration must be zero or greater (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.16.7 Minimum security level

function.waveform-builder.staircase.start-hold-duration.get	No security required
function.waveform-builder.staircase.start-hold-duration.set	User level security required

16.3.16.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.17 function.waveform-builder.staircase.return-step-duration.get/set**16.3.17.1 Description**

Staircase waveform builder duration to step back to initial position

16.3.17.2 Command

```
function.waveform-builder.staircase.return-step-duration.get <channel>
function.waveform-builder.staircase.return-step-duration.set <channel>
<value>
```

16.3.17.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.17.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Duration to step back to initial position

16.3.17.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Duration to step back to initial position

16.3.17.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Duration must be zero or greater (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.17.7 Minimum security level

function.waveform-builder.staircase.return-step-duration.get	No security required
function.waveform-builder.staircase.return-step-duration.set	User level security required

16.3.17.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.18 function.waveform-builder.staircase.return-hold-duration.get/set**16.3.18.1 Description**

Staircase waveform builder duration to hold after step back to initial position

16.3.18.2 Command

```
function.waveform-builder.staircase.return-hold-duration.get <channel>
function.waveform-builder.staircase.return-hold-duration.set <channel>
<value>
```

16.3.18.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.18.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Duration to hold after step back to initial position

16.3.18.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Duration to hold after step back to initial position

16.3.18.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Duration must be zero or greater (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.18.7 Minimum security level

function.waveform-builder.staircase.return-hold-duration.get	No security required
function.waveform-builder.staircase.return-hold-duration.set	User level security required

16.3.18.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.19 function.waveform-builder.staircase.repeat-count.get/set**16.3.19.1 Description**

Staircase waveform builder repeat count for staircase (0=continuous)

16.3.19.2 Command

```
function.waveform-builder.staircase.repeat-count.get <channel>
```

```
function.waveform-builder.staircase.repeat-count.set <channel> <value>
```

16.3.19.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.19.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit unsigned integer				Repeat count for staircase 0 = continuous 1 = single pass (no repeat) 2+ = repeat specified number of times

16.3.19.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Repeat count for staircase 0 = continuous 1 = single pass (no repeat) 2+ = repeat specified number of times

16.3.19.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.19.7 Minimum security level

function.waveform-builder.staircase.repeat-count.get	No security required
function.waveform-builder.staircase.repeat-count.set	User level security required

16.3.19.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.20 function.waveform-builder.staircase.trigger-output.trigger.get/set**16.3.20.1 Description**

Staircase waveform builder trigger output type after each step

16.3.20.2 Command

```
function.waveform-builder.staircase.trigger-output.trigger.get
<channel>
```

```
function.waveform-builder.staircase.trigger-output.trigger.set
<channel> <value>
```

16.3.20.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.20.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	String	Enumeration “trigger-output-event-type”			<p>Trigger output type after each step</p> <p>See 16.3.3 for available trigger event types</p>

16.3.20.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “trigger-output-event-type”			<p>Trigger output type after each step</p> <p>See 16.3.3 for available trigger event types</p>

16.3.20.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Invalid trigger type specified for staircase (set only). See 16.3.3 for available trigger event types.

Error return “errcode” string reported	Description
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.20.7 Minimum security level

function.waveform-builder.staircase.trigger-output.trigger.get	No security required
function.waveform-builder.staircase.trigger-output.trigger.set	User level security required

16.3.20.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

16.3.21 function.waveform-builder.staircase.trigger-output.delay.get/set**16.3.21.1 Description**

Staircase waveform builder trigger output delay time after step

16.3.21.2 Command

```
function.waveform-builder.staircase.trigger-output.delay.get <channel>
function.waveform-builder.staircase.trigger-output.delay.set <channel>
<value>
```

16.3.21.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel

16.3.21.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Function playback channel
value	32-bit floating-point	s	0		Trigger output delay time after step

16.3.21.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Trigger output delay time after step

16.3.21.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Delay must be zero or greater (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, or waveform generator prepare is in progress, so configuration cannot be changed (set only)

16.3.21.7 Minimum security level

function.waveform-builder.staircase.trigger-output.delay.get	No security required
function.waveform-builder.staircase.trigger-output.delay.set	User level security required

16.3.21.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

17 Snapshot capture

17.1 Snapshot configuration and control

The snapshot captures the measured positions for all channels for up to 10s at the controller sample rate. A 6000 series controller with a 50kHz internal control loop will therefore store up to 500,000 samples. The value `controller.sample-time` (see 5.1.10) reports the controller's internal control loop period, so that PC software can represent this data appropriately. Once started, the snapshot will run to completion unless stopped by the user.

Commands to start and stop snapshot work identically to the function playback commands or digital trigger inputs commanding snapshot start or stop (see 14.4 and 14.5), and can be integrated with command to start function playback in order for acquisition to be synchronised. Snapshots can also be triggered independently if function playback is not required.

A major use of the snapshot is to measure the system step response. To enable this without the complexity of setting up a function playback waveform, the snapshot includes a simple step generator. More complex waveforms may be created using function playback (see section 14), the waveform generator (see section 15) or the waveform builder (see section 16) as required.

A total of 4 snapshot channels are provided on all controllers, regardless of the number of physical channels present. By default, snapshot channels 1-3 capture measured position for stages on channels 1-3, where these physical channels are present on the controller. Where physical channels are not present (i.e. on a single-channel or two-channel controller), snapshot defaults to capturing measured position for the stage on channel 1. Snapshot captures the measured position to the control loop, as shown in the diagram in section 8.2.3.

If desired, the scope feature (see section 18) may be used to capture other signals instead. An additional snapshot channel 0 is available which is primarily intended for scope signal capture in addition to the 3 position channels. Alternatively any (or all) of the 4 snapshot channels may be assigned to capture scope signals instead of capturing measured position. Section 18 describes the scope feature in more detail.

17.1.1 snapshot.fire

17.1.1.1 Description

Start a snapshot capture

17.1.1.2 Command

snapshot.fire

17.1.1.3 Parameters

None

17.1.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Always returns 1

17.1.1.5 Possible error reports

None

17.1.1.6 Minimum security level

snapshot.fire	User level security required
---------------	------------------------------

17.1.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.1.2 snapshot.stop

17.1.2.1 Description

Stop a snapshot capture

17.1.2.2 Command

snapshot.stop

17.1.2.3 Parameters

None

17.1.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Always returns 0

17.1.2.5 Possible error reports

None

17.1.2.6 Minimum security level

snapshot.stop	User level security required
---------------	------------------------------

17.1.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.1.3 snapshot.capture.count.get/set

17.1.3.1 Description

Snapshot capture duration

17.1.3.2 Command

```
snapshot.capture.count.get  
snapshot.capture.count.set <value>
```

17.1.3.3 Parameters (get)

None

17.1.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 17.1	Capture duration

17.1.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 17.1	Capture duration

17.1.3.6 Possible error reports

Error return “errcode” string reported	Description
Value out of range	Capture duration must be within specified range (set only)

17.1.3.7 Minimum security level

snapshot.capture.count.get	No security required
snapshot.capture.count.get	User level security required

17.1.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.1.4 snapshot.trigger.to-target.get/set

17.1.4.1 Description

Time from start to start of snapshot commanded position step

17.1.4.2 Command

snapshot.trigger.to-target.get <channel>

snapshot.trigger.to-target.set <channel> <value>

17.1.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel

17.1.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel
value	32-bit unsigned integer	samples	0	See 17.1	Start of step

17.1.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 17.1	Start of step

17.1.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of snapshot channels
Value out of range	Start of step must be within specified range (set only)

17.1.4.7 Minimum security level

snapshot.trigger.to-target.get	No security required
snapshot.trigger.to-target.set	User level security required

17.1.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.1.5 snapshot.trigger.from-target.get/set

17.1.5.1 Description

Time from start to end of snapshot commanded position step

17.1.5.2 Command

```
snapshot.trigger.from-target.get <channel>  
snapshot.trigger.from-target.set <channel> <value>
```

17.1.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel

17.1.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel
value	32-bit unsigned integer	samples	0	See 17.1	End of step

17.1.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 17.1	End of step

17.1.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of snapshot channels
Value out of range	End of step must be within specified range (set only)

17.1.5.7 Minimum security level

snapshot.trigger.from-target.get	No security required
snapshot.trigger.from-target.set	User level security required

17.1.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.1.6 snapshot.trigger.step.get/set

17.1.6.1 Description

Snapshot commanded position step size

17.1.6.2 Command

```
snapshot.trigger.step.get <channel>  
snapshot.trigger.step.set <channel> <value>
```

17.1.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel

17.1.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel
value	32-bit floating-point	pm			Step size

17.1.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm			Step size

17.1.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of snapshot channels
Value out of range	Start of step must be within specified range (set only)

17.1.6.7 Minimum security level

snapshot.trigger.step.get	No security required
snapshot.trigger.step.set	User level security required

17.1.6.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.2 Snapshot results

17.2.1 snapshot.response.data.get

17.2.1.1 Description

Snapshot captured data

17.2.1.2 Command

```
snapshot.response.data.get <channel> <index>
```

17.2.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel
index	32-bit unsigned integer	samples	0	See 17.1	Sample index for data point

17.2.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point				Snapshot data

Where the snapshot channel reports measured position for a channel (i.e. default operation), units for snapshot data will be pm as for other commands reporting position.

Where the snapshot channel is used by the scope feature to report other data (see section 18), units will depend on the signal being monitored.

17.2.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of snapshot channels
Index out of range	Sample index must be within specified range

17.2.1.6 Minimum security level

snapshot.response.data.get	No security required
----------------------------	----------------------

17.2.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.2.2 snapshot.response.count.get

17.2.2.1 Description

Snapshot captured data duration

17.2.2.2 Command

snapshot.response.count.get

17.2.2.3 Parameters

None

17.2.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 17.1	Capture duration

17.2.2.5 Possible error reports

None

17.2.2.6 Minimum security level

snapshot.response.count.get	No security required
-----------------------------	----------------------

17.2.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

17.2.3 snapshot.response.data-select.get

17.2.3.1 Description

Snapshot captured data selected

17.2.3.2 Command

```
snapshot.response.data-select.get <channel>
```

17.2.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Snapshot channel

17.2.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
data-selection	16-bit unsigned integer	enum			Scope data selected
physical-channel	8-bit unsigned integer		0	3	Physical channel

Where the snapshot channel reports measured position for a channel (i.e. default operation), data-selection will be zero and physical-channel will be the channel number.

Where the snapshot channel is used by the scope feature to report other data (see section 18), these will specify the signal being monitored.

17.2.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of snapshot channels

17.2.3.6 Minimum security level

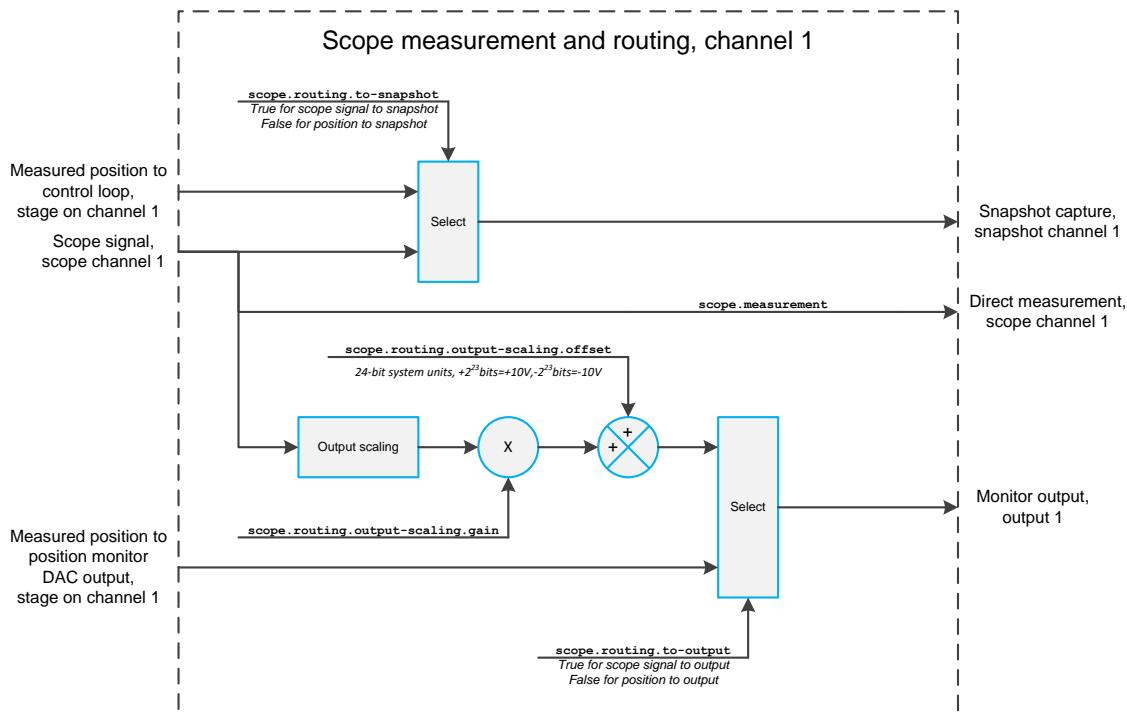
snapshot.response.data.get	No security required
----------------------------	----------------------

17.2.3.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18 Scope monitoring of internal data

18.1 Scope measurement and routing



Users may need access to internal signals for calibration and monitoring purposes. It is frequently useful to be able to directly observe the response of a filter, for example, or to observe how the commanded velocity and acceleration are controlled by the feedforward command trajectory limiting. As with snapshots (see section 17), there are four channels of scope measurement, numbered 0-3. The number of scope channels is not reduced for controllers with fewer physical channels.

As the diagram above shows, the signal for scope channel 1 may be routed to snapshot channel 1 and/or to monitor output 1, as well as reported directly on request. If the scope signal is not routed to the snapshot or monitor output, the measured position for the stage on channel 1 will be routed to that snapshot and monitor output instead. Similarly for scope channels 2 and 3, these may be routed to snapshot channels 2 and 3 and/or to monitor outputs 2 and 3 respectively.

Scope channel 0 is not associated with a physical channel for measured position or monitor output. The same routing options exist, but routing to monitor output has no effect. The same applies for channels 2 and 3 on single-channel controllers, or channel 3 on dual-channel controllers, where the corresponding physical channel does not exist.

Scope signals are identified by an ID number and a physical channel number. Each scope signal may come from any physical channel on the controller; scope channel 1 could be assigned to report unfiltered measured position for the stage on channel 3, for example. The ID number for each signal is shown in the diagrams in section 8 in the format 123; to continue the previous example, the unfiltered measured position is shown as ID 313 in diagram 8.2.3. See the list in 18.2 for full details of scope signals available.

The stage datasheet specifies the standard scaling from measured position (in picometres/nanometres/microns) to volts. Where a scope signal reports a position and that

signal is routed to the monitor output, this scaling is applied in all cases. Other scope signals may report values in signed 24-bit system units (i.e. a range +/- 2^{23}), which are sent to the output scaled as +/- 2^{23} bits = +/-10V on the monitor output.

Particularly small or large signals may require further scaling in order to be easily observable on the monitor output. A gain and offset may be applied to the scope signal if required.

The original intention of the scope functionality was to facilitate stage calibration, but customers are free to use this for other purposes as required. For example:-

- For some applications, it may be more appropriate for the position setpoint to be available on the monitor output instead of the measured position.
- Customers may wish to disable the analogue input as a position command source and inject an unrelated signal which can be read by the host PC, avoiding the need for an extra ADC in the system.
- Function playback may require an extra output signal synchronised to the function, for example generating a sine wave as the stage moves. Function playback channel 0 can be used to generate this extra waveform (see 14.1), and scope channel routing allows this to be sent to a monitor output.

Note that since this is primarily intended as a debugging and calibration tool, scope routing selection is not saved on power-off. If scope routing is to be used as a feature of the control system, the host PC must configure the required routing options every time the controller powers on.

18.1.1 scope.data-select.get/set**18.1.1.1 Description**

Scope data signal selected

18.1.1.2 Command

```
scope.data-select.get <channel>
```

```
scope.data-select.set <channel> <data-selection> <physical-channel>
```

18.1.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel

18.1.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel
data-selection	16-bit unsigned integer	enum			Scope signal ID in control system
physical-channel	8-bit unsigned integer		0	3	Stage channel for scope signal

18.1.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
data-selection	16-bit unsigned integer	enum			Scope signal ID in control system
physical-channel	8-bit unsigned integer				Stage channel for scope signal

18.1.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of scope channels
Value out of range	Physical channel parameter is larger than number of controller channels, or data selection is not a valid signal ID

18.1.1.7 Minimum security level

scope.data-select.get	No security required
scope.data-select.set	User level security required

18.1.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18.1.2 scope.measurement.get

18.1.2.1 Description

Current value of scope data signal

18.1.2.2 Command

scope.measurement.get <channel>

18.1.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel

18.1.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point				Scope data signal

18.1.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of scope channels

18.1.2.6 Minimum security level

scope.measurement.get	No security required
-----------------------	----------------------

18.1.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18.1.3 scope.routing.to-snapshot.get/set

18.1.3.1 Description

Scope data signal routed to snapshot

18.1.3.2 Command

```
scope.routing.to-snapshot.get <channel>  
scope.routing.to-snapshot.set <channel> <value>
```

18.1.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel

18.1.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel
value	32-bit unsigned integer	Boolean	0	1	Scope channel routed to snapshot

18.1.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Scope channel routed to snapshot

18.1.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of scope channels

18.1.3.7 Minimum security level

scope.routing.to-snapshot.get	No security required
scope.routing.to-snapshot.set	User level security required

18.1.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18.1.4 scope.routing.to-output.get/set

18.1.4.1 Description

Scope data signal routed to monitor output

18.1.4.2 Command

```
scope.routing.to-output.get <channel>  
scope.routing.to-output.set <channel> <value>
```

18.1.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel

18.1.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel
value	32-bit unsigned integer	Boolean	0	1	Scope channel routed to monitor output

18.1.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Scope channel routed to monitor output

18.1.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of scope channels

18.1.4.7 Minimum security level

scope.routing.to-output.get	No security required
scope.routing.to-output.set	User level security required

18.1.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18.1.5 scope.routing.output-scaling.gain.get/set

18.1.5.1 Description

Gain applied to scope data signal when routed to monitor output

18.1.5.2 Command

scope.routing.output-scaling.gain.get <channel>

scope.routing.output-scaling.gain.set <channel> <value>

18.1.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel

18.1.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel
value	32-bit floating-point				Monitor output gain

18.1.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point				Monitor output gain

18.1.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of scope channels

18.1.5.7 Minimum security level

scope.routing.output-scaling.gain.get	No security required
scope.routing.output-scaling.gain.set	User level security required

18.1.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18.1.6 scope.routing.output-scaling.offset.get/set**18.1.6.1 Description**

Offset applied to scope data signal when routed to monitor output

18.1.6.2 Command

```
scope.routing.output-scaling.offset.get <channel>
```

```
scope.routing.output-scaling.offset.set <channel> <value>
```

18.1.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel

18.1.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Scope channel
value	32-bit floating-point	24-bit DAC bits	-16777215	16777215	Monitor output offset

18.1.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	24-bit DAC bits	-16777215	16777215	Monitor output offset

18.1.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of scope channels

18.1.6.7 Minimum security level

scope.routing.output-scaling.offset.get	No security required
scope.routing.output-scaling.offset.set	User level security required

18.1.6.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

18.2 Scope data signal selection

The following scope data signals are available for each physical channel, with the specified scalings when signals are routed to monitor outputs. Where the scaling is listed as “position scaling for stage”, the scaling from microns to voltage on the monitor output is as specified in the stage datasheet.

Signal ID	Signal description	Diagram or reference	Output scaling
0	Measured position to control loop and user <code>stage.position.measured</code>	8.2.1, 8.2.3, 8.2.4, 8.2.5, 8.2.7, 8.2.9, 8.2.10, 8.2.11, 8.2.12, 8.4.1	Position scaling for stage
2	Measured position to position monitor DAC output	8.2.3	Position scaling for stage
101	Analogue input	12.1.2, 12.1.3	Voltage
111	Channel heatsink temperature <code>protection.thermal.heatsink.temperature</code>	19.1.1, 19.3.2	$\pm 10V = \pm 200^\circ C$
112	Internal power supply temperature <code>protection.thermal.psu.temperature</code>	19.1.1, 19.3.7	$\pm 10V = \pm 200^\circ C$
120	Function playback trigger input	14.5	+5V=true, 0V=false
122	Stepped input A	13.1	+5V=true, 0V=false
123	Stepped input B	13.1	+5V=true, 0V=false
124	Stepped input steps received each control loop period	13.1	+5V=one or more forward steps received 0V=no steps received -5V=one or more backward steps received
130	Function playback trigger output	15.6	+5V=true, 0V=false
131	“In position” output	8.2.1, 8.2.4	+5V=true, 0V=false

Signal ID	Signal description	Diagram or reference	Output scaling
134	Stepped output steps sent each control loop period	13.3	+5V=one or more forward steps sent 0V=no steps sent -5V=one or more backward steps sent
137	Servo output at limits <code>stage.status.servo-output-at-limits.get</code>	8.2.1, 8.2.4, 8.2.15, 8.12.2	+5V=true,0V=false
138	Stage is currently moving <code>stage.status.stage-moving.get</code>	8.2.4, 8.5.9	+5V=true,0V=false
300	Analogue position command	8.2.2	Position scaling for stage
301	Combined digital command	8.2.2	Position scaling for stage
302	Snapshot step command	8.2.1, 8.2.2, 17	Position scaling for stage
303	Combined position setpoint to control loop	8.2.1, 8.2.2, 8.2.4, 8.2.5, 8.2.6	Position scaling for stage
304	Analogue input command	8.2.1, 8.2.2, 12.1.2, 12.1.3	Position scaling for stage
307	Stepped position command <code>stage.position.stepped-command</code>	8.2.1, 8.2.2, 13.1, 13.2, 13.2.1	Position scaling for stage
308	Absolute position command <code>stage.position.absolute-command</code>	8.2.1, 8.2.2, 8.3.1	Position scaling for stage
310	Analogue position command edge-boost filtered	8.2.2	Position scaling for stage
311	Digital position command edge-boost filtered	8.2.2	Position scaling for stage
313	Measured position scaled and linearised	8.2.3	Position scaling for stage
314	Function playback command	8.2.1, 8.2.2, 13	Position scaling for stage

Signal ID	Signal description	Diagram or reference	Output scaling
350	Digital position command <code>stage.position.command</code>	8.2.1, 8.2.2, 8.3.4	Position scaling for stage
400	Closed-loop control output	8.2.5, 8.2.9, 8.2.10, 8.2.11, 8.2.12	+10V=full-scale positive HV -10V=full-scale negative HV
410	Open-loop control output	8.2.5, 8.2.8	+10V=full-scale positive HV -10V=full-scale negative HV
420	Closed-loop velocity control output	8.2.11, 8.2.12	+10V=full-scale positive HV -10V=full-scale negative HV
421	Commanded velocity	8.2.7, 8.2.9, 8.2.11, 8.2.12	Indicative only; no fixed scaling
422	Measured velocity	8.2.7, 8.2.9, 8.2.10, 8.2.11, 8.2.12, 8.2.13, 8.2.14	Indicative only; no fixed scaling
423	Commanded velocity plus position control commanded velocity	8.2.11, 8.2.12,	Indicative only; no fixed scaling
430	Closed-loop acceleration control output	8.2.12	Indicative only; no fixed scaling
431	Commanded acceleration	8.2.7, 8.2.9, 8.2.12	Indicative only; no fixed scaling
432	Measured acceleration	8.2.7, 8.2.9, 8.2.10, 8.2.11, 8.2.12, 8.2.13, 8.2.14	Indicative only; no fixed scaling
433	Commanded acceleration plus velocity control commanded acceleration	8.2.12	Indicative only; no fixed scaling
450	Combined command after command trajectory limiting (if selected)	8.2.5, 8.2.6, 8.2.7, 8.2.8, 8.2.9, 8.2.10, 8.2.11, 8.2.12	Position scaling for stage

Signal ID	Signal description	Diagram or reference	Output scaling
451	Command velocity after command trajectory limiting	8.2.6	Position scaling for stage per ms
600	Servo loop DAC output	8.2.1, 8.2.11	+10V=full-scale positive HV -10V=full-scale negative HV
601	“In position” error after LPF smoothing	8.2.4	Position scaling for stage
602	“In position” confirmed flag, using LPF error smoothing for debouncing <code>stage.status.in-position.lpf-confirmed.get</code>	8.2.4, 8.5.6	+5V=true,0V=false
603	“In position” unconfirmed flag <code>stage.status.in-position.unconfirmed.get</code>	8.2.4, 8.5.5	+5V=true,0V=false
604	“In position” confirmed flag, using window filter for debouncing <code>stage.status.in-position.window-filter-confirmed.get</code>	8.2.4, 8.5.7	+5V=true,0V=false
605	“In position” error	8.2.4	Position scaling for stage
606	“In position” window filter depth	8.2.4	Proportion of window filter depth full 0V=window filter empty +2.5V=window filter half full +5V=window filter full
620	Control loop output before damping	8.2.5, 8.2.13	+10V=full-scale positive HV -10V=full-scale negative HV
621	Control loop output after filtering	8.2.13, 8.2.14	+10V=full-scale positive HV -10V=full-scale negative HV
622	Control loop output after damping	8.2.1, 8.2.5, 8.2.13, 8.2.14, 8.2.15	+10V=full-scale positive HV -10V=full-scale negative HV

19 Controller protection

19.1 Overview

The controller firmware contains a range of protective measures which ensure the controller cannot become damaged due to common operating conditions or stage failures.

19.1.1 Automatic fan control

The controller has cooling fans which ensure internal temperatures are not excessive. Passive cooling is sufficient when stages are not being driven fast, so the controller is normally configured to only turn the fans on when temperatures become elevated. The temperatures at which the fans turn on and off may be configured by the user if required. These may be set individually for each controller channel and for the high voltage power supply.

The acoustic noise from fans may be unacceptable in some applications though, so the user has the option to fully disable the cooling fans if required. The user must ensure the controller is not required to move the stage(s) rapidly or continuously in this case, and ensure that the controller is operated in an environment which is not excessively warm. If operating conditions change and the controller is inadvertently caused to overheat, protection will temporarily shut down the controller to attempt to prevent damage (see 19.1.2).

Conversely, some systems may be operated with high loads/velocities for long periods of time. In these cases the user has the option to permanently enable the cooling fans to ensure the controller continues to operate at a lower temperature, which may give improved lifespan.

NOTE: Disabling cooling fans and changing the temperatures at which fans turn on/off are carried out at the user's risk. Whilst controller thermal protection will attempt to prevent system damage in the event of overheating, repeated overheating may reduce the controller lifespan. If cooling fans are disabled whilst the system is operated continuously with high loads or high velocities, and any resulting overheating causes controller damage, Queensgate will consider this to be intentional or negligent damage caused by the user. This may invalidate the controller warranty.

Commands to configure automatic fan control are described in 19.2.

19.1.2 Controller thermal (overtemperature) protection

The user may disable the cooling fans if required (see 19.1.1). If the controller is driven continuously with high load and high velocities for a long period without cooling, the controller electronics may heat up to an unacceptable level which could result in damage. Note that the cooling fans will prevent this occurring in normal use.

The controller monitors internal temperatures for each channel's HV electronics and for the controller power supply. If these exceed safe levels, the firmware will immediately carry out a protective shutdown of the controller. When all temperatures have returned to a safe operating range, the protective shutdown is cleared and the controller will automatically restart.

Commands to configure thermal protection are described in 19.3.

19.1.3 Stage short-circuit (overcurrent) protection

The piezo actuators in a stage may become damaged and fail short-circuit. Damage to stage wiring may also cause short-circuit faults. The controller electronics are protected against short-circuit faults and limit the HV current output, but long-term faults could cause the controller to overheat.

Whilst the controller has protection against overheating (see 19.1.2), long-term operation at high temperatures can reduce the controller lifespan, so it is preferable to avoid this. The controller firmware monitors HV output currents for each channel, and carries out a protective shutdown of the controller if any channel persistently draws high current for a long period (which is not possible for any normal stage movement). If the faulty stage is disconnected, the protective shutdown is cleared and the controller will automatically restart.

Commands to configure overcurrent protection are described in 19.4.

19.1.4 Controller actions during and after protective shutdown

During protective shutdown, the HV outputs for all channels are shut down for safety. The front panel LEDs for all channels flash red to alert the user.

When a protective shutdown clears, the controller restarts. All connected stages will have their default configuration loaded and will restart as if the controller had just been powered on.

Any stage configuration changed before the protective shutdown will be lost. This is a safety measure, because it is possible that a channel may overheat due to a misconfiguration causing the stage to resonate. Returning to the default configuration should ensure that the controller resumes operation in a safe state.

19.2 Automatic fan control

19.2.1 protection.fan.mode.get/set

19.2.1.1 Description

Fan mode.

When this is set, the setting is stored to the controller and is retrieved at next power-on.

19.2.1.2 Command

protection.fan.mode.get

protection.fan.mode.set <value>

19.2.1.3 Parameters (get)

None

19.2.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “fan-mode”			Fan mode “off” : Fan permanently off “on”: Fan permanently on “auto”: Fan turned on and off automatically based on temperature

19.2.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	String	Enumeration “fan-mode”			Fan mode “off” : Fan permanently off “on”: Fan permanently on “auto”: Fan turned on and off automatically based on temperature

19.2.1.6 Possible error reports

None

19.2.1.7 Minimum security level

protection.fan.mode.get	No security required
protection.fan.mode.set	Superuser level security required

19.2.1.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

19.2.2 protection.fan.state.get

19.2.2.1 Description

Fan state (on/off)

19.2.2.2 Command

protection.fan.state.get

19.2.2.3 Parameters

None

19.2.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Fan state 0 = off 1 = on

19.2.2.5 Possible error reports

None

19.2.2.6 Minimum security level

protection.fan.state.get	No security required
--------------------------	----------------------

19.2.2.7 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

19.2.3 protection.fan.psu.on-temperature.get/set

19.2.3.1 Description

In auto fan mode, PSU temperature at which fan turns on.

When this is set, the setting is stored to the controller and is retrieved at next power-on.

19.2.3.2 Command

protection.fan.psu.on-temperature.get

protection.fan.psu.on-temperature.set <value>

19.2.3.3 Parameters (get)

None

19.2.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			PSU temperature at which fan turns on

19.2.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			PSU temperature at which fan turns on

19.2.3.6 Possible error reports

None

19.2.3.7 Minimum security level

protection.fan.psu.on-temperature.get	No security required
protection.fan.psu.on-temperature.set	Superuser level security required

19.2.3.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

19.2.4 protection.fan.psu.off-temperature.get/set

19.2.4.1 Description

In auto fan mode, PSU temperature at which fan turns off.

When this is set, the setting is stored to the controller and is retrieved at next power-on.

19.2.4.2 Command

```
protection.fan.psu.off-temperature.get
```

```
protection.fan.psu.off-temperature.set <value>
```

19.2.4.3 Parameters (get)

None

19.2.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			PSU temperature at which fan turns off

19.2.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			PSU temperature at which fan turns ooff

19.2.4.6 Possible error reports

None

19.2.4.7 Minimum security level

protection.fan.psu.off-temperature.get	No security required
protection.fan.psu.off-temperature.set	Superuser level security required

19.2.4.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

19.2.5 protection.fan.heatsink.on-temperature.get/set

19.2.5.1 Description

In auto fan mode, heatsink temperature at which fan turns on.

When this is set, the setting is stored to the controller and is retrieved at next power-on.

19.2.5.2 Command

```
protection.fan.heatsink.on-temperature.get  
protection.fan.heatsink.on-temperature.set <value>
```

19.2.5.3 Parameters (get)

None

19.2.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			Heatsink temperature at which fan turns on

19.2.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			Heatsink temperature at which fan turns on

19.2.5.6 Possible error reports

None

19.2.5.7 Minimum security level

protection.fan.heatsink.on-temperature.get	No security required
protection.fan.heatsink.on-temperature.set	Superuser level security required

19.2.5.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

19.2.6 protection.fan.heatsink.off-temperature.get/set

19.2.6.1 Description

In auto fan mode, heatsink temperature at which fan turns off.

When this is set, the setting is stored to the controller and is retrieved at next power-on.

19.2.6.2 Command

```
protection.fan.heatsink.off-temperature.get
```

```
protection.fan.heatsink.off-temperature.set <value>
```

19.2.6.3 Parameters (get)

None

19.2.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			Heatsink temperature at which fan turns off

19.2.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	°C			Heatsink temperature at which fan turns ooff

19.2.6.6 Possible error reports

None

19.2.6.7 Minimum security level

protection.fan.heatsink.off-temperature.get	No security required
protection.fan.heatsink.off-temperature.set	Superuser level security required

19.2.6.8 Supported in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

19.3 Controller thermal (overtemperature) protection

19.3.1 protection.thermal.status.get

19.3.1.1 Description

Status of thermal cutout protection

19.3.1.2 Command

protection.thermal.status.get

19.3.1.3 Parameters

None

19.3.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Bitfield			Bit 0: Protection shutdown triggered by channel 1 heatsink overtemperature Bit 1: Protection shutdown triggered by channel 2 heatsink overtemperature Bit 2: Protection shutdown triggered by channel 3 heatsink overtemperature Bit 7: Protection shutdown triggered by internal power supply overtemperature

19.3.1.5 Possible error reports

None

19.3.1.6 Minimum security level

protection.thermal.status.get	No security required
-------------------------------	----------------------

19.3.1.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.2 protection.thermal.heatsink.temperature.get

19.3.2.1 Description

Heatsink temperature

19.3.2.2 Command

```
protection.thermal.heatsink.temperature.get <channel>
```

19.3.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.3.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Heatsink temperature

19.3.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.3.2.6 Minimum security level

protection.thermal.heatsink.temperature.get	No security required
---	----------------------

19.3.2.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.3 protection.thermal.heatsink.overtemperature-detect-threshold.get**19.3.3.1 Description**

Heatsink overtemperature detect threshold

19.3.3.2 Command

```
protection.thermal.heatsink.overtemperature-detect-threshold.get
<channel>
```

19.3.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.3.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Detect threshold

19.3.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.3.3.6 Minimum security level

protection.thermal.heatsink.overtemperature-detect-threshold.get	No security required
--	----------------------

19.3.3.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.4 protection.thermal.heatsink.overtemperature-detect-time.get**19.3.4.1 Description**

Heatsink overtemperature detect time before cutout

19.3.4.2 Command

```
protection.thermal.heatsink.overtemperature-detect-time.get <channel>
```

19.3.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.3.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	s	0		Detect time before cutout

19.3.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.3.4.6 Minimum security level

protection.thermal.heatsink.overtemperature-detect-time.get	No security required
---	----------------------

19.3.4.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.5 protection.thermal.heatsink.overtemperature-clear-threshold.get

19.3.5.1 Description

Heatsink overtemperature clear threshold

19.3.5.2 Command

```
protection.thermal.heatsink.overtemperature-clear-threshold.get  
<channel>
```

19.3.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.3.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Clear threshold

19.3.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.3.5.6 Minimum security level

protection.thermal.heatsink.overtemperature-clear-threshold.get	No security required
---	----------------------

19.3.5.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.6 protection.thermal.heatsink.overtemperature-clear-time.get**19.3.6.1 Description**

Heatsink overtemperature detect threshold

19.3.6.2 Command

```
protection.thermal.heatsink.overtemperature-clear-time.get <channel>
```

19.3.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.3.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Clear time before restart

19.3.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.3.6.6 Minimum security level

protection.thermal.heatsink.overtemperature-clear-time.get	No security required
--	----------------------

19.3.6.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.7 protection.thermal.psu.temperature.get

19.3.7.1 Description

Internal power supply temperature

19.3.7.2 Command

protection.thermal.psu.temperature.get

19.3.7.3 Parameters

None

19.3.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Power supply temperature

19.3.7.5 Possible error reports

None

19.3.7.6 Minimum security level

protection.thermal.psu.temperature.get	No security required
--	----------------------

19.3.7.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.8 protection.thermal.psu.overtemperature-detect-threshold.get**19.3.8.1 Description**

Internal power supply overtemperature detect threshold

19.3.8.2 Command

protection.thermal.psu.overtemperature-detect-threshold.get

19.3.8.3 Parameters

None

19.3.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Detect threshold

19.3.8.5 Possible error reports

None

19.3.8.6 Minimum security level

protection.thermal.psu.overtemperature-detect-threshold.get	No security required
---	----------------------

19.3.8.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.9 protection.thermal.psu.overtemperature-detect-time.get**19.3.9.1 Description**

Internal power supply overtemperature detect time before cutout

19.3.9.2 Command

protection.thermal.psu.overtemperature-detect-time.get

19.3.9.3 Parameters

None

19.3.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	s	0		Detect time before cutout

19.3.9.5 Possible error reports

None

19.3.9.6 Minimum security level

protection.thermal.psu.overtemperature-detect-time.get	No security required
--	----------------------

19.3.9.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.10 protection.thermal.psu.overtemperature-clear-threshold.get**19.3.10.1 Description**

Internal power supply overtemperature clear threshold

19.3.10.2 Command

protection.thermal.psu.overtemperature-clear-threshold.get

19.3.10.3 Parameters

None

19.3.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Clear threshold

19.3.10.5 Possible error reports

None

19.3.10.6 Minimum security level

protection.thermal.psu.overtemperature-clear-threshold.get	No security required
--	----------------------

19.3.10.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.3.11 protection.thermal.psu.overtemperature-clear-time.get**19.3.11.1 Description**

Internal power supply overtemperature detect threshold

19.3.11.2 Command

protection.thermal.psu.overtemperature-clear-time.get

19.3.11.3 Parameters

None

19.3.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Clear time before restart

19.3.11.5 Possible error reports

None

19.3.11.6 Minimum security level

protection.thermal.psu.overtemperature-clear-time.get	No security required
---	----------------------

19.3.11.7 Supported in

Controller application firmware	6.2.8 onwards
Controller interface library	2.2.4 onwards

19.4 Stage short-circuit (overcurrent) protection

19.4.1 protection.overcurrent.status.get

19.4.1.1 Description

Status of overcurrent cutout protection

19.4.1.2 Command

```
protection.overcurrent.status.get
```

19.4.1.3 Parameters

None

19.4.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Bitfield			<p>Bit 0: Protection shutdown triggered by channel 1 overcurrent</p> <p>Bit 1: Protection shutdown triggered by channel 2 overcurrent</p> <p>Bit 2: Protection shutdown triggered by channel 3 overcurrent</p>

19.4.1.5 Possible error reports

None

19.4.1.6 Minimum security level

protection.overcurrent.status.get	No security required
-----------------------------------	----------------------

19.4.1.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.20 onwards

19.4.2 protection.overcurrent.current.get

19.4.2.1 Description

Measured HV current

19.4.2.2 Command

```
protection.overcurrent.current.get <channel>
```

19.4.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.4.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	A			Measured HV current

19.4.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.4.2.6 Minimum security level

protection.overcurrent.current.get	No security required
------------------------------------	----------------------

19.4.2.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.20 onwards

19.4.3 protection.overcurrent.detect-threshold.get

19.4.3.1 Description

Overcurrent detect threshold

19.4.3.2 Command

```
protection.overcurrent.heatsink.detect-threshold.get <channel>
```

19.4.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.4.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	A	0		Detect threshold

19.4.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.4.3.6 Minimum security level

protection.overcurrent.detect-threshold.get	No security required
---	----------------------

19.4.3.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.20 onwards

19.4.4 protection.overcurrent.detect-time.get

19.4.4.1 Description

Overcurrent detect time before cutout

19.4.4.2 Command

```
protection.overcurrent.heatsink.overtemperature-detect-time.get  
<channel>
```

19.4.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

19.4.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	s	0		Detect time before cutout

19.4.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller

19.4.4.6 Minimum security level

protection.overcurrent.detect-time.get	No security required
--	----------------------

19.4.4.7 Supported in

Controller application firmware	6.6.22 onwards
Controller interface library	2.6.20 onwards

20 Controller synchronisation

20.1 Controller sensor synchronisation

In order to measure the position of a stage, AC signals must be sent down the stage cable to the sensor. The returning signals from the sensor are measured to determine the sensor capacitance, and hence the measured stage position (as reported by command `stage.position.measured.get`, see 8.4.1). Measurement of the returning signals is carefully timed, because the rising and falling edges of the transmitted AC signal inevitably create some electrical noise. This is very small, but can be significant at metrology precisions.

When two or more controllers or stages are placed close together, the AC signals sent by one controller may cause some noise to appear on the returning signals for another controller's stages. This is clearly undesirable. By connecting appropriate cables (supplied by Queensgate) between the "SYNC IN/OUT" ports of all controllers, AC signal timings for all controllers are synchronised so that this does not happen.

Sync cables are connected from the "out" connection of one controller to the "in" connection of the next. One controller will have no "in" connection, and is considered the "master" of the chain. As many controllers can be synchronised together as required.

20.1.1 controller.synchronisation.master.get

20.1.1.1 Description

Controller is synchronisation master, or controller synchronisation cable is not connected

20.1.1.2 Command

controller.synchronisation.master.get

20.1.1.3 Parameters

None

20.1.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Controller is synchronisation master

20.1.1.5 Possible error reports

None

20.1.1.6 Minimum security level

controller.synchronisation.master.get	No security required
---------------------------------------	----------------------

20.1.1.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

20.1.2 controller.synchronisation.slave.get

20.1.2.1 Description

Controller is synchronisation slave and is locked to the master

20.1.2.2 Command

controller.synchronisation.slave.get

20.1.2.3 Parameters

None

20.1.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Controller is synchronisation slave

20.1.2.5 Possible error reports

None

20.1.2.6 Minimum security level

controller.synchronisation.slave.get	No security required
--------------------------------------	----------------------

20.1.2.7 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

21 Controller operation with sensor-only systems

21.1 Capacitive sensor systems

As well as nanopositioning stages, Queensgate also offer a range of capacitive sensors for contactless measurement of nanometre-precision displacements. These may be ordered for use with an NPC-D-6000 series controller, where the controller is also to control stages, or for an NS-D-6000 series sensor controller. The NPC-D-6000 and NS-D-6000 digital controllers offer additional features which are not available to analogue sensor controllers, such as plug-and-play functionality due to the stage calibration being contained within the stage EEPROM (see 7), control system features such as linearisation and filtering (see diagram 8.2.3 and 8.4), and a convenient range of interfacing options.

Since this is only a sensor and has no actuator, the rest of the control system for a sensor channel is non-functional. Commands relating to closed-loop control will report an error “Command not valid for sensor” when used with this channel, as described in 3.2.2.

Sensors may be used with standard NPC-D-6000 controllers without issues. For multi-channel controllers, the same controller may be used for both stages and sensors. Channels used for sensors will have their HV piezo actuator outputs disabled.

NS-D-6000 sensor controllers share much of the same platform, but omit the HV supplies and piezo actuator outputs. NS-D-6000 controllers can only be used with sensors and cannot drive stages.

21.2 Front-panel LEDs for sensor systems

The front panel “closed/open loop” LED on NPC-D-6000 controllers is illuminated yellow for a channel connected to a sensor, differentiating it from a stage which would normally show green for closed-loop control or red for open-loop control.

The equivalent lower LED on NS-D-6000 controllers is simply illuminated green when a sensor is connected.

For both NPC-D-6000 and NS-D-6000 controllers, the “in-position” LED is used for sensors to indicate whether the measured position is within acceptable ranges, instead of its normal functionality for stages.

- If the measured position is within the nominal calibrated range of the sensor, the “in-position” LED is illuminated green.
- If the measured position is outside the nominal calibrated range of the sensor, but the sensor is still capable of measuring the position, the “in-position” LED is illuminated yellow.
- If the sensor position cannot be measured (because the sensor plates are either touching or too far apart), the “in-position” LED is illuminated red.

21.3 Capacitive sensor startup

The controller must carry out some auto-calibration on sensors when a stage/sensor is first connected. For a stage, both sensor plates are contained within the stage assembly. For a

sensor system though, it is possible that the sensor may be connected with the sensor plates disconnected, not assembled with the correct spacing, or shorted together.

During normal startup for stages or sensors, both front-panel LEDs for the channel are initially amber. If the controller finds that the sensor position cannot be measured (because the sensor plates are either touching or too far apart), the “in-position” LED is illuminated red as described above, and the lower LED remains amber. The lower LED will not change to its normal state (either yellow for an NPC-D-6000 controller or green for an NS-D-6000 controller) until the sensor is brought within an acceptable range.

21.4 Sensor-only commands

21.4.1 controller.sensor-only.get

21.4.1.1 Description

Controller is sensor-only

21.4.1.2 Command

controller.sensor-only.get

21.4.1.3 Parameters

None

21.4.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Controller is sensor-only

21.4.1.5 Possible error reports

None

21.4.1.6 Minimum security level

controller.sensor-only.get	No security required
----------------------------	----------------------

21.4.1.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

21.4.2 stage.mode.is-sensor-only.get

21.4.2.1 Description

Nanomechanism is sensor only and has no actuator

21.4.2.2 Command

```
stage.mode.is-sensor-only.get <channel>
```

21.4.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

21.4.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Nanomechanism is sensor only and has no actuator

21.4.2.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

21.4.2.6 Minimum security level

stage.mode.is-sensor-only.get	No security required
-------------------------------	----------------------

21.4.2.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

21.4.3 stage.position.measured.is-in-calibrated-range.get**21.4.3.1 Description**

Stage measured position is within calibrated range

21.4.3.2 Command

```
stage.position.measured.is-in-calibrated-range.get <channel>
```

21.4.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

21.4.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage measured position is within calibrated range

21.4.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

21.4.3.6 Minimum security level

stage.position.measured.is-in-calibrated-range.get	No security required
--	----------------------

21.4.3.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

21.4.4 stage.position.measured.is-readable.get

21.4.4.1 Description

Stage measured position is within readable range

21.4.4.2 Command

```
stage.position.measured.is-readable.get <channel>
```

21.4.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

21.4.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Stage measured position is within readable range

21.4.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

21.4.4.6 Minimum security level

stage.position.measured.is-readable.get	No security required
---	----------------------

21.4.4.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

21.4.5 stage.position.calibrated-range.minimum.get

21.4.5.1 Description

Stage measured position lower end of calibrated range

21.4.5.2 Command

```
stage.position.calibrated-range.minimum.get <channel>
```

21.4.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

21.4.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage measured position lower end of calibrated range

21.4.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

21.4.5.6 Minimum security level

stage.position.calibrated-range.minimum.get	No security required
---	----------------------

21.4.5.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

21.4.6 stage.position.calibrated-range.maximum.get

21.4.6.1 Description

Stage measured position higher end of calibrated range

21.4.6.2 Command

```
stage.position.calibrated-range.maximum.get <channel>
```

21.4.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

21.4.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage measured position higher end of calibrated range

21.4.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

21.4.6.6 Minimum security level

stage.position.calibrated-range.maximum.get	No security required
---	----------------------

21.4.6.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

21.4.7 stage.position.calibrated-range.range.get

21.4.7.1 Description

Stage measured position total calibrated range

21.4.7.2 Command

```
stage.position.calibrated-range.range.get <channel>
```

21.4.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

21.4.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	picometres			Stage measured position total calibrated range

21.4.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

21.4.7.6 Minimum security level

stage.position.calibrated-range.range.get	No security required
---	----------------------

21.4.7.7 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

22 Diagnostics logging

22.1 Overview

Diagnostics logging is captured for both controllers and stages. Because controllers and stages are designed to be interchangeable, this ensures either can be returned for servicing or repair and the appropriate information will still be available.

Maximum or worst-case values are captured since the controller or stage was last serviced. In addition, the controller diagnostics also captures values for the current power-on. The following data is logged and may be accessed by users.

- Number of power ons and total power on time
- Number of times fan was turned on and total fan on time (see 19.1.1)
- Number of protective shutdowns
- HV output current
- Internal controller temperatures (see 19.1.1)

Due to production procedures, testing by distributors, or other similar factors, customers may find that they receive stages or controllers with some power ons already logged. This is normal and should not be considered a problem.

22.2 Controller diagnostics

22.2.1 diagnostics-logging.controller.since-last-service.power-on.time.get

22.2.1.1 Description

Diagnostics logging total power on time for controller or channel

22.2.1.2 Command

```
diagnostics-logging.controller.since-last-service.power-on.time.get  
<channel>
```

22.2.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total power on time for controller 1-3 = Report total power on time for controller channel

22.2.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Total power on time

22.2.1.5 Possible error reports

None

22.2.1.6 Minimum security level

diagnostics-logging.controller.since-last-service.power-on.time.get	User level security required
---	------------------------------

22.2.1.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.2 diagnostics-logging.controller.since-last-service.power-on.total.get**22.2.2.1 Description**

Diagnostics logging total power ons for controller or channel

22.2.2.2 Command

```
diagnostics-logging.controller.since-last-service.power-on.total.get
<channel>
```

22.2.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total power ons for controller 1-3 = Report total power ons for controller channel

22.2.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total power ons

22.2.2.5 Possible error reports

None

22.2.2.6 Minimum security level

diagnostics-logging.controller.since-last-service.power-on.total.get	User level security required
--	------------------------------

22.2.2.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.3 diagnostics-logging.controller.since-last-service.fan-on.time.get**22.2.3.1 Description**

Diagnostics logging total fan on time

22.2.3.2 Command

diagnostics-logging.controller.since-last-service.fan-on.time.get

22.2.3.3 Parameters

None

22.2.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Total fan on time

22.2.3.5 Possible error reports

None

22.2.3.6 Minimum security level

diagnostics-logging.controller.since-last-service.fan-on.time.get	User level security required
---	------------------------------

22.2.3.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.4 diagnostics-logging.controller.since-last-service.fan-on.total.get**22.2.4.1 Description**

Diagnostics logging total fan ons

22.2.4.2 Command

diagnostics-logging.controller.since-last-service.fan-on.total.get

22.2.4.3 Parameters

None

22.2.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total fan ons

22.2.4.5 Possible error reports

None

22.2.4.6 Minimum security level

diagnostics-logging.controller.since-last-service.fan-on.total.get	User level security required
--	------------------------------

22.2.4.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.5 diagnostics-logging.controller.since-last-service.shutdowns.total.get**22.2.5.1 Description**

Diagnostics logging total protective shutdowns

22.2.5.2 Command

```
diagnostics-logging.controller.since-last-service.shutdowns.total.get
<channel>
```

22.2.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total power ons for controller 1-3 = Report total power ons for controller channel

22.2.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total shutdowns

22.2.5.5 Possible error reports

None

22.2.5.6 Minimum security level

diagnostics-logging.controller.since-last-service.shutdowns.total.get	User level security required
---	------------------------------

22.2.5.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.6 diagnostics-logging.controller.since-last-service.current.max-positive.get**22.2.6.1 Description**

Diagnostics logging maximum positive HV output current for controller or channel

22.2.6.2 Command

```
diagnostics-logging.controller.since-last-service.current.max-
positive.get <channel>
```

22.2.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total current for all controller channels 1-3 = Report current for controller channel

22.2.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	mA			Maximum positive current

22.2.6.5 Possible error reports

None

22.2.6.6 Minimum security level

diagnostics-logging.controller.since-last-service.current.max-positive.get	User level security required
--	------------------------------

22.2.6.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.7 diagnostics-logging.controller.since-last-service.current.max-negative.get**22.2.7.1 Description**

Diagnostics logging maximum negative HV output current for controller or channel

22.2.7.2 Command

```
diagnostics-logging.controller.since-last-service.current.max-negative.get <channel>
```

22.2.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total current for all controller channels 1-3 = Report current for controller channel

22.2.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	mA			Maximum negative current

22.2.7.5 Possible error reports

None

22.2.7.6 Minimum security level

diagnostics-logging.controller.since-last-service.current.max-negative.get	User level security required
--	------------------------------

22.2.7.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.8 diagnostics-logging.controller.since-last-service.temperature.max.get**22.2.8.1 Description**

Diagnostics logging maximum temperature for PSU (channel=0) or heatsink (channel=1-3)

22.2.8.2 Command

```
diagnostics-logging.controller.since-last-service.temperature.max.get
<channel>
```

22.2.8.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Controller PSU temperature (see 19.3.7) 1-3 = Controller channel heatsink temperature (see 19.3.2)

22.2.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Maximum temperature

22.2.8.5 Possible error reports

None

22.2.8.6 Minimum security level

diagnostics-logging.controller.since-last-service.temperature.max.get	User level security required
---	------------------------------

22.2.8.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.9 diagnostics-logging.controller.this-power-on.power-on.time.get

22.2.9.1 Description

Diagnostics logging total power on time for controller or channel

22.2.9.2 Command

```
diagnostics-logging.controller.this-power-on.power-on.time.get
<channel>
```

22.2.9.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total power on time for controller 1-3 = Report total power on time for controller channel

22.2.9.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Total power on time

22.2.9.5 Possible error reports

None

22.2.9.6 Minimum security level

diagnostics-logging.controller.this-power-on.power-on.time.get	User level security required
--	------------------------------

22.2.9.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.10 diagnostics-logging.controller.this-power-on.power-on.total.get**22.2.10.1 Description**

Diagnostics logging total power ons for controller or channel

22.2.10.2 Command

```
diagnostics-logging.controller.this-power-on.power-on.total.get
<channel>
```

22.2.10.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total power ons for controller 1-3 = Report total power ons for controller channel

22.2.10.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total power ons

22.2.10.5 Possible error reports

None

22.2.10.6 Minimum security level

diagnostics-logging.controller.this-power-on.power-on.total.get	User level security required
---	------------------------------

22.2.10.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.11 diagnostics-logging.controller.this-power-on.fan-on.time.get**22.2.11.1 Description**

Diagnostics logging total fan on time

22.2.11.2 Command

diagnostics-logging.controller.this-power-on.fan-on.time.get

22.2.11.3 Parameters

None

22.2.11.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Total fan on time

22.2.11.5 Possible error reports

None

22.2.11.6 Minimum security level

diagnostics-logging.controller.this-power-on.fan-on.time.get	User level security required
--	------------------------------

22.2.11.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.12 diagnostics-logging.controller.this-power-on.fan-on.total.get**22.2.12.1 Description**

Diagnostics logging total fan ons

22.2.12.2 Command

diagnostics-logging.controller.this-power-on.fan-on.total.get

22.2.12.3 Parameters

None

22.2.12.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total fan ons

22.2.12.5 Possible error reports

None

22.2.12.6 Minimum security level

diagnostics-logging.controller.this-power-on.fan-on.total.get	User level security required
---	------------------------------

22.2.12.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.13 diagnostics-logging.controller.this-power-on.shutdowns.total.get**22.2.13.1 Description**

Diagnostics logging total protective shutdowns

22.2.13.2 Command

```
diagnostics-logging.controller.this-power-on.shutdowns.total.get
<channel>
```

22.2.13.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total power ons for controller 1-3 = Report total power ons for controller channel

22.2.13.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total shutdowns

22.2.13.5 Possible error reports

None

22.2.13.6 Minimum security level

diagnostics-logging.controller.this-power-on.shutdowns.total.get	User level security required
--	------------------------------

22.2.13.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.14 diagnostics-logging.controller.this-power-on.current.max-positive.get**22.2.14.1 Description**

Diagnostics logging maximum positive HV output current for controller or channel

22.2.14.2 Command

```
diagnostics-logging.controller.this-power-on.current.max-positive.get
<channel>
```

22.2.14.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total current for all controller channels 1-3 = Report current for controller channel

22.2.14.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	mA			Maximum negative current

22.2.14.5 Possible error reports

None

22.2.14.6 Minimum security level

diagnostics-logging.controller.this-power-on.current.max-positive.get	User level security required
---	------------------------------

22.2.14.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.15 diagnostics-logging.controller.this-power-on.current.max-negative.get**22.2.15.1 Description**

Diagnostics logging maximum negative HV output current for controller or channel

22.2.15.2 Command

```
diagnostics-logging.controller.this-power-on.current.max-negative.get
<channel>
```

22.2.15.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Report total current for all controller channels 1-3 = Report current for controller channel

22.2.15.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	mA			Maximum positive current

22.2.15.5 Possible error reports

None

22.2.15.6 Minimum security level

diagnostics-logging.controller.this-power-on.current.max-negative.get	User level security required
---	------------------------------

22.2.15.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.2.16 diagnostics-logging.controller.this-power-on.temperature.max.get**22.2.16.1 Description**

Diagnostics logging maximum temperature for PSU (channel=0) or heatsink (channel=1-3)

22.2.16.2 Command

```
diagnostics-logging.controller.this-power-on.temperature.max.get
<channel>
```

22.2.16.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				0 = Controller PSU temperature (see 19.3.7) 1-3 = Controller channel heatsink temperature (see 19.3.2)

22.2.16.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating point	°C			Maximum temperature

22.2.16.5 Possible error reports

None

22.2.16.6 Minimum security level

diagnostics-logging.controller.this-power-on.temperature.max.get	User level security required
--	------------------------------

22.2.16.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.3 Stage diagnostics

22.3.1 diagnostics-logging.stage.since-last-service.power-on.time.get

22.3.1.1 Description

Diagnostics logging total power on time for stage

22.3.1.2 Command

```
diagnostics-logging.stage.since-last-service.power-on.time.get  
<channel>
```

22.3.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

22.3.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	0		Total power on time

22.3.1.5 Possible error reports

None

22.3.1.6 Minimum security level

diagnostics-logging.stage.since-last-service.power-on.time.get	User level security required
--	------------------------------

22.3.1.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.3.2 diagnostics-logging.stage.since-last-service.power-on.total.get**22.3.2.1 Description**

Diagnostics logging total power ons for stage

22.3.2.2 Command

```
diagnostics-logging.stage.since-last-service.power-on.total.get
<channel>
```

22.3.2.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

22.3.2.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total power ons

22.3.2.5 Possible error reports

None

22.3.2.6 Minimum security level

diagnostics-logging.stage.since-last-service.power-on.total.get	User level security required
---	------------------------------

22.3.2.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.3.3 diagnostics-logging.stage.since-last-service.shutdowns.total.get**22.3.3.1 Description**

Diagnostics logging total protective shutdowns

22.3.3.2 Command

```
diagnostics-logging.stage.since-last-service.shutdowns.total.get
<channel>
```

22.3.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

22.3.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Total shutdowns

22.3.3.5 Possible error reports

None

22.3.3.6 Minimum security level

diagnostics-logging.stage.since-last-service.shutdowns.total.get	User level security required
--	------------------------------

22.3.3.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.3.4 diagnostics-logging.stage.since-last-service.current.max-positive.get**22.3.4.1 Description**

Diagnostics logging maximum positive HV output current for stage

22.3.4.2 Command

```
diagnostics-logging.stage.since-last-service.current.max-positive.get
<channel>
```

22.3.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

22.3.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	mA			Maximum positive current

22.3.4.5 Possible error reports

None

22.3.4.6 Minimum security level

diagnostics-logging.stage.since-last-service.current.max-positive.get	User level security required
---	------------------------------

22.3.4.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

22.3.5 diagnostics-logging.stage.since-last-service.current.max-negative.get**22.3.5.1 Description**

Diagnostics logging maximum negative HV output current for stage

22.3.5.2 Command

```
diagnostics-logging.stage.since-last-service.current.max-negative.get
<channel>
```

22.3.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

22.3.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	mA			Maximum negative current

22.3.5.5 Possible error reports

None

22.3.5.6 Minimum security level

diagnostics-logging.stage.since-last-service.current.max-negative.get	User level security required
---	------------------------------

22.3.5.7 Supported in

Controller application firmware	6.5.1 onwards
Controller interface library	2.5.1 onwards

23 Deprecated commands

The following commands have been replaced with versions which are considered more user-friendly. However they are retained in the interface DLL and firmware for backwards compatibility with older code.

23.1 Command processing

23.1.1 stage.mode.digital-command.get/set

23.1.1.1 Description

Enable digital position commands (from host PC, snapshot step and function playback)

23.1.1.2 Command

```
stage.mode.digital-command.get <channel>  
stage.mode.digital-command.set <channel> <value>
```

23.1.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.1.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	Boolean	0	1	Enable digital position commands 1 = enable 0 = disable

23.1.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	Boolean	0	1	Enable digital position commands 1 = enable 0 = disable

23.1.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.1.1.7 Minimum security level

stage.mode.digital-command.get	No security required
stage.mode.digital-command.set	Superuser level security required

23.1.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.1.1.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.1.1.10 Replaced by

None.

Whilst this command still exists for backwards compatibility, globally disabling all digital commands has effects which may not give the desired results, in particular when setting absolute position. It may also be unclear to the user why operations such as triggering function playback do not take effect after disabling. All digital commands require positive user action so cannot be carried out accidentally, and have no negative side effects such as noise, so there is no benefit for the user. It is therefore recommended that users do not use this command.

23.2 Measured position processing

23.2.1 stage.position.low-pass-filter.filter-location.get/set

23.2.1.1 Description

Enabling and location of measured position low-pass filter in control loop

23.2.1.2 Command

```
stage.position.low-pass-filter.filter-location.get <channel>
```

```
stage.position.low-pass-filter.filter-location.set <channel> <value>
```

23.2.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.2.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	enum	0	2	Location of filter 0=filter is disabled 1=filter is enabled for both control loop and position monitor output 2=filter is enabled only for position monitor output; filter is disabled for control loop

23.2.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	enum	0	2	Location of filter 0=filter is disabled 1=filter is enabled for both control loop and position monitor output 2=filter is enabled only for position monitor output; filter is disabled for control loop

23.2.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Low-pass filter location value is invalid (set only)

23.2.1.7 Minimum security level

stage.position.low-pass-filter.filter-location.get	User level security required
stage.position.low-pass-filter.filter-location.set	Superuser level security required

23.2.1.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.2.1.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.2.1.10 Replaced by

Equivalent `stage.position.low-pass-filter.enable`, `stage.position.low-pass-filter.to-output` and `stage.position.output.low-pass-filter.enable` commands as described in 8.4.2 and 8.4.7, allowing more fine-grained control of filter use. The deprecated “get” command will return the closest result possible if a routing does not exactly correspond to those supported by this command, and the deprecated “set” command will still allow routings to be selected from those it supports.

23.2.2 stage.position.low-pass-filter.q.get/set

23.2.2.1 Description

Measured position low-pass filter Q factor

23.2.2.2 Command

```
stage.position.low-pass-filter.q.get <channel>  
stage.position.low-pass-filter.q.set <channel> <value>
```

23.2.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.2.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0		Filter Q factor

23.2.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0		Filter Q factor

23.2.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Q factor must be within specified range (set only)

23.2.2.7 Minimum security level

stage.position.low-pass-filter.q.get	User level security required
stage.position.low-pass-filter.q.set	Superuser level security required

23.2.2.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.2.2.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.2.2.10 Replaced by

Latest firmware does not use a Q factor for the low-pass filter, so no equivalent command is required.

23.4 Control loop configuration

23.4.1 stage.mode.get

23.4.1.1 Description

Stage mode and status

23.4.1.2 Command

stage.mode.get <channel>

23.4.1.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.1.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			<p>Bit 12: Controller is synchronisation slave and synchronisation is locked to master</p> <p>Bit 11: Controller is synchronisation master</p> <p>Bit 10: Protection shutdown active (see 19.1.1)</p> <p>Bit 4: Stage in position unconfirmed</p> <p>Bit 2: Stage “in position” confirmed by window filter</p> <p>Bit 1: Stage connected</p> <p>Bit 0: Stage “in position” confirmed by low-pass filter</p>
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 3: Servo output frozen Bit 1: Closed loop control enabled

23.4.1.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.1.6 Minimum security level

stage.mode.get	No security required
----------------	----------------------

23.4.1.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.1.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.1.9 Replaced by

Equivalent commands as described in 20.1.1, 20.1.2, 19.1.1, 5.2.1, 8.3.5, 8.5.5, 8.5.6, 8.5.7, 8.5.8, 8.6.1, 8.12.1.

23.4.2 stage.mode-only.get/set

23.4.2.1 Description

Stage mode

23.4.2.2 Command

```
stage.mode-only.get <channel>  
stage.mode-only.set <channel> <mode>
```

23.4.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
mode	16-bit unsigned integer	Bitfield			Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1=window filter selected Bit 9: Digital command enabled Bit 6: Analogue input command enabled Bit 3: Servo output frozen Bit 1: Closed loop control enabled

This allows the user to change multiple settings in one operation.

Each bit within the bitfield is protected by security, corresponding with security levels for corresponding commands. No error will be reported if bits are set without having the correct security level, but that setting will not be changed. The return value reports the actual state of all these modes.

Stage “in position” output confirmation	Superuser level security required
Digital command enabled	Superuser level security required
Analogue input command enabled	Superuser level security required

Servo output frozen	User level security required
Closed loop control enabled	Superuser level security required

23.4.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1=window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p> <p>Bit 1: Closed loop control enabled</p>

23.4.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.2.7 Minimum security level

stage.mode-only.get	No security required
stage.mode-only.set	User level required

23.4.2.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.2.9 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.2.10 Replaced by

Equivalent commands as described in 20.1.1, 20.1.2, 19.1.1, 5.2.1, 8.3.5, 8.5.5, 8.5.6, 8.5.7, 8.5.8, 8.6.1, 8.12.1.

23.4.3 stage.mode-mask.set

23.4.3.1 Description

Set stage mode using mask to select changes

23.4.3.2 Command

```
stage.mode-mask.set <channel> <mask> <mode>
```

23.4.3.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
mask	16-bit unsigned integer	Bitfield			<p>Mode setting mask</p> <p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1=window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p> <p>Bit 1: Closed loop control enabled</p>
mode	16-bit unsigned integer	Bitfield			<p>Mode setting states</p> <p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1=window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p> <p>Bit 1: Closed loop control enabled</p>

This allows the user to change multiple settings in one operation, but leave other settings unchanged if required. Bits set in the “mask” cause the settings in the “mode” to take effect. Where bits are cleared for settings in the “mask”, those mode settings are left unchanged.

Each bit within the bitfield is protected by security, corresponding with security levels for corresponding commands. No error will be reported if bits are set without having the correct security level, but that setting will not be changed. The return value reports the actual state of all these modes.

Stage “in position” output confirmation	Superuser level security required
Digital command enabled	Superuser level security required
Analogue input command enabled	Superuser level security required
Servo output frozen	User level security required
Closed loop control enabled	Superuser level security required

23.4.3.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			Bit 12: Controller is synchronisation slave and synchronisation is locked to master Bit 11: Controller is synchronisation master Bit 10: Protection shutdown active (see 19.1.1) Bit 4: Stage in position unconfirmed Bit 2: Stage “in position” confirmed by window filter Bit 1: Stage connected Bit 0: Stage “in position” confirmed by low-pass filter
mode	16-bit unsigned integer	Bitfield			Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected Bit 9: Digital command enabled Bit 6: Analogue input command enabled

Name	Type	Units	Minimum	Maximum	Description
					Bit 3: Servo output frozen Bit 1: Closed loop control enabled

23.4.3.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.3.6 Minimum security level

stage.mode-mask.set	User level required
---------------------	---------------------

23.4.3.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.3.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.3.9 Replaced by

Equivalent commands as described in 20.1.1, 20.1.2, 19.1.1, 5.2.1, 8.3.5, 8.5.5, 8.5.6, 8.5.7, 8.5.8, 8.6.1, 8.12.1.

23.4.4 stage.mode.analogue-command.enable/disable

23.4.4.1 Description

Enable/disable analogue input position commands

23.4.4.2 Command

```
stage.mode.analogue-command.enable <channel>
```

```
stage.mode.analogue-command.disable <channel>
```

23.4.4.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.4.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			<p>Bit 12: Controller is synchronisation slave and synchronisation is locked to master</p> <p>Bit 11: Controller is synchronisation master</p> <p>Bit 10: Protection shutdown active (see 19.1.1)</p> <p>Bit 4: Stage in position unconfirmed</p> <p>Bit 2: Stage “in position” confirmed by window filter</p> <p>Bit 1: Stage connected</p> <p>Bit 0: Stage “in position” confirmed by low-pass filter</p>
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Closed loop control enabled

23.4.4.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.4.6 Minimum security level

stage.mode.analogue-command.enable/disable	Superuser level security required
--	-----------------------------------

23.4.4.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.4.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.4.9 Replaced by

Equivalent stage.mode.analogue-command.get/set commands as described in 8.3.5.

23.4.5 stage.mode.digital-command.enable/disable

23.4.5.1 Description

Enable/disable digital position command (direct command from PC, function playback and snapshot step)

23.4.5.2 Command

```
stage.mode.digital-command.enable <channel>
```

```
stage.mode.digital-command.disable <channel>
```

23.4.5.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.5.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			<p>Bit 12: Controller is synchronisation slave and synchronisation is locked to master</p> <p>Bit 11: Controller is synchronisation master</p> <p>Bit 10: Protection shutdown active (see 19.1.1)</p> <p>Bit 4: Stage in position unconfirmed</p> <p>Bit 2: Stage “in position” confirmed by window filter</p> <p>Bit 1: Stage connected</p> <p>Bit 0: Stage “in position” confirmed by low-pass filter</p>
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 3: Servo output frozen Bit 1: Closed loop control enabled

23.4.5.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.5.6 Minimum security level

stage.mode.digital-command.enable/disable	Superuser level security required
---	-----------------------------------

23.4.5.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.5.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.5.9 Replaced by

Equivalent stage.mode.digital-command.get/set commands as described in 23.1.1, also deprecated.

23.4.6 stage.mode.closed-loop.enable/disable

23.4.6.1 Description

Enable/disable closed-loop control operation (and hence disable/enable open-loop operation)

23.4.6.2 Command

```
stage.mode.closed-loop.enable <channel>
```

```
stage.mode.closed-loop.disable <channel>
```

23.4.6.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.6.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			<p>Bit 12: Controller is synchronisation slave and synchronisation is locked to master</p> <p>Bit 11: Controller is synchronisation master</p> <p>Bit 10: Protection shutdown active (see 19.1.1)</p> <p>Bit 4: Stage in position unconfirmed</p> <p>Bit 2: Stage “in position” confirmed by window filter</p> <p>Bit 1: Stage connected</p> <p>Bit 0: Stage “in position” confirmed by low-pass filter</p>
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Closed loop control enabled

23.4.6.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.6.6 Minimum security level

stage.mode.closed-loop.enable/disable	Superuser level security required
---------------------------------------	-----------------------------------

23.4.6.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.6.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.6.9 Replaced by

Equivalent stage.mode.closed-loop.get/set commands as described in 8.6.1.

23.4.7 stage.mode.freeze-servo-output.enable/disable

23.4.7.1 Description

Freeze/unfreeze servo output

23.4.7.2 Command

```
stage.mode.freeze-servo-output.enable <channel>
```

```
stage.mode.freeze-servo-output.disable <channel>
```

23.4.7.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.7.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			<p>Bit 12: Controller is synchronisation slave and synchronisation is locked to master</p> <p>Bit 11: Controller is synchronisation master</p> <p>Bit 10: Protection shutdown active (see 19.1.1)</p> <p>Bit 4: Stage in position unconfirmed</p> <p>Bit 2: Stage “in position” confirmed by window filter</p> <p>Bit 1: Stage connected</p> <p>Bit 0: Stage “in position” confirmed by low-pass filter</p>
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Closed loop control enabled

23.4.7.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.7.6 Minimum security level

stage.mode.freeze-servo-output.enable/disable	User level security required
---	------------------------------

23.4.7.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.7.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.7.9 Replaced by

Equivalent stage.mode.freeze-servo-output.get/set commands as described in 8.12.1.

23.4.8 stage.mode.in-position-output-select.lpf/window-filter

23.4.8.1 Description

Select stage “in position” (ready) digital output to be confirmed by low-pass filter or window filter

23.4.8.2 Command

```
stage.mode.in-position-output-select.lpf <channel>
```

```
stage.mode.in-position-output-select.window-filter <channel>
```

23.4.8.3 Parameters

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.4.8.4 Results

Name	Type	Units	Minimum	Maximum	Description
status	16-bit unsigned integer	Bitfield			<p>Bit 12: Controller is synchronisation slave and synchronisation is locked to master</p> <p>Bit 11: Controller is synchronisation master</p> <p>Bit 10: Protection shutdown active (see 19.1.1)</p> <p>Bit 4: Stage in position unconfirmed</p> <p>Bit 2: Stage “in position” confirmed by window filter</p> <p>Bit 1: Stage connected</p> <p>Bit 0: Stage “in position” confirmed by low-pass filter</p>
mode	16-bit unsigned integer	Bitfield			<p>Bit 10: Stage “in position” output confirmation, 0=low-pass filter selected, 1>window filter selected</p> <p>Bit 9: Digital command enabled</p> <p>Bit 6: Analogue input command enabled</p> <p>Bit 3: Servo output frozen</p>

Name	Type	Units	Minimum	Maximum	Description
					Bit 1: Closed loop control enabled

23.4.8.5 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel

23.4.8.6 Minimum security level

stage.mode.in-position-output-select.lpf/window-filter	Superuser level security required
--	-----------------------------------

23.4.8.7 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.4.8.8 Deprecated in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.4.8.9 Replaced by

Equivalent stage.mode.in-position-output-select.get/set commands as described in 8.5.8.

23.5 Closed-loop control

23.5.1 stage.pid.proportional.gain.get/set

23.5.1.1 Description

Closed-loop control PID loop proportional gain

23.5.1.2 Command

```
stage.pid.proportional.gain.get <channel>
stage.pid.proportional.gain.set <channel> <value>
```

23.5.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		-100	100	PID loop proportional gain

23.5.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		-100	100	PID loop proportional gain

23.5.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.1.7 Minimum security level

stage.pid.proportional.gain.get	User level security required
stage.pid.proportional.gain.set	Superuser level security required

23.5.1.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.5.1.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.1.10 Replaced by

Equivalent stage.closed-loop.pid.position.proportional.gain.get/set commands as described in 8.10.4.

23.5.2 stage.pid.proportional.set-point-weighting.get/set

23.5.2.1 Description

Closed-loop control PID loop proportional setpoint weighting

23.5.2.2 Command

```
stage.pid.proportional.set-point-weighting.get <channel>  
stage.pid.proportional.set-point-weighting.set <channel> <value>
```

23.5.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	1	PID loop proportional setpoint weighting

23.5.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	1	PID loop proportional setpoint weighting

23.5.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Setpoint weighting must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.2.7 Minimum security level

stage.pid.proportional.set-point-weighting.get	User level security required
stage.pid.proportional.set-point-weighting.set	Superuser level security required

23.5.2.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.5.2.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.2.10 Replaced by

Equivalent stage.closed-loop.pid.position.proportional.set-point-weighting.get/set commands as described in 8.10.5.

23.5.3 stage.pid.integral.time-constant.get/set

23.5.3.1 Description

Closed-loop control PID loop integral time constant

23.5.3.2 Command

```
stage.pid.integral.time-constant.get <channel>
stage.pid.integral.time-constant.set <channel> <value>
```

23.5.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	PID loop integral time constant

23.5.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	PID loop integral time constant

23.5.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.3.7 Minimum security level

stage.pid.integral.time-constant.get	User level security required
stage.pid.integral.time-constant.set	Superuser level security required

23.5.3.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.5.3.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.3.10 Replaced by

Equivalent stage.closed-loop.pid.position.integral.time-constant.get/set commands as described in 8.10.6.

23.5.4 stage.pid.integral.error-magnitude.max.get/set

23.5.4.1 Description

Closed-loop control PID integral maximum error magnitude

23.5.4.2 Command

```
stage.pid.integral.error-magnitude.get <channel>
```

```
stage.pid.integral.error-magnitude.set <channel> <value>
```

23.5.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	system units	0	16777215	PID loop integral maximum error magnitude

23.5.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	system units	0	16777215	PID loop integral maximum error magnitude

23.5.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Limit must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.4.7 Minimum security level

stage.pid.integral.error-magnitude.max.get	User level security required
stage.pid.integral.error-magnitude.max.set	Superuser level security required

23.5.4.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.5.4.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.4.10 Replaced by

Latest firmware does not apply limits to position error magnitude, so no equivalent command is required,

23.5.5 stage.pid.differential.gain.get/set

23.5.5.1 Description

Closed-loop control PID loop differential gain

23.5.5.2 Command

```
stage.pid.differential.gain.get <channel>  
stage.pid.differential.gain.set <channel> <value>
```

23.5.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		-10	10	PID loop differential gain

23.5.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		-10	10	PID loop differential gain

23.5.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.5.7 Minimum security level

stage.pid.differential.gain.get	User level security required
stage.pid.differential.gain.set	Superuser level security required

23.5.5.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.5.5.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.5.10 Replaced by

Equivalent stage.closed-loop.pid.position.differential.gain.get/set commands as described in 8.10.7.

23.5.6 stage.pid.differential.time-constant.get/set

23.5.6.1 Description

Closed-loop control PID loop differential time constant

23.5.6.2 Command

```
stage.pid.differential.time-constant.get <channel>  
stage.pid.differential.time-constant.set <channel> <value>
```

23.5.6.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.6.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	s	1e-6	1	PID loop differential time constant

23.5.6.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	s	1e-6	1	PID loop differential time constant

23.5.6.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Time constant must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.6.7 Minimum security level

stage.pid.differential.time-constant.get	User level security required
stage.pid.differential.time-constant.set	Superuser level security required

23.5.6.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.5.6.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.6.10 Replaced by

Equivalent stage.position.velocity-accel-filter.time-constant.get/set commands as described in 8.8.1.

23.5.7 stage.pid.feedforward.gain.get/set

23.5.7.1 Description

Closed-loop control PID loop feedforward gain

23.5.7.2 Command

```
stage.pid.feedforward.gain.get <channel>
```

```
stage.pid.feedforward.gain.set <channel> <value>
```

23.5.7.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.5.7.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		-10	10	PID loop feedforward gain

23.5.7.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		-10	10	PID loop feedforward gain

23.5.7.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.5.7.7 Minimum security level

stage.pid.feedforward.gain.get	User level security required
stage.pid.feedforward.gain.set	Superuser level security required

23.5.7.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.7.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.5.7.10 Replaced by

Equivalent stage.closed-loop.pid.position.feedforward.gain.get/set commands as described in 8.10.8.

Note that this was never available in earlier firmware. However where customers are still using older command names for PID configuration, this command allows position feedforward support to be added in a way which fits with older command names.

23.6 Damping

23.6.1 stage.notch-filter.filter-location.get/set

23.6.1.1 Description

Location and type of notch filter(s) in control loop

23.6.1.2 Command

```
stage.notch-filter.filter-location.get <channel>
```

```
stage.notch-filter.filter-location.set <channel> <value>
```

23.6.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.6.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit integer	Enum			<p>Location and type of notch filter(s) in control loop</p> <p>0=notch filter disabled</p> <p>1=single 2nd-order notch filter on control output</p> <p>2=deprecated</p> <p>3=single 4th-order notch filter (two 2nd-order filters in series) on control output (applied in open-loop and closed-loop)</p> <p>4=deprecated</p> <p>5=dual 2nd-order notch filters on control output (applied in open-loop and closed-loop)</p> <p>6=single 4th-order notch filter on control output</p>

23.6.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit integer	Enum			<p>Location and type of notch filter(s) in control loop</p> <p>0=notch filter disabled</p> <p>1=single 2nd-order notch filter on control output</p> <p>2=deprecated</p> <p>3=single 4th-order notch filter (two 2nd-order filters in series) on control output (applied in open-loop and closed-loop)</p> <p>4=deprecated</p> <p>5=dual 2nd-order notch filters on control output (applied in open-loop and closed-loop)</p> <p>6=single 4th-order notch filter on control output</p>

23.6.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Notch filter location value is invalid (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.6.1.7 Minimum security level

stage.notch-filter.filter-location.get	User level security required
stage.notch-filter.filter-location.set	Superuser level security required

23.6.1.8 Supported in

Controller application firmware	6.0.0 onwards
Controller interface library	2.0.0 onwards

23.6.1.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.6.1.10 Replaced by

Equivalent stage.notch-filter.filter-type.get/set commands as described in 8.11.1.

The latest firmware has removed the ability to move the filter location, because this was not found to be beneficial. As a result, setting 2 (previously setting single 4th-order notch filter on PID output) or 4 (previously setting dual 2nd-order notch filters on PID output) will return values 3 (single 4th-order notch filter on control output) or 5 (dual 2nd-order notch filters on control output) respectively.

Firmware version 6.6.1 adds two additional options which can be selected.

- 1=single 2nd-order notch filter on control output
- 6=single 4th-order notch filter on control output

These options were not available on earlier firmware.

23.7 Active damping

23.7.1 stage.active-damping.accel.gain.get/set

23.7.1.1 Description

Active damping acceleration damping gain

23.7.1.2 Command

```
stage.active-damping.accel.gain.get <channel>  
stage.active-damping.accel.gain.set <channel> <value>
```

23.7.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.7.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point		0	10	Acceleration damping gain

23.7.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point		0	10	Acceleration damping gain

23.7.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Gain must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.7.1.7 Minimum security level

stage.active-damping.accel.gain.get	User level security required
stage.active-damping.accel.gain.set	Superuser level security required

23.7.1.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.7.1.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.7.1.10 Replaced by

None.

Direct negative feedback of acceleration creates “virtual mass”. However adding mass (real or “virtual”) to the system under control tends to make closed-loop PID control less stable. The commands remain available, but it is recommended that they are not used.

23.7.2 stage.active-damping.accel.deadband.get/set

23.7.2.1 Description

Active damping acceleration deadband

23.7.2.2 Command

```
stage.active-damping.accel.deadband.get <channel>  
stage.active-damping.accel.deadband.set <channel> <value>
```

23.7.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.7.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit floating-point	pm/s/s	0		Acceleration deadband

23.7.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit floating-point	pm/s/s	0		Acceleration deadband

23.7.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Channel not available	No stage is connected to the specified channel
Value out of range	Deadband must be within specified range (set only)
Command not valid for sensor	The command is related to closed-loop control and is not valid for a sensor connected (see section 21).

23.7.2.7 Minimum security level

stage.active-damping.accel.deadband.get	User level security required
stage.active-damping.accel.deadband.set	Superuser level security required

23.7.2.8 Supported in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.7.2.9 Deprecated in

Controller application firmware	6.6.1 onwards
Controller interface library	2.6.1 onwards

23.7.2.10 Replaced by

None.

Direct negative feedback of acceleration creates “virtual mass”. However adding mass (real or “virtual”) to the system under control tends to make closed-loop PID control less stable. The commands remain available, but it is recommended that they are not used.

23.8 Function playback configuration

23.8.1 function.waveform.steps-per-sample.get/set

23.8.1.1 Description

Function playback control loop sample time steps per waveform data point

23.8.1.2 Command

```
function.waveform.steps-per-sample.get <channel>
```

```
function.waveform.steps-per-sample.set <channel> <value>
```

23.8.1.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.8.1.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	1		Steps per data point

23.8.1.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	1		Steps per data point

23.8.1.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Number of steps must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

23.8.1.7 Minimum security level

function.waveform.steps-per-sample.get	No security required
function.waveform.steps-per-sample.set	User level security required

23.8.1.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.8.1.9 Deprecated in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

23.8.1.10 Replaced by

Equivalent function.waveform.sample-period.get/set commands to set value in seconds as described in 14.2.5.

23.8.2 function.waveform.steps-per-trigger-out-pulse.get/set

23.8.2.1 Description

Function playback waveform trigger output pulse duration

23.8.2.2 Command

```
function.waveform.steps-per-trigger-out-pulse.get <channel>
```

```
function.waveform.steps-per-trigger-out-pulse.set <channel> <value>
```

23.8.2.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.8.2.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	1		Pulse duration

23.8.2.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	1		Pulse duration

23.8.2.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Value must be as specified
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

23.8.2.7 Minimum security level

function.waveform.trigger-out-event.get	No security required
function.waveform.trigger-out-event.set	User level security required

23.8.2.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.8.2.9 Deprecated in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

23.8.2.10 Replaced by

Equivalent function.trigger-output.pulse-time.get/set commands to set value in seconds as described in 14.6.10.

23.8.3 function.waveform.waveform-start.get/set

23.8.3.1 Description

Function playback waveform data index for start of waveform section (and end of opening section)

23.8.3.2 Command

```
function.waveform.waveform-start.get <channel>  
function.waveform.waveform-start.set <channel> <value>
```

23.8.3.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.8.3.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	0	See 14.2	Waveform start index

23.8.3.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 14.2	Waveform start index

23.8.3.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	Start index must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

23.8.3.7 Minimum security level

function.waveform.waveform-start.get	No security required
function.waveform.waveform-start.set	User level security required

23.8.3.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.8.3.9 Deprecated in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

23.8.3.10 Replaced by

Equivalent `function.waveform.repeat-start.get/set` commands to match waveform generator commands as described in 14.3.1.

23.8.4 function.waveform.waveform-end.get/set

23.8.4.1 Description

Function playback waveform data index for start of closing section (and end of waveform section)

23.8.4.2 Command

```
function.waveform.waveform-end.get <channel>
function.waveform.waveform-end.set <channel> <value>
```

23.8.4.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.8.4.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer	samples	0	See 14.2	Waveform end index

23.8.4.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer	samples	0	See 14.2	Waveform end index

23.8.4.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Value out of range	End index must be within specified range (set only)
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

23.8.4.7 Minimum security level

function.waveform.waveform-end.get	No security required
function.waveform.waveform-end.set	User level security required

23.8.4.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.8.4.9 Deprecated in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

23.8.4.10 Replaced by

Equivalent function.waveform.repeat-end.get/set commands to match waveform generator commands as described in 14.3.2.

23.8.5 function.waveform.iterations.get/set

23.8.5.1 Description

Function playback number of waveform iterations

23.8.5.2 Command

```
function.waveform.iterations.get <channel>  
function.waveform.iterations.set <channel> <value>
```

23.8.5.3 Parameters (get)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage

23.8.5.4 Parameters (set)

Name	Type	Units	Minimum	Maximum	Description
channel	8-bit unsigned integer				Controller channel for stage
value	32-bit unsigned integer				Number of iterations 0 = run indefinitely 1+ = run this many iterations

23.8.5.5 Results

Name	Type	Units	Minimum	Maximum	Description
value	32-bit unsigned integer				Number of iterations 0 = run indefinitely 1+ = run this many iterations

23.8.5.6 Possible error reports

Error return “errcode” string reported	Description
Channel number invalid	Channel parameter is larger than number of channels on controller
Command could not be carried out	Function playback is running or waiting for digital input trigger, so configuration cannot be changed

23.8.5.7 Minimum security level

function.waveform.iterations.get	No security required
function.waveform.iterations.set	User level security required

23.8.5.8 Supported in

Controller application firmware	6.2.1 onwards
Controller interface library	2.2.1 onwards

23.8.5.9 Deprecated in

Controller application firmware	6.4.1 onwards
Controller interface library	2.4.1 onwards

23.8.5.10 Replaced by

Equivalent function.waveform.repeat-count.get/set commands to match waveform generator commands as described in 14.3.3.

Queensgate is a registered trade mark and a trading name of Prior Scientific Instruments Limited, a company incorporated and registered in England and Wales with company number 00404087 and whose registered office is 3/4 Fielding Industrial Estate, Wilbraham Road, Fulbourn, Cambridge, Cambridgeshire, CB21 5ET. All references throughout this User Manual to Queensgate are references to Prior Scientific Instruments Limited.



Prior Scientific Ltd
3-4 Fielding Industrial Estate
Wilbraham Road • Fulbourn
Cambridge • CB21 5ET • UK

t: +44 (0)1223 881711
e: uksales@prior.com
www.prior-scientific.co.uk



Prior Scientific Inc
80 Reservoir Park Drive
Rockland • MA. 02370
U.S.A.

t: +1 781-878-8442
e: info@prior.com
www.prior-us.com



Prior Scientific GmbH
Wildenbruchstr. 15
D-07745
Jena • Germany

t: +49 (0) 3641 675 650
e: jena@prior.com
www.prior-instruments.de



Prior Scientific KK
Kayabacho 3rd Nagaoka Bldg 10F
2-7-10 • Nihonbashi Kayabacho
Chuo-Ku • Tokyo • 103-0025 • Japan

t: +81-3-5652-8831
e: info-japan@prior.com
www.priorjp.co.jp